

模块四 汽车仪表系统控制

项目一 汽车里程表

任务一 LED 数码管显示认知

知识目标

1. 掌握 LED 数码管的结构。
2. 了解 LED 数码管的分类。
3. 掌握 LED 数码管的工作原理。
4. 掌握 LED 数码管的编码方式。

能力目标

1. 能进行 LED 数码管的接口电路设计。
2. 能用单片机进行一位数码管循环显示 0~9 的设计。
3. 能用单片机进行汽车生产线产品计件系统的设计。

任务相关知识

显示器是单片机应用系统常用的设备，显示器接口是实现单片机信息输出的重要部分。用户的程序、数据、命令等都需要通过显示器才能显示，才能知道输入的正确与否。目前单片机应用中，常用的有发光二极管（LED）显示器、液晶显示器（Liquid Crystal Display, LCD）和阴极射线管（Cathode Ray Tube, CRT）显示器。

发光二极管和数码管是单片机应用系统的主要显示器，发光二极管可以用于状态的指示，数码管用于数字和字符的显示。在控制中，有静态显示和动态显示两种，与单片机的接口也有并行和串行两种方式。由于单片机接口的驱动能力较差，一般需要在单片机和 LED 之间加接电路实现译码驱动。应该根据 LED 数码管的位数和电流来决定接口电路的形式。

LED 数码显示器，又称 LED 数码管，是将若干 LED 按一定图形排列并封装在仪器中的最常用的数码显示器之一。LED 数码管具有发光显示清晰、响应速度快、耗电小、体积小、寿命长、耐冲击、易与各种驱动电路连接等优点，在各种数显仪器仪表、数字控制设备中得到了广泛应用。

在单片机系统中，经常用 LED 数码管来显示单片机系统的工作状态、运算结果等各种

信息，LED 数码管显示器是单片机实现人机对话的一种重要输出设备。

1. LED 数码管的结构

LED 数码管实际上由 8 个 LED 构成，其中 7 个独立的条形 LED 排列成“8”字形的笔画段，1 个 LED 为圆点形状，安装在显示器的右下角作为小数点使用。每段的亮灭均可单独控制。图 4-1 所示为 LED 数码管的外形结构及引脚，共有 10 个引脚，其中两个引脚为公共电极，在内部已经连接在一起，连接在公共端 com（com 被定义为位选端），剩余的 8 个引脚分别对应 LED 数码管上的 8 个段选端。通过 LED 亮暗的不同组合，从而可显示出“0~9”数字、“A~F”字母等各种字符。

LED 数码管的内部结构共有两种不同形式，一种是共阳极结构，称为共阳极 Common Anode（数码管），如图 4-2a 所示，即 8 个 LED 的正极全部连接在一起组成公共端。另一种是共阴极结构，称为共阴极 Common Cathode（数码管），如图 4-2b 所示，即 8 个 LED 的负极全部连接在一起组成公共端。

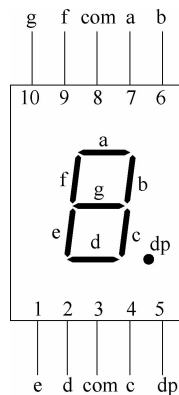


图 4-1 LED 数码管的外形结构及引脚

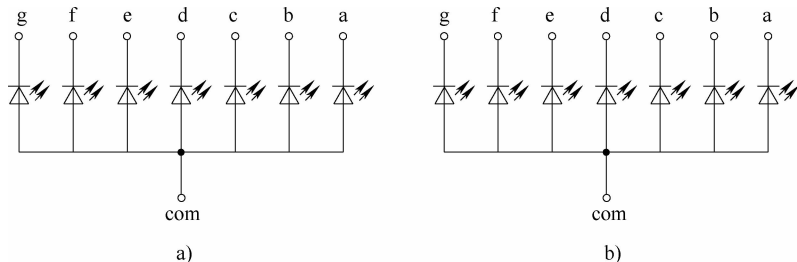


图 4-2 LED 数码管的内部结构

a) 共阳极 LED 数码管内部结构 b) 共阴极 LED 数码管内部结构

2. LED 数码管的工作原理

通过控制段选位的高低电平控制相应段的 LED 发光与熄灭，LED 数码管可以显示不同的数字、字母和符号。

共阳极 LED 数码管的 8 个 LED 的正极全部连接在一起组成公共端，8 个 LED 的负极各自独立引出。使用时公共端接电源 +5V，这时负极接低电平的 LED 就导通点亮，接高电平的则不点亮，相应的段被显示。根据 LED 的不同组合可显示出各种数字或字符。

共阴极 LED 数码管的 8 个 LED 的负极全部连接在一起组成公共端，8 个 LED 的正极则各自独立引出。

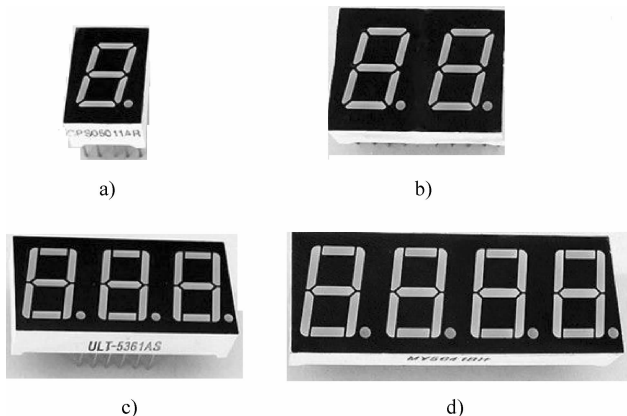


图 4-3 不同位 LED 数码管

a) 1 位 b) 2 位 c) 3 位 d) 4 位

使用时公共端接地，这时负极接高电平的 LED 就导通点亮，接低电平的则不点亮，相应的段被显示。根据 LED 的不同组合可显示出各种数字或字符。

4. LED 数码管的字形编码

要使 LED 数码管显示出数字或字符，必须使相应的段位点亮，这就要为 LED 数码管的各段提供一个字节的二进制代码，即字形编码（也称段码）。7 段 LED，再加上 1 个小数点位，共计 8 位代码，由一个数据字节提供，习惯上以“a”段对应字形编码字节的最低位。各数据位的对应关系见表 4-1。

表 4-1 LED 数码管数据位的对应关系表

数据位	D7	D6	D5	D4	D3	D2	D1	D0
显示段	dp	g	f	e	d	c	b	a

如使用共阳极 LED 数码管，数据为“0”表示对应字段亮，数据为“1”表示对应字段暗；如使用共阴极 LED 数码管，数据为“0”表示对应字段暗，数据为“1”表示对应字段亮。如要 LED 数码管显示“0”，则共阳极 LED 数码管的字形编码应为 1100000B（即 C0H）；共阴极 LED 数码管的字形编码应为 00111111（即 3FH）。各种字符的字形编码见表 4-2。如要在数码管上显示某一字符，只需将该字符的字形编码加到各段上即可。

表 4-2 LED 数码显示器的字形编码表

显示字形	字形编码（共阳极）	字形编码（共阴极）	显示字形	字形编码（共阳极）	字形编码（共阴极）
0	C0H	3FH	9	90H	6FH
1	F9H	06H	A	88H	77H
2	A4H	5BH	b	83H	7CH
3	B0H	4FH	C	C6H	39H
4	99H	66H	d	A1H	5EH
5	92H	6DH	E	86H	79H
6	82H	7DH	F	8EH	71H
7	F8H	07H	熄灭	FFH	00H
8	80H	7FH			

初步训练

利用单片机控制 1 位 LED 数码管循环显示“0~9”。

任务分析：

利用单片机 P1 口控制 1 位共阳极 LED 数码管 DS1 的 8 位段选位，公共端接 +5V 电源 V_{CC} 。当 P1 口输出不同字形编码的高低电平时，LED 数码管显示相应的字形。根据以上分析，画出电路图，如图 4-4 所示。程序流程图如图 4-5 所示。

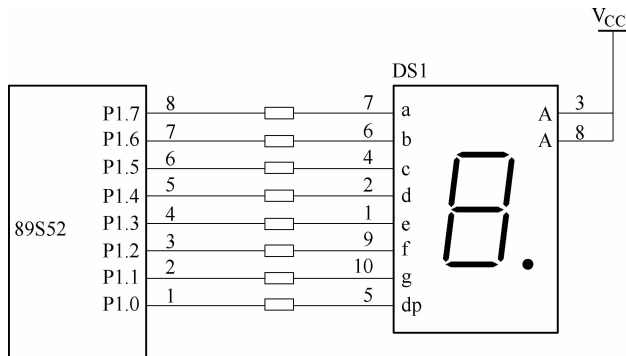


图 4-4 LED 数码管静态显示电路图

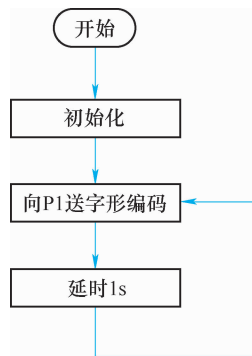


图 4-5 程序流程图

操作步骤:

- 1) 将单片机 V_{CC} 接 +5V, GND 接地, \overline{EA} 接 +5V。
- 2) 将 P1.0 ~ P1.7 接显示模块 LED 数码管的 D0 ~ D7, 打开电源。
- 3) 打开 Keil μ Vision4 软件, 根据流程图进行编程, 并调试通过, 生成 HEX 文件。
- 4) 将生成的 HEX 文件下载到单片机中。
- 5) 观察 LED 数码管的现象, 并记录在任务工单中。
- 6) 完成实训操作, 关闭电源, 拆除接线, 整理归位。
- 7) 完成任务工单中相关题目。

参考程序:

```

ORG    0000H
AJMP   MAIN
ORG    0030H

MAIN:                                     ; 主程序
MOV    P1, #00H                          ; 把 8 位 LED 数码管选中
MOV    P2, #0C0H                          ; 送“0”的字形编码
ACALL  DELAY                              ; 延时 1s
MOV    P2, #0F9H                          ; 送“1”的字形编码
ACALL  DELAY                              ; 延时 1s
MOV    P2, #0A4H                          ; 送“2”的字形编码
ACALL  DELAY                              ; 延时 1s
MOV    P2, #0B0H                          ; 送“3”的字形编码
ACALL  DELAY                              ; 延时 1s
MOV    P2, #99H                            ; 送“4”的字形编码
ACALL  DELAY                              ; 延时 1s
MOV    P2, #92DH                          ; 送“5”的字形编码
ACALL  DELAY                              ; 延时 1s
MOV    P2, #82H                            ; 送“6”的字形编码
ACALL  DELAY                              ; 延时 1s
MOV    P2, #0F8H                          ; 送“7”的字形编码
ACALL  DELAY                              ; 延时 1s
MOV    P2, #80H                            ; 送“8”的字形编码
ACALL  DELAY                              ; 延时 1s
MOV    P2, #90H                            ; 送“9”的字形编码
ACALL  DELAY                              ; 延时 1s
AJMP   MAIN                               ; 跳回到主程序

DELAY:                                    ; 延时 1s 程序
MOV    R5, #10

LI:    MOV    R6, #250

```

```

L2:   MOV    R7, #200
L3:   DJNZ   R7, L3
      DJNZ   R6, L2
      DJNZ   R5, L1
      RET
      END

```

强化训练

利用定时/计数器模拟汽车生产线产品件数。具体要求：以按键模拟产品检测，按一次键相当于产品计数一次；检测到的产品数送 P1 口显示，采用 1 位 LED 数码管显示，计满 16 次后从头开始，依次循环；系统采用 12MHz 晶振。

任务分析：

根据题意可设计出硬件电路图如图 4-6 所示。用单片机 P1 口与 1 位共阳极 LED 数码管相连接，P3.3 引脚连接按键 S。

程序流程图如图 4-7 所示。

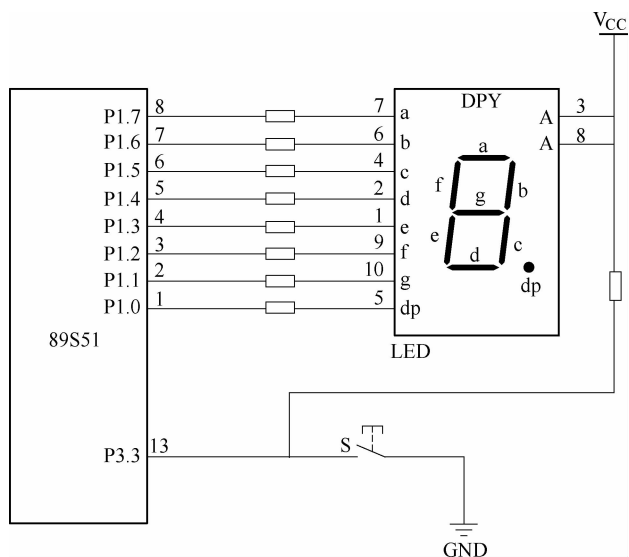


图 4-6 模拟汽车生产线产品件数数码管显示电路图

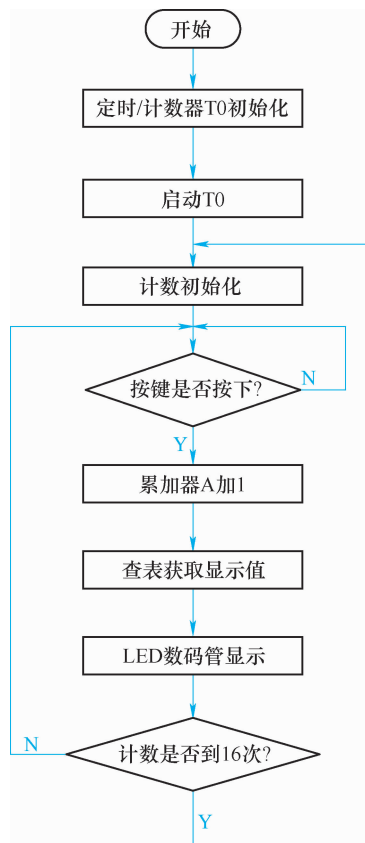


图 4-7 程序流程图

参考程序：

```

ORG    0000H
MOV    TMOD, #06H           ; 定时/计数器 T0 工作方式 2
MOV    TH0, #0F0H          ; T0 置初值
MOV    TL0, #0F0H
SETB   TR0                 ; 启动 T0
MAIN:  MOV    A, #00H        ; 计数显示初始化
      MOV    P1, #0C0H      ; LED 数码管显示 “0”
DSP:   JB    P3.3, DSP       ; 检测按键信号
      ACALL  DELAY          ; 消抖延时
      JB    P3.3, DSP       ; 确认低电平信号
DSP1:  JNB   P3.3, DSP1     ; 检测按键信号
      ACALL  DELAY          ; 消抖延时
      JNB   P3.3, DSP1     ; 确认高电平信号
      CLR   P3.4           ; T0 引脚产生负跳变
      NOP
      NOP
      SETB  P3.4           ; T0 引脚恢复高电平
      INC   A               ; 累加器加 1
      MOV   R1, A           ; 保存累加器计数值
      ADD  A, #08H         ; 变址调整
      MOVC A, @A + PC      ; 查表获取 LED 数码管显示值
      MOV  P1, A           ; LED 数码管显示查表值
      MOV  A, R1           ; 恢复累加器计数值
      JBC  TF0, MAIN       ; 查询 T0 计数溢出
      SJMP DSP             ; 16 次不到，继续计数
TAB:   DB    0C0H, 0F9H, 0A4H, 0B0H, 99H, 92H, 82H, 0F8H
      DB    80H, 90H, 88H, 83H, 0C6H, 0A1H, 86H, 8EH
DELAY: MOV   R2, #14H      ; 10ms 延时
DY1:   MOV   R3, #0FAH
      DJNZ  R3, $
      DJNZ  R2, DY1
      RET
      END

```

习 题

1. LED 数码管实际上由_____构成。
2. LED 数码管按照内部 LED 连接方式不同可分为_____和_____两种；按照显示亮度不同可分为_____和_____；按照显示字形不同可分为_____和_____。

3. 共阳极 LED 数码管的 8 个 LED 的_____全部连接在一起组成公共端, 8 个 LED 的_____各自独立引出, 使用时, 公共端接_____。
4. 共阴 LED 极数码管的 8 个 LED 的_____全部连接在一起组成公共端, 8 个 LED 的_____各自独立引出, 使用时, 公共端接_____。
5. 如要 LED 数码管显示“0”, 共阳极 LED 数码管的字形编码应为_____; 共阴极 LED 数码管的字形编码应为_____。
6. 如要 LED 数码管显示“一”, 共阳极数码管的字形编码应为_____; 共阴极 LED 数码管的字形编码应为_____。

任务二 LED 数码管静态显示设计



知识目标

1. 掌握 LED 数码管静态显示的原理。
2. 了解 LED 数码管静态显示的特点。
3. 掌握 74HC377 芯片的引脚及功能真值表。
4. 理解 74HC377 芯片的两种工作状态。



能力目标

1. 会使用 74HC377 芯片的锁存功能操作。
2. 能进行 LED 数码管显示的电路接口设计。
3. 能用单片机设计一个数字显示牌。
4. 能用单片机控制 8 位 LED 数码管实现“0~9”轮流显示。



任务相关知识

在单片机应用系统中, LED 数码管的显示方法有两种: 静态显示和动态显示。

1. LED 数码管静态显示原理

静态显示就是当显示器显示某个字符时, 相应的段恒定导通或截止, 直到显示另一个字符为止。

多位 LED 数码管工作于静态显示方式时, 各位 LED 数码管的共阴极 (或共阳极) 连接到仪器并接地 (或接 +5V); 每位 LED 数码管的段码线 (a~dp) 分别与一个单片机控制的 I/O 口锁存器输出相连接。如果送往各个 LED 数码管所显示字符的段码一经确定, 则相应 I/O 口锁存器锁存的段码输出将维持不变, 直到送入下一个显示字符的段码。因此, 本方法显示比较稳定, 亮度较高, 软件控制较容易。静态显示时每个数码管的每一个段码位都由一个单片机的 I/O 口驱动。当 N 位显示时, 则需 $8N$ 根输出控制线。

2. LED 数码管静态显示的特点

(1) 优点

显示程序十分简单, 显示亮度大, 由于 CPU 不必经常扫描显示器, 所以节约了 CPU 的工作时间。

(2) 缺点

占用的 I/O 口线较多，硬件电路复杂，成本也较高。所以静态显示法常用在显示位数较少的应用系统中。

3. LED 数码管驱动芯片

常用的地址锁存器芯片有 74HC373、74HC377、74HC573 等，下面以 74HC377 为例学习其应用。

(1) 74HC377 引脚及功能

74HC377 芯片是 8 位 D 型边沿触发器，芯片引脚图如图 4-8 所示，各引脚符号及名称见表 4-3。

(2) 74HC377 功能真值表

74HC377 功能真值表见表 4-4。

(3) 74HC377 的两种工作状态

1) 当 \bar{E} 为低电平、CP 为上升沿时，能把输入端的数据输出到输出端，实现数据输出。

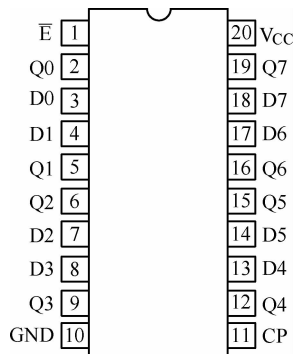


图 4-8 74HC377 引脚图

表 4-3 74HC573 引脚表

引脚号	引脚符号	引脚名称
1	\bar{E}	数据使能输入端（低电平有效）
2, 5, 6, 9, 12, 15, 16, 19	Q0 ~ Q7	数据输出端
3, 4, 7, 8, 13, 14, 17, 18	D0 ~ D7	数据输入端
11	CP	时钟输入端（上升沿触发）
10	GND	接地端
20	V _{cc}	电源端

表 4-4 74HC377 功能真值表

输 入			输 出
CP	\bar{E}	D _n	Q _n
↑	L	H	H
↑	L	L	L
↑	H	X	不变
X	H	X	不变

注：X = 不关心。

2) 当 \bar{E} 为高电平时，即使输入端数据变化，也不会影响输出端状态，实现锁存功能。

4. LED 数码管驱动电路

使用 74HC377 驱动 8 位 LED 数码管显示电路如图 4-9 所示。

使用两片 74HC377 作为 LED 数码管的驱动锁存芯片。其中，U1 的输出端 Q0 ~ Q7 接 8 位 LED 数码管的段选端，U2 的输出端 Q0 ~ Q7 接 8 位 LED 数码管的位选端，U1 和 U2 的输

入端 D0 ~ D7 连接在一起, 接单片机的 I/O 口。CS1 连接 U1 的 \bar{E} 端, 作为段选控制端; CS2 连接 U2 的 \bar{E} 端, 作为位选控制端; WR 连接 U1 和 U2 的 CP, 作为写入使能端。

通过单片机控制 CS1、CS2 和 WR 的电平, 能实现段选和位选的功能, 见表 4-5。

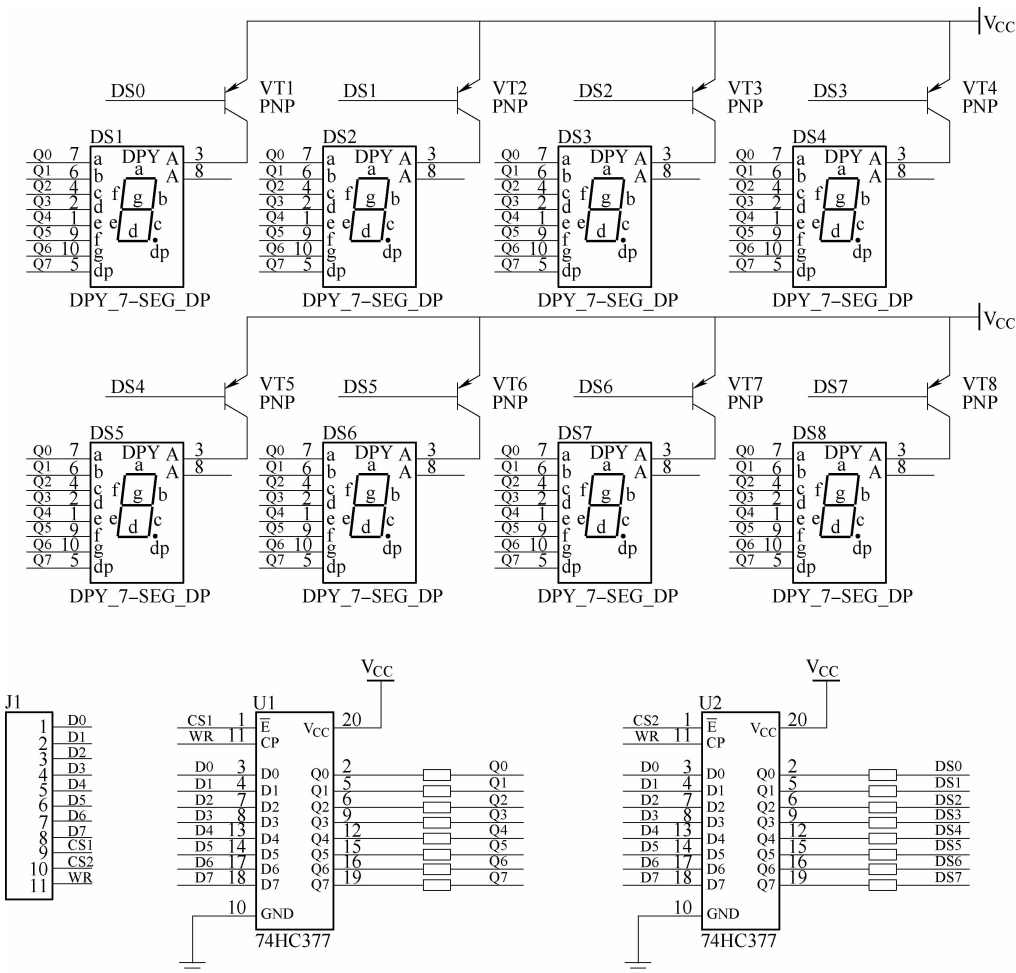


图 4-9 数码管显示电路图

表 4-5 段选和位选功能表

CS1	CS2	WR	U1	U2	功能
0	1	0→1	数据输出	数据锁存	写入段选码
1	0	0→1	数据锁存	数据输出	写入位选码

初步训练

利用单片机设计一个数字显示牌。具体要求: 8 个共阳极 LED 数码管同时显示任意一个数字或字符。

任务分析:

单片机的 P0 口连接两片 74HC377 的数据输入端 D7 ~ D0; P1.0 接 CS1, P1.1 接 CS2, P1.2 接 WR。

假设让 8 位 LED 数码管同时显示字符“0”，首先输出“0”段选码， $P0 = 0xC0H$ ，然后输出位选码 $P0 = 00H$ ，选中 8 位 LED 数码管。程序流程图如图 4-10 所示。

操作步骤：

- 1) 将单片机 V_{CC} 接 +5V，GND 接地， \overline{EA} 接 +5V。
- 2) 单片机的 P0 口接 74HC377 的 D7 ~ D0，P1.0 接 CS1，P1.1 接 CS2，P1.2 接 WR，打开电源。
- 3) 打开 Keil μ Vision4 软件，根据流程图进行编程，并调试通过，生成 HEX 文件。
- 4) 将生成的 HEX 文件下载到单片机中。
- 5) 观察 LED 数码管的现象，并记录在任务工单中。
- 6) 完成实训操作，关闭电源，拆除接线，整理归位。
- 7) 完成任务工单中相关题目。

参考程序：

```

CS1    EQU  P1.0    ; 定义段选端
CS2    EQU  P1.1    ; 定义位选端
WRR    EQU  P1.2    ; 定义数据使能端
ORG    0000H
AJMP   MAIN
ORG    0030H

MAIN:
CLR    CS1          ; CS1 置 0，选中段选端
SETB   CS2          ; CS2 置 1，关闭位选端
MOV    PO, #0C0H    ; 送“0”的段码
CLR    WRR          ; WR 上升沿输出数据
SETB   WRR
SETB   CS1          ; CS1 置 1，关闭段选端
CLR    CS2          ; CS2 置 0，选中位选端
MOV    PO, #00H     ; 送 8 位 LED 数码管的位选码
CLR    WRR          ; WR 上升沿输出数据
SETB   WRR
AJMP   MAIN
END

```

强化训练

利用单片机控制 8 位 LED 数码管实现“0~9”轮流显示。

任务分析：

硬件电路设计使用图 4-9 所示的电路，轮流显示采用延时程序实现，程序流程图如图 4-11 所示。

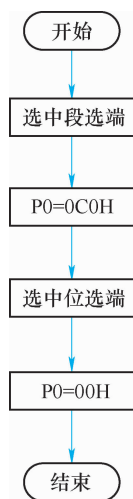


图 4-10 程序流程图

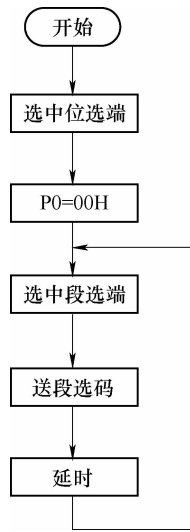


图 4-11 程序流程图

参考程序：

```

ORG    0000H
AJMP   0030H
ORG    0030H

MAIN:
MOV    DPTR, #TAB
MOV    R0, #00H
CLR    P2.1                ; 选中位选端
SETB   P2.0                ; 关闭段选端
MOV    P0, #00H           ; 送 8 位 LED 数码管的位选码
CLR    P2.2                ; 上升沿输出数据
SETB   P2.2

LOOP:
MOV    A, R0
MOVC   A, @ A + DPTR      ; 取显示的字形编码
CLR    P2.0                ; 选中段选端
SETB   P2.1                ; 关闭位选端
MOV    P0, A              ; 送字形编码
CLR    P2.2                ; 上升沿输出数据
SETB   P2.2
INC    R0
ACALL  DELAY
CJNE   R0, #0AH, LOOP     ; 判断显示是否到 10 次
MOV    R0, #00H           ; 到达 10 次, 重新开始
AJMP   LOOP

```

```

DELAY:                                ; 延时子程序
        MOV     R7, #05H
D2:     MOV     R5, #0FFH
D1:     MOV     R6, #0FFH
        DJNZ   R6, $
        DJNZ   R5, D1
        DJNZ   R7, D2
        RET
TAB:    DB     0C0H, 0F9H, 0A4H, 0B0H, 99H ; 共阳极 LED 数码管的段码
DB:     92H, 83H, 0F8H, 80H, 90H
END

```

习 题

1. LED 数码管静态显示就是当显示器显示某个字符时，相应的段恒定_____，直到显示另一个字符为止。
2. LED 数码管静态显示的优、缺点是什么？
3. 74HC377 各引脚功能是什么？
4. 74HC377 有哪两种工作状态？

◆ 任务三 LED 数码管动态显示设计

知识目标

1. 掌握 LED 数码管动态显示原理。
2. 掌握单片机控制 LED 数码管动态显示的电路接口设计和程序方法。
3. 了解 LED 数码管动态显示特点。

能力目标

1. 能用单片机进行日历显示系统的设计。
2. 能用单片机设计一个 LED 数码管显示的秒表。

任务相关知识

1. LED 数码管动态显示原理

动态显示即轮流向各位 LED 数码管送出字形编码和相应的位选码，各 LED 数码管分时轮流显示，利用发光二极管的余辉和人眼视觉暂留作用，使人感觉好像各位 LED 数码管同时都在显示。要使 LED 数码管稳定显示，必须采用扫描方式，即某一时刻只选通一位 LED 数码管，并送出相应的段选码，在另一时刻选通另一位 LED 数码管，并送出相应的段选码，依次循环。

为了使人眼看到稳定的显示，每个 LED 数码管刷新频率必须大于 25Hz，即每个 LED 数

码管相邻两次点亮的时间间隔要小于 40ms。

2. LED 数码管动态显示特点

(1) 优点

采用动态显示方式比较节省 I/O 口，适合显示位数较多的场合；硬件电路较静态显示方式简单，使用元器件少、硬件开销低；各 LED 数码管轮流显示，发光亮度和稳定度不如静态显示方式。

(2) 缺点

软件编程比较复杂，而且在显示位数较多时，CPU 要依次扫描，占用 CPU 较多时间。

初步训练

利用单片机设计一个显示年月日的日历显示系统，例如显示“17-03-20”。

任务分析：

利用图 4-9 所示的 LED 数码管显示电路，P2.0 接 CS1，P2.1 接 CS2，P2.2 接 WR，P0 口接 D0 ~ D7。首先让 P0 = 7FH，输出位选数据，选中第一位数码管；再输出第一位字符的段选数据，让 P0 = 0C0H，即显示“1”。延时 1ms，再依次点亮其他数码管，循环实现。位选信号的变化是有规律的，适宜采用移位指令控制，采用右移位指令“RR A”实现。段选数据采用查表指令控制，在单片机 ROM 定义一个表格 TABLE 存放“17-03-20”的字形编码，采用查表指令“MOVC A, @A + DPTR”实现。程序流程图如图 4-12 所示。

操作步骤：

- 1) 将单片机 V_{CC} 接 +5V，GND 接地，EA 接 +5V。
- 2) 将 P2.0 接 CS1，P2.1 接 CS2，P2.2 接 WR，P0 口接 D0 ~ D7，打开电源。
- 3) 打开 Keil μ Vision4 软件，根据流程图进行编程，并调试通过，生成 HEX 文件。
- 4) 将生成的 HEX 文件下载到单片机中。
- 5) 观察 LED 数码管显示现象，并记录在任务工单中。
- 6) 完成实训操作，关闭电源，拆除接线，整理归位。
- 7) 完成任务工单中相关题目。

参考程序：

```
CS1    EQU  P2.0
CS2    EQU  P2.1
WRR    EQU  P2.2
```

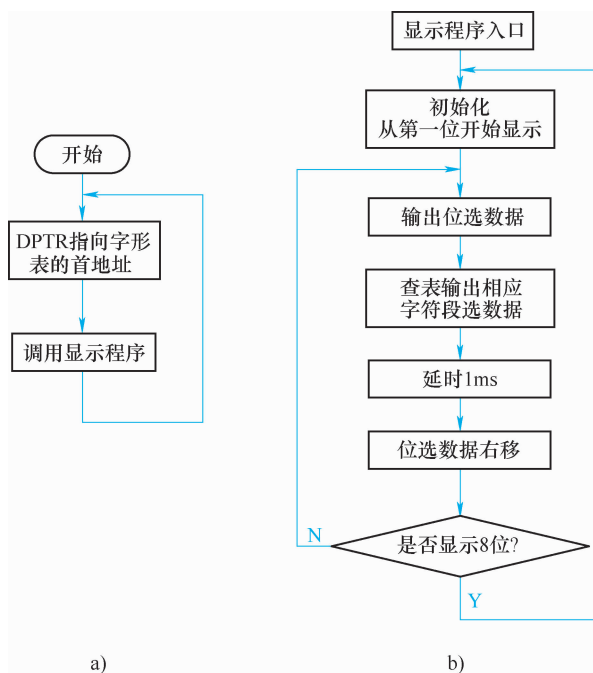


图 4-12 程序流程图

a) 主程序流程图 b) 显示程序流程图

```

        ORG    0000H
        LJMP   MAIN
        ORG    0030H
MAIN:    MOV    DPTR, #TABLE           ; DPTR 指向字形表的起始地址
        ACALL  SCAN                   ; 调用显示程序
        AJMP   MAIN
SCAN:    MOV    R1, #07FH             ; 从最一位数码管开始显示
        MOV    R2, #0A                ; 共显示 8 位
        MOV    R0, #0                 ; 字形编码的偏移地址
LOOP:    MOV    A, R0
        MOVC   A, @ A + DPTR          ; 取字形编码
        CLR    CS1                     ; 开段选控制
        SETB   CS2                     ; 关位选控制
        MOV    P0, A                   ; 送段选位
        CLR    WRR                     ; 传输数据
        SETB   WRR
        INC    R0                       ; 指向下一个字形码的偏移地址
        CLR    CS2                     ; 关段选控制
        SETB   CS1                     ; 开位选控制
        MOV    P0, R1                   ; 送位选位
        CLR    WRR                     ; 传输数据
        SETB   WRR
        ACALL  DELAY                   ; 调用 1ms 延时程序
        MOV    A, R1
        RR     A                         ; 右移一位
        MOV    R1, A
        DJNZ   R2, LOOP                 ; 判断 8 位是否显示完毕
        RET
DELAY:   MOV    R7, #02H               ; 延时子程序
D1:      MOV    R6, #0AH
        DJNZ   R6, $
        DJNZ   R7, D1
        RET
TABLE:   DB    0F9H, 0F8H, 0BFH, 0C0H
        DB    0B0, 0BFH, 0A4H, 0C0H ; “17-03-20” 的字形编码
        END

```

强化训练

利用单片机设计一个 LED 数码管显示的秒表。具体要求：使用 2 位 LED 数码管显示计时时间，最小计时单位为 s，计时范围为 1 ~ 59s。晶振频率为 12MHz。

任务分析：

本秒表显示采用图 4-9 所示的数码管显示电路中的最末两位数码管显示，采用单片机的定时/计数器 T0 工作方式 1 实现 1s 定时。两位数值的显示通过算法实现：首先把要显示的数值整除 10，商为十位字符，余数为个位字符。程序流程图如图 4-13 所示。

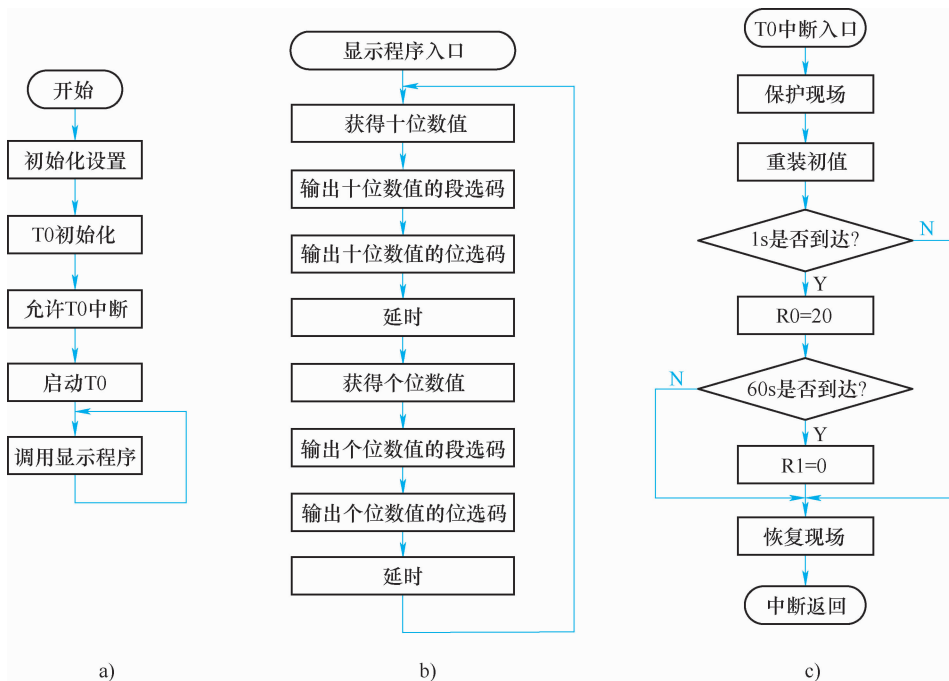


图 4-13 程序流程图

a) 主程序 b) 显示子程序 c) 中断服务子程序

参考程序：

可扫二维码获取。

习 题

1. 什么是 LED 数码管动态显示？
2. LED 数码管动态显示的优缺点是什么？

任务四 汽车里程表设计



知识目标

1. 理解汽车里程表的设计原理。
2. 理解光电晶体管的工作原理。
3. 回顾定时/计数器的定时和计数使用方法。



能力目标

1. 能用光电晶体管模拟检测车轮转动圈数。
2. 能用单片机定时/计数器进行定时和计数设计。
3. 能用单片机进行汽车里程表系统的设计。
4. 能用单片机进行显示运行时间的汽车里程表系统的设计。



任务相关知识

典型的机械式里程表连接一根软轴，软轴内有一根钢丝缆，软轴另一端连接到变速器某一个齿轮上，齿轮旋转带动钢丝缆旋转，钢丝缆带动里程表罩圈内一块磁铁旋转，罩圈与指针连接并通过游丝将指针置于零位，磁铁旋转速度的变化引起磁通密度的变化，平衡被打破，指针因此被带动。这种车速里程表简单实用，被广泛用在大小型汽车上。绝大多数轿车使用没有软轴的电子传感器车速表，常见的一种是从变速器上的速度传感器获取信号，通过脉冲频率的变化使指针偏转或者显示数字表示汽车的时速。汽车里程表还包括由连接同一信号源的两个液晶数字显示窗，分别累计本次里程和总里程。本次里程通常有四位数，供短期计数，可以清零；总里程则有六位数，不能清零。电子式里程表累积的里程数字存储在非易失性存储器内，在无电状态下数据也能保存。

里程表由蜗轮蜗杆机构和数字轮组成。汽车行驶时，主动轴经3对蜗轮蜗杆驱动里程表最右边的第一数字轮，使第一数字轮上的数字显示1/10km。从第一数字轮向左，每两个相邻的数字轮之间，又通过本身的内传动齿和进位数字轮传动齿轮，形成1:10的传动比。当第一数字轮转动一周，由9转到0时，由内传动齿拨动左侧第二数字轮转动1/10圈，形成1km数递增；当第二数字轮转动一周，由9转到0时，其左侧第三数字轮转动1/10，以10km数递增。其余数字轮由低位到高位显示、计数方式均依次类推，即可显示汽车行驶里程数。

初步训练

设计一款简易的汽车里程表系统。具体要求：利用4位LED数码管显示汽车里程数值，显示格式为“XX.X”，单位为km。假设车轮周长为2m。

任务分析：

1. 硬件电路设计

利用光电晶体管作为光电传感器检测车轮转动圈数。当光电晶体管未接收到光照而截止时，其集电极输出高电平，受光照而导通时其集电极输出低电平，产生一个负脉冲，模拟车

轮转动一圈。利用单片机的定时/计数器 T0，对脉冲信号进行采样计数，得到车轮的转动圈数，已知车轮周长，通过计算得到汽车里程数值。

为了防止计数过程中外界信号的干扰，该脉冲经施密特触发器整形后，接到单片机定时/计数器 T0 的输入端，通过图 4-9 所示的 4 位 LED 数码管 DS7 ~ DS4 显示汽车的里程数。根据以上分析，电路设计如图 4-14 所示。

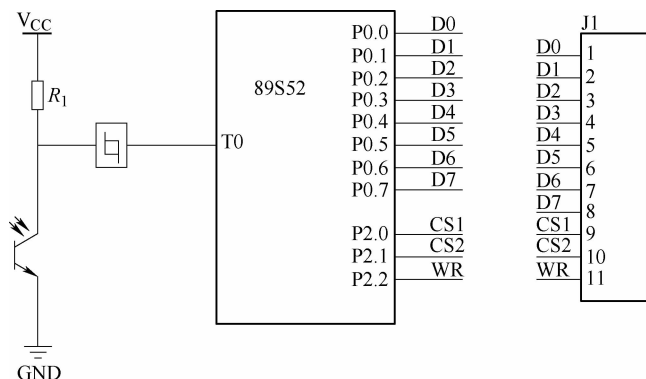


图 4-14 汽车里程表电路图

检测脉冲的计数输入端接到单片机定时/计数器 T0 引脚（P3.4），P0 口接 LED 数码管显示模块的数据端口 D0 ~ D7，P2.0 ~ P2.2 接控制端口 CS1、CS2 和 WR。

2. 软件设计

汽车里程数值显示格式为“XX.Xkm”，则里程数值的最小单位为 100m，车轮周长为 2m，车轮转动 50 圈时，显示数值加 1。

利用 T0 作计数器，每计数 50 次计数器溢出，里程数值 R0 加 1。T0 采用工作方式 2，则 TMOD = 06H，计数初值 = 28 - 50 = 0CEH。程序流程图如图 4-15 所示。

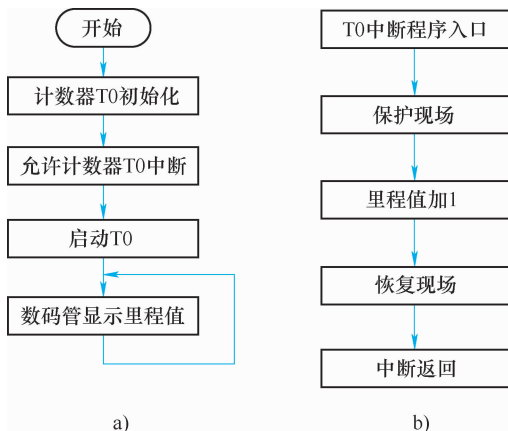


图 4-15 程序流程图

a) 主程序 b) 中断服务子程序

操作步骤：

- 1) 将单片机 V_{CC} 接 +5V, GND 接地, \overline{EA} 接 +5V。
- 2) 将 P3.4 接光电检测电路, P0 口接 LED 数码管显示模块的数据端口 D0 ~ D7, P2.0 ~ P2.2 接控制端口 CS1、CS2 和 WR, 打开电源。
- 3) 打开 Keil μ Vision4 软件, 根据流程图进行编程, 并调试通过, 生成 HEX 文件。
- 4) 将生成的 HEX 文件下载到单片机中。
- 5) 观察 LED 数码管现象, 并记录在任务工单中。
- 6) 完成实训操作, 关闭电源, 拆除接线, 整理归位。
- 7) 完成任务工单中相关题目。

参考程序：

```

        CS1      EQU P2.0                ; 定义 LED 数码管的控制端口
        CS2      EQU P2.1
        WRR      EQU P2.2
        ORG      0000H
        AJMP     MAIN
        ORG      000BH
        AJMP     INT_T0                 ; T0 计数中断服务程序
        ORG      0030H
MAIN:    MOV     SP, #60H                ; 定义堆栈指针地址
        MOV     R0, #00H                ; 里程数初值为 0
        MOV     DPTR, #TAB              ; DPTR 存入字形编码表格首地址
        MOV     TMOD, #06H              ; T0 计数模式, 工作方式 2
        MOV     TH0, #0CEH              ; 设置 T0 初值
        MOV     TLO, #0CEH
        MOV     IE, #82H                ; T0 允许中断
        SETB    TR0                     ; 启动 T0
LOOP:    ACALL   DISP                   ; 调用 LED 数码管显示子程序
        AJMP    LOOP
DISP:    MOV     A, R0                   ; LED 数码管显示子程序
        MOV     B, #100                  ; 取里程数的最高位
        DIV     AB
        MOVC    A, @ A + DPTR           ; 取字符字形编码
        CLR     CS1
        SETB    CS2
        MOV     P0, A
        CLR     WRR
        SETB    WRR
        SETB    CS1
        CLR     CS2

```

```

MOV    P0, #07FH           ; 设置显示在第一位数码管
CLR    WRR
SETB   WRR
ACALL  DELAY
MOV    A, B                 ; 取里程数值的第二位并显示
MOV    B, #10
DIV    AB
MOVC   A, @ A + DPTR
CLR    CS1
SETB   CS2
MOV    P0, A
CLR    WRR
SETB   WRR
SETB   CS1
CLR    CS2
MOV    P0, #0BFH
CLR    WRR
SETB   WRR
ACALL  DELAY
MOV    A, #10              ; 数码管第三位显示小数点
MOVC   A, @ A + DPTR
CLR    CS1
SETB   CS2
MOV    P0, A
CLR    WRR
SETB   WRR
SETB   CS1
CLR    CS2
MOV    P0, #0DFH
CLR    WRR
SETB   WRR
ACALL  DELAY
MOV    A, B                 ; 取里程数值的最末位显示
MOVC   A, @ A + DPTR
CLR    CS1
SETB   CS2
MOV    P0, A
CLR    WRR
SETB   WRR

```

```

SETB    CS1
CLR     CS2
MOV     P0, #0FEH
CLR     WRR
SETB    WRR
ACALL   DELAY
RET
DELAY:  MOV    R6, #05FH           ; 延时子程序
D1:     MOV    R5, #0FFH
        DJNZ  R5, $
        DJNZ  R6, D1
        RET
INT_T0: PUSH   ACC                ; T0 中断服务程序
        PUSH  PSW
        PUSH  DPH
        PUSH  DPL
        INC   R0                  ; 计数 50, R0 加 1
        POP   DPH
        POP   DPL
        POP   PSW
        POP   ACC
        RETI
TAB:    DB     0C0H, 0F9H, 0A4H, 0B0H, 99H
        DB     92H, 83H, 0F8H, 80H, 90H, 7FH
        END

```

强化训练

设计一款显示运行时间的汽车里程表系统。具体要求：在初步训练要求的基础上，利用 4 位 LED 数码管显示汽车的运行时间。晶振频率为 12MHz。

任务分析：

硬件设计电路用图 4-14 所示电路，LED 数码管显示电路采用如图 4-9 所示电路，其中 DS7 ~ DS4 显示汽车的里程数，DS3 ~ DS0 显示运行时间（DS3、DS2 显示分钟数，DS1、DS0 显示秒数）。

此处 T0 用作计数器工作，采用工作方式 2，T0 每计数 50 个，计满溢出，申请中断，里程数值 R0 加 1。采用定时器 T1 工作方式 1 定时 50ms，用 R1 作 50ms 计数单元，实现 1s 定时。R2 存放分钟数，R3 存放秒数。

工作方式控制器 TMOD = 16H，T1 计数初值 = $65536 - 50000 = 15536 = 3CB0H$ 。

LED 数码管显示的 8 位字符存放在 30H ~ 37H 单元，通过查表指令获得字符段码，显示位码通过移位指令实现。

根据以上分析，程序流程图如图 4-16 所示。

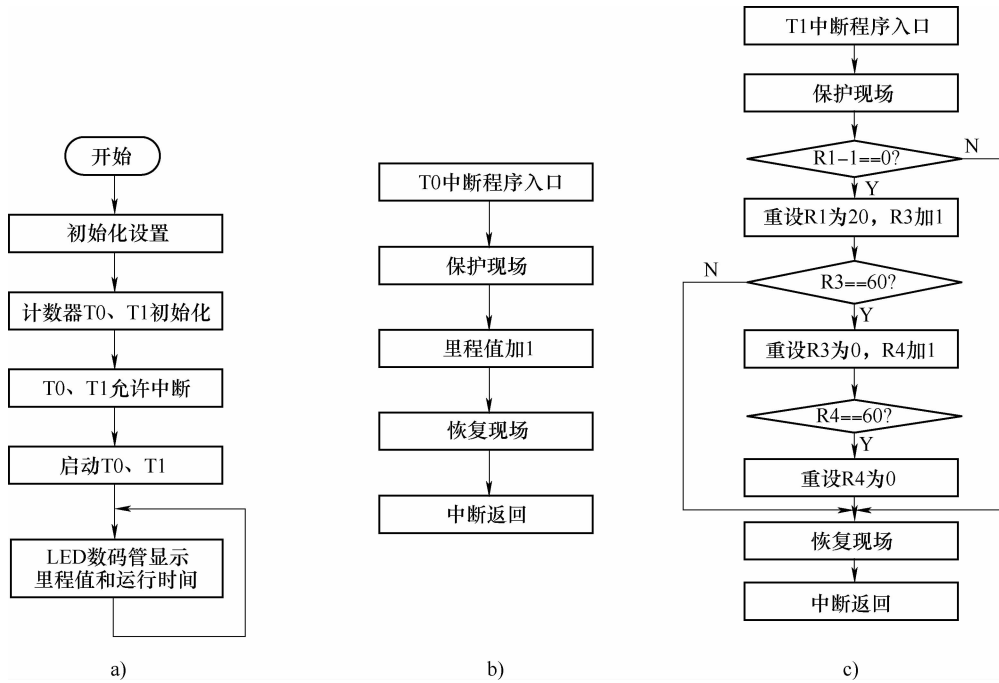


图 4-16 程序流程图

a) 主程序 b) T0 计数中断服务子程序 c) T1 定时中断服务子程序

参考程序：

可扫二维码获取。

项目二 汽车转速表

任务一 LCD 显示设计

知识目标

1. 理解 LCD1602 的引脚功能。
2. 掌握 LCD1602 的基本操作时序。
3. 掌握 LCD1602 的 11 条控制指令。
4. 掌握 LCD1602 的 DDRAM 的地址映射关系。

能力目标

1. 能用 LCD1602 进行显示模式设置。
2. 能用 LCD1602 进行字符显示位置的设置。
3. 能用 LCD1602 进行初始化设置。
4. 能进行 LCD1602 简单显示的设计。
5. 能进行 LCD1602 的移位显示设计。

任务相关知识

LCD 为平面超薄的显示器，由一定数量的彩色或黑白像素组成，放置于光源或者反射面前方。

LCD 有字段型、字符型、点阵型，在使用时，有的 LCD 内部有控制器，使用比较方便。

液晶显示模块具有体积小、功耗低、显示内容丰富、超薄轻巧等优点，在单片机应用领域得到越来越广泛的应用。单片机应用系统中最常见的 LCD 有 LCD12864、LCD1602 和 LCD3310 等。LCD1602 是最常见的字符型显示器。

1. LCD1602 简介

LCD1602 是一种工业字符型液晶显示模块，能够同时显示 16×02 （即 32）个字符（16 行 \times 2 列），是专门用于显示字幕、数字、符号等的点阵型液晶显示模块。采用 +5V 电源供电，外围电路配置简单、价格便宜，具有很高的性价比。

LCD1602 内部具有字符库 ROM（CGROM），能显示出 192 个字符。每一个字符都有一个固定的代码，其中数字和字母的代码采用 ASCII 码编码。单片机控制 LCD1602 显示字符时，只需将待显示字符的 ASCII 码写入内部的显示用数据存储器（DDRAM），内部控制电路就可将字符显示在显示器上。例如，要显示字符“A”，单片机只需将字符“A”的 ASCII 码（41H）写入 DDRAM，控制电路就会将对应的字符库 ROM（CGROM）中字符“A”的点阵数据找出来显示在 LCD 上。

2. LCD1602 引脚控制

LCD1602 的工作电压为 4.5 ~ 5.5V，典型工作电压为 5V，工作电流为 2mA。LCD1602 (16 引脚，有背光) 的外形及引脚图如图 4-17 所示。

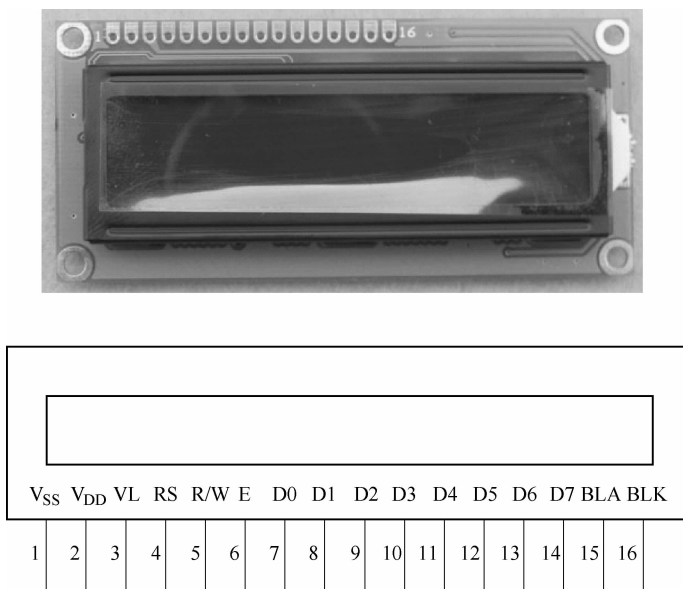


图 4-17 LCD1602 的外形及引脚图

引脚包括 8 个数据端、3 个控制端、3 个电源端和 2 个背光源端，各引脚的功能见表 4-6。

表 4-6 LCD1602 引脚功能表

引脚标号	引脚名称	引脚功能
1	V_{SS}	电源地
2	V_{DD}	电源正极
3	VL	液晶显示器偏压；调节液晶屏对比度
4	RS	数据/命令选择： RS = 0，将 D0 ~ D7 上的数据送到数据寄存器 RS = 1，将 D0 ~ D7 上的数据送到指令寄存器
5	R/W	读/写选择： R/W = 0，进行写操作 R/W = 1，进行读操作
6	E	使能信号： 读操作时，下降沿有效 写操作时，高电平有效
7 ~ 14	D0 ~ D7	数据输入/输出口
15	BLA	背光源正极
16	BLK	背光源负极

3. LCD1602 的基本操作时序

LCD1602 共有四种基本操作，每一种操作都有自己的操作控制时序，见表 4-7。

表 4-7 LCD1602 的基本操作控制时序

基本操作	输 入				输 出
	RS	R/W	E	D0 ~ D7	
读状态	0	1	1		D0 ~ D7 = 状态字
写指令	0	0	高脉冲	指令码	无
读数据	1	1	1		D0 ~ D7 = 数据
写数据	1	0	高脉冲	数据	无

4. LCD1602 显示模式设置

LCD1602 液晶模块内部的控制器共有 11 条控制指令。它的读写操作、屏幕和光标的操作都是通过指令编程来实现的。各指令的功能见表 4-8。

表 4-8 LCD1602 控制指令表

序号	控制指令	RS	R/W	D7	D6	D5	D4	D3	D2	D1	D0
1	清显示	0	0	0	0	0	0	0	0	0	1
2	光标返回	0	0	0	0	0	0	0	0	1	*
3	置输入模式	0	0	0	0	0	0	0	1	I/D	S
4	显示开关控制及光标设置	0	0	0	0	0	0	1	D	C	B
5	光标字符移位	0	0	0	0	0	1	S/C	R/L	*	*
6	置功能	0	0	0	0	1	DL	N	F	*	*
7	置字符存储器地址	0	0	0	1	字符发生存储器地址					
8	置数据存储器地址	0	0	1	显示数据存储器发生地址						
9	读忙标志或地址	0	1	BF	计数器地址						
10	写数据到 CGRAM 或 DDRAM	1	0	要写的的数据							
11	从 CGRAM 或 DDRAM 读数据	1	1	读出的数据							

注：1 为高电平，0 为低电平。

指令说明：

指令 1：清显示，指令码为 01H，光标复位到地址 00H 位置。

指令 2：光标复位，光标返回地址 00H。

指令 3：光标和显示模式设置。

I/D：光标移动方向，高电平右移，低电平左移。

S：屏幕上所有文字是否左移或者右移。高电平表示有效，低电平则无效。

指令 4：显示开关控制及光标设置。

D：控制整体显示的开与关，高电平表示开显示，低电平表示关显示。

C：控制光标的开与关，高电平表示有光标，低电平表示无光标。

B：控制光标是否闪烁，高电平表示闪烁，低电平表示不闪烁。

指令 5：光标或显示移位。

S/C：高电平时移动显示的文字，低电平时移动光标。

R/L: 高电平时光标向右移动, 低电平时光标向左移动。

指令 6: 功能设置命令。

DL: 高电平时为 4 位总线, 低电平时为 8 位总线。

N: 低电平时为单行显示, 高电平时双行显示。

F: 低电平时显示 5×7 的点阵字符, 高电平时显示 5×10 的点阵字符。

指令 7: 字符发生器 RAM 地址设置。

指令 8: DDRAM 地址设置。

指令 9: 读忙信号和光标地址。

BF: 为忙标志位, 高电平表示忙, 此时模块不能接收命令或者数据, 低电平表示不忙。

指令 10: 写数据。

指令 11: 读数据。

例如, 将 LCD1602 显示模式设置为“ 16×2 显示, 5×7 点阵, 8 位数据接口”, 只需要向 LCD1602 写入置功能指令 (指令 6) “00111000B” (即 38H) 即可。

例如, 要求 LCD 开显示, 显示光标且光标闪烁, 则根据显示开关控制及光标设置指令 (指令 4), 令 $D=1$, $C=1$, $B=1$, 写入命令 “00001111B” (即 0FH), 就可实现所需的显示模式。

5. LCD1602 字符显示位置的确定

LCD1602 的控制器内部带有 80KB 的 DDRAM, 用来寄存待显示的字符代码, 其与显示屏的字符显示位置是一一对应的。LCD1602 的 DDRAM 地址与字符显示位置的对应关系如图 4-18 所示。

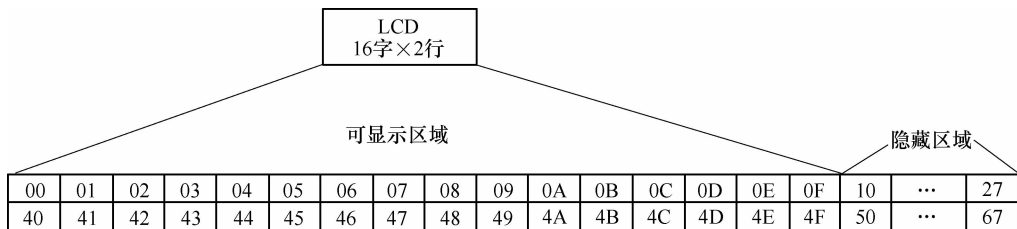


图 4-18 LCD1602 的 DDRAM 地址映射

当向 DDRAM 的 00H~0FH (第 1 行)、40H~4FH (第 2 行) 地址中的任意一处写入数据时, LCD 将立即显示出来, 该区域称为可显示区域。当写入 10H~27H 或 50H~67H 地址中时, 字符不会显示出来, 该区域称为隐藏区域。要显示写入隐藏区域的字符, 只需通过光标字符移位指令 (指令 5) 将它们移入可显示区域即可。

在向 DDRAM 写入字符时, 首先要设置 DDRAM 地址, 此操作可通过指令 8 完成。例如要写入字符到 DDRAM 的 40H 处, 则指令 8 的格式为: $80H + 40H = C0H$, 其中 80H 为命令代码, 40H 为写入字符的地址。

6. LCD1602 初始化设置

- 1) 写命令 38H, 设置显示模式为 16×2 显示、 5×7 点阵、8 位数据接口。
- 2) 写命令 08H, 显示关闭。
- 3) 写命令 01H, 显示清屏, 数据指针清 0。

4) 写命令 0CH, 设置显示开及光标不显示。

5) 写命令 06H, 设置光标右移, 即写一个字符后地址指针加 1。

需要说明, LCD1602 是慢显示器, 在进行设置和数据读取时, 需要检测忙标志位 BF, 如果为 1, 则说明忙, 需要等待; 如果为 0, 则可进行下一步操作。实际上, 由于单片机的操作速度慢于显示器的反应速度, 可以不进行忙检测, 或只进行简短延时即可。

初步训练

利用 LCD1602 显示 “Dian Zi Xi”。

任务分析:

利用单片机 P0 口接 D0 ~ D7, P2.5 接 RS, P2.6 接 R/W, P2.7 接 E。画出电路图, 如图 4-19 所示。

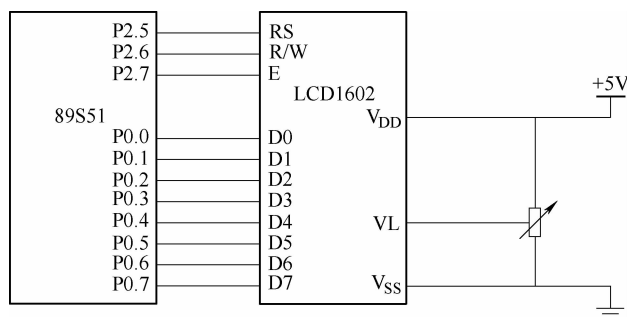


图 4-19 LCD1602 显示电路图

程序流程图如图 4-20 所示。

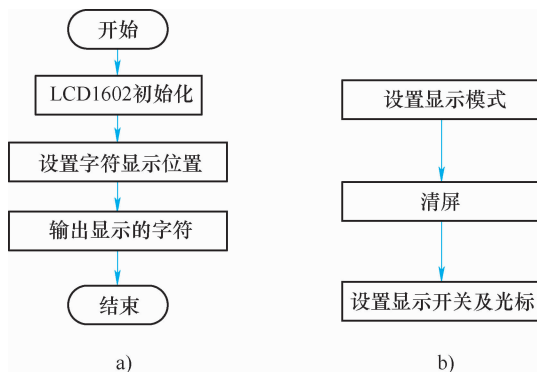


图 4-20 程序流程图

a) 主程序 b) LCD1602 初始化子程序

操作步骤:

- 1) 将单片机 V_{CC} 接 +5V, GND 接地, \overline{EA} 接 +5V。
- 2) 将 P0 口接 D0 ~ D7, P2.5 接 RS, P2.6 接 R/W, P2.7 接 E, 打开电源。
- 3) 打开 Keil μ Vision4 软件, 根据流程图进行编程, 并调试通过, 生成 HEX 文件。
- 4) 将生成的 HEX 文件下载到单片机中。

- 5) 观察 LCD1602 显示现象, 并记录在任务工单中。
- 6) 完成实训操作, 关闭电源, 拆除接线, 整理归位。
- 7) 完成任务工单中相关题目。

参考程序:

```

        RS      EQU P2.5          ; LCD1602 数据/命令选择端口
        RW      EQU P2.6          ; LCD1602 读/写选择端口
        E       EQU P2.7          ; LCD1602 使能端口
        ORG     0000H
        AJMP    START
        ORG     0030H

START:
        ACALL   LCD_INIT          ; LCD1602 初始化
        MOV     A, #80H           ; 设置显示位置在第 1 行第 1 列
        ACALL   WRITE1
        MOV     DPTR, #TABLE
        ACALL   WRITEST          ; 显示字符串
LOOP:   AJMP    LOOP

LCD_INIT:
        MOV     A, #38H          ; 8 位两行显示, 5×7 点阵字符
        ACALL   WRITE1
        MOV     A, #08H          ; 显示关闭
        ACALL   WRITE1
        MOV     A, #01H          ; 清屏
        ACALL   WRITE1
        MOV     A, #0CH          ; 开显示, 光标关闭
        ACALL   WRITE1
        MOV     A, #06H          ; 光标右移
        ACALL   WRITE1
        RET

WRITE1:
        CLR     RS
        CLR     RW
        CLR     E
        MOV     P1, ACC
        SETB    E                ; E 高脉冲
        NOP
        CLR     E
        RET

WRITE2:
        ; 写数据子程序

```

```

SETB   RS
CLR    RW
CLR    E
MOV    P1, ACC
SETB   E                ; E 高脉冲
NOP
CLR    E
RET
WRITEST:                ; 写字符串子程序
MOV    R1, #00H
A1:    MOV    A, R1
        MOVC  A, @ A + DPTR
        ACALL WRITE2
        INC   R1
        CJNE  A, #00H, A1
        RET
TABLE:  DB "Dian Zi Xi"
        END

```

强化训练

进行 LCD1602 的移位显示设计。

任务分析:

首先进行第一行数据 DDRAM 地址设置，第一行数据的隐藏区域地址为 10H，指令代码为 $80H + 10H = 90H$ ，再写入数据。同样，第二行数据的地址为 50H，先写入地址命令代码 0D0H，再写入第二行数据。然后使用整屏左移指令 18H，每 125ms 移动一个地址，共移动 16 个地址，正好将要显示的内容全部移到可显示区域。根据以上分析，程序流程图如图 4-21 所示。

参考程序:

可扫二维码获取。

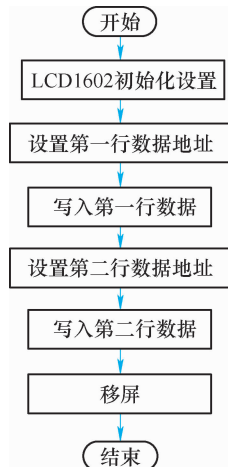


图 4-21 程序流程图

习 题

1. RS 引脚功能是_____，当 RS = _____时，将 D0 ~ D7 上的数据送到数据寄存器；当 RS = _____时，将 D0 ~ D7 上的数据送到指令寄存器。
2. R/W 引脚功能是_____，当 R/W = _____时，进行写操作；当 R/W = _____时，进行读操作。

3. E 引脚功能是_____，读操作时，_____有效；写操作时，_____有效。
4. 将 LCD1602 显示模式设置为“16 × 2 显示、5 × 10 点阵、4 位数据接口”，需要向 LCD1602 写入的置功能指令为_____。
5. 要求 LCD 开显示，显示光标且光标闪烁，显示开关控制及光标设置指令为_____。
6. 什么是 DDRAM？如何设置？
7. LCD1602 的基本操作时序如何设置？

任务二 汽车转速表设计

知识目标

1. 理解简单电子转速表工作原理。
2. 掌握 LCD1602 显示原理。
3. 掌握霍尔式传感器的工作原理。
4. 掌握霍尔式传感器 A3144 的引脚和内部电路。

能力目标

1. 学会霍尔式传感器 A3144 的使用。
2. 能用 LCD1602 进行速度显示。
3. 能进行简单的电子转速表的设计。

任务相关知识

转速表安装在汽车仪表板内，单位是 1000r/min，即显示发动机每分钟转多少千转，能够直观地显示发动机在各个工况下的转速，驾驶人可以随时知道发动机的运转情况，配合变速器档位和油门位置，使之保持最佳的工作状态，减少油耗，延长发动机寿命。

霍尔式传感器是根据霍尔效应制成的一种磁场传感器。将置于磁场中的导体或半导体通入电流，若电流与磁场垂直，则在磁场和电流都垂直的方向上产生一个电势差，这种现象就是霍尔效应。利用霍尔效应制成的元件称为霍尔式传感器。

霍尔式传感器 A3144 是一种单极（S）开关型传感器，引脚和内部电路如图 4-22 所示。只要 S 极有“靠近-远离”的动作，A3144 就产生脉冲信号。需要注意，输出端为 OC 门，使用时需要加上拉电阻。

初步训练

设计一款简单电子转速表。具体要求：利用霍尔式传感器 A3144 检测车轮转速，用 LCD1602 显示转速值。

任务分析：

1. 硬件设计

以霍尔式传感器 A3144 最为输入器件，然后利用单片机的定时/计数器 T0 的计数输入

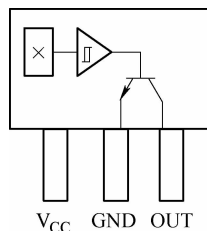


图 4-22 A3144 引脚和内部电路

端,对传感器输入的脉冲进行计数,即可很方便地完成对转速的测量。图 4-23 所示为电子转速表电路图。输入主要由霍尔传感器 A3144 构成,经输出驱动后接至单片机的定时/计数器 T0 计数输入端 T0 引脚(P3.4),输出由 LCD1602 来显示转速值。

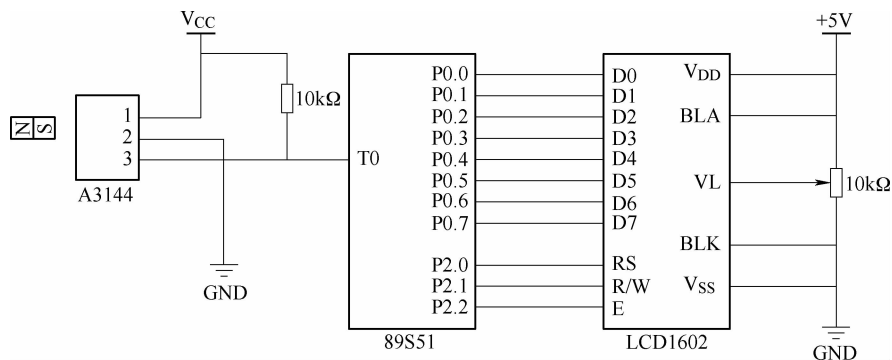


图 4-23 电子转速表电路图

2. 软件设计

转速表的转速测量采用直接测量法。由定时/计数器 T1 实现 1min 定时,由定时/计数器 T0 完成对转速的测量,测量范围为 0~255r/min。程序流程图如图 4-24 所示。

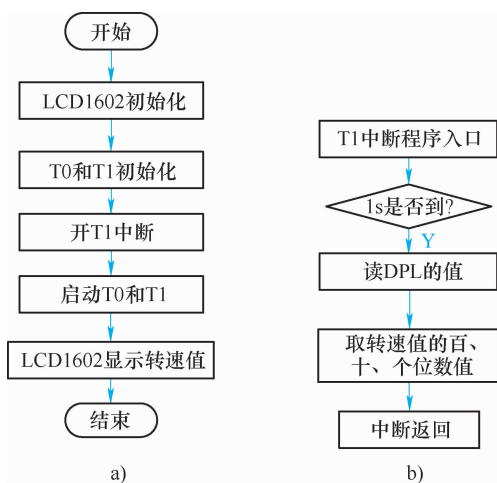


图 4-24 程序流程图

a) 主程序 b) 中断服务程序

操作步骤:

- 1) 将单片机 V_{CC} 接 +5V, GND 接地, \overline{EA} 接 +5V。
- 2) 将 P0 口接 D0~D7, P2.0 接 RS, P2.1 接 R/W, P2.2 接 E, T0 引脚(P3.4) 接霍尔式传感器 A3144, 打开电源。
- 3) 打开 Keil μ Vision4 软件, 根据流程图进行编程, 并调试通过, 生成 HEX 文件。
- 4) 将生成的 HEX 文件下载到单片机中。
- 5) 观察 LCD1602 显示现象, 并记录在任务工单中。

6) 完成实训操作, 关闭电源, 拆除接线, 整理归位。

7) 完成任务工单中相关题目。

参考程序:

```

    RS      EQU P2.0           ; 定义 LCD1602 使能端口
    RW      EQU P2.1
    E       EQU P2.2
    ORG     0000H
    AJMP    START
    ORG     001BH
    AJMP    SPEED             ; 跳转 T1 中断服务程序
    ORG     0030H

START:
    MOV     31H, #00H         ; 存放转速值的百位值
    MOV     32H, #00H         ; 存放转速值的十位值
    MOV     33H, #00H         ; 存放转速值的个位值
    ACALL   LCD_INIT         ; 调用 LCD1602 初始化子程序
    MOV     TMOD, #15H        ; T0 计数工作方式 1, T1 定时工作方式 1
    MOV     TH0, #00H         ; T0 计数初值设置
    MOV     TL0, #00H
    MOV     TH1, #0CH         ; T1 定时 125ms 初值设置
    MOV     TL1, #40H
    MOV     R0, #08H          ; 软件计数器 R0 设为 8, 定时 1s
    MOV     R1, #3CH          ; 软件计数器 R1 设为 3C, 定时 1min
    MOV     IE, #88H          ; 定时/计数器 T1 中断允许
    SETB    TR0               ; 启动 T0
    SETB    TR1               ; 启动 T1
LOOP:   ACALL LCD_DIS         ; 调用 LCD1602 显示子程序
        SJMP    LOOP

LCD_INIT:                       ; LCD1602 初始化设置子程序
    MOV     A, #38H
    ACALL   WRITE1
    MOV     A, #08H
    ACALL   WRITE1
    MOV     A, #01H
    ACALL   WRITE1
    MOV     A, #0CH
    ACALL   WRITE1
    MOV     A, #06H
    ACALL   WRITE1

```

```

        RET
WRITE1: CLR    RS                ; LCD1602 写指令子程序
        CLR    RW
        CLR    E
        MOV    P1, ACC
        SETB   E
        NOP
        CLR    E
        RET
WRITE2:                ; LCD1602 写数据子程序
        SETB   RS
        CLR    RW
        CLR    E
        MOV    P1, ACC
        SETB   E
        NOP
        CLR    E
        RET
LCD_DIS:                ; LCD1602 显示子程序
        MOV    A, #80H          ; 设置字符在第一行第一列开始显示
        ACALL WRITE1
        MOV    DPTR, #TABLE
        MOV    A, 31H           ; 取转速值百位数值
        MOVC   A, @ A + DPTR    ; 显示百位数的字符
        ACALL WRITE2
        MOV    DPTR, #TABLE
        MOV    A, 32H           ; 取转速值十位数值
        MOVC   A, @ A + DPTR    ; 显示十位数的字符
        ACALL WRITE2
        MOV    DPTR, #TABLE
        MOV    A, 33H           ; 取转速值个位数值
        MOVC   A, @ A + DPTR    ; 显示个位数的字符
        ACALL WRITE2
        ACALL DELAY            ; 延时
        RET
SPEED:                ; T1 中断服务子程序
        DJNZ   R0, DONE2        ; 1s 未到
        DJNZ   R1, DONE1        ; 1min 未到
        CLR    TR0              ; 1min 到, 关闭 T0

```



```

CLR    TR1                ; 关闭 T1
MOV    A, TL0             ; 取转速值
MOV    B, #100           ; 取转速值百位值
DIV    AB
MOV    31H, A             ; 保存百位值到 31H
MOV    A, B
MOV    B, #10             ; 取转速值百位和个位值
DIV    AB
MOV    32H, A             ; 保存十位值到 32H
MOV    33H, B             ; 保存个位值到 33H
MOV    TH0, #00H         ; 重新设置 T0 的计数初值
MOV    TL0, #00H
MOV    TH1, #0CH         ; 重新设置 T1 的定时初值
MOV    TL1, #40H
MOV    R0, #08H
MOV    R1, #3CH
SETB   TR0                ; 重新启动 T0 和 T1
SETB   TR1
RETI
DONE1: MOV    R0, #08H    ; 1s 未到, 重新设置软件计数 R0
DONE2: MOV    TH1, #0CH  ; 1min 未到, 重新设置 T1 定时初值
      MOV    TL1, #40H
      RETI
DELAY: ; 延时子程序
      MOV    R6, #250
DD1:   MOV    R7, #249
DD2:   DJNZ   R7, DD2
      DJNZ   R6, DD1
      RET
TABLE: DB    " 0123456789"
      END

```

习 题

1. 霍尔式传感器是根据_____制成的一种磁场传感器。
2. 霍尔式传感器 A3144 是一种_____传感器, 只要 S 极有“靠近-远离”的动作, 就产生_____。需要注意, 输出为 OC 门, 使用时需要_____。
3. 霍尔式传感器 A3144 的三个引脚分别是_____、_____和_____。
4. 什么是霍尔效应?
5. 霍尔式传感器的测速原理是什么?

项目三 点阵 LED 控制

任务 点阵 LED 显示设计

知识目标

1. 理解点阵显示屏的组成。
2. 掌握点阵显示屏的显示原理。
3. 掌握 LED 点阵控制器的控制原理。

能力目标

1. 能写出点阵显示编码。
2. 能完成点阵显示器与单片机的硬件电路连接。
3. 能利用 4 块 8×8 LED 点阵合并而成一个 16×16 显示屏。

任务相关知识

随着 LED 应用领域的扩大，要求生产更为直接和方便的 LED 显示器件。因而出现了数码管、字符管、电平管、LED 点阵等多种 LED 显示器。无论显示器的结构如何变，它的核心部件仍然是发光半导体芯片。

例如，一个 8×8 LED 点阵是由 64 个 LED 按一个规律组成的，如图 4-25 所示。

图 4-25b 所示的 LED，行接低电平，列接高电平，LED 导通发光。

根据 LED 显示屏的显示原理及结构，分析 LED 点阵控制器的控制原理。

如显示 10 个汉字，先将这 10 个汉字的点阵从字库中读出，放到显示缓存中，如果要实现左移或者其他的显示效果，则将显示缓存中的每个位进行移位或者其他处理，然后再调用扫描显示函数就可以实现所规定的效果。

扫描显示函数是显示缓存的内容，如果要实现不同的内容，比如说图片、汉字、英文等内容，只需要将这些内容按扫描显示函数对显示缓存的协议要求对应起来，就可以显示出来。

由于 LED 显示屏中采用 74HC595 移位寄存器，所以需要设计 74HC595 的驱动，这个相对简单，只要将数据按位传输，一位送一个时钟，送完一行所有的数据送一个锁存时钟，再通过 74HC138 选通该行，这样一直循环，人眼就会看到一幅完整的内容。

对于 LED 显示屏来说，显示使能端是比较重要的，主要是因为送完一行后需要有一个消隐的动作。所谓消隐，就是让显示屏黑屏一段时间，如果不做该动作，则在显示的过程中会有拖影的现象。

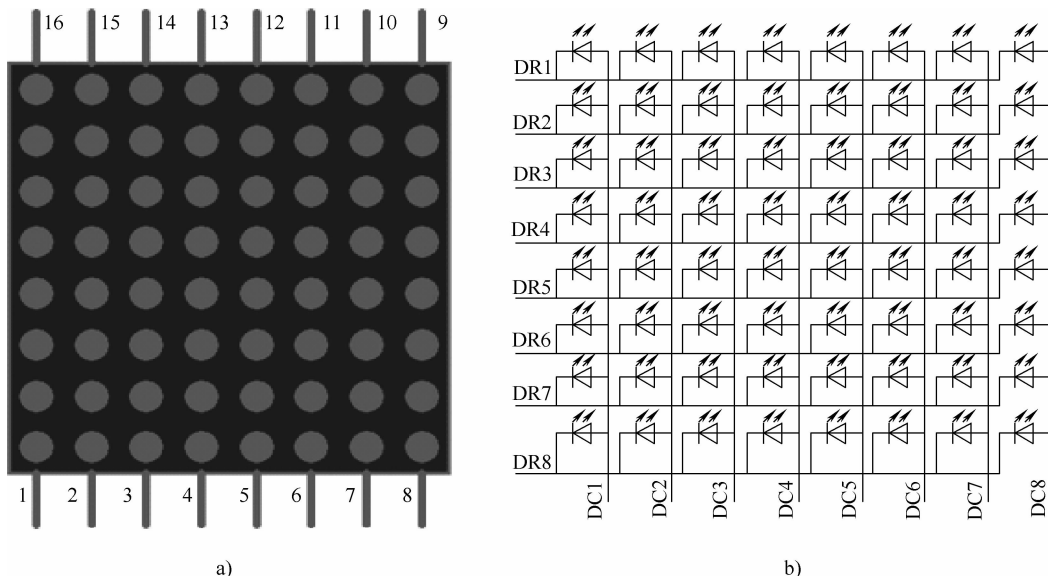


图 4-25 8×8 LED 的点阵

a) 8×8 LED 点阵实物图 b) 8×8 LED 点阵内部结构图

初步训练

利用 8×8 LED 点阵显示数字 0~9。具体要求：点阵显示屏依次显示 0~9 十个数字。

任务分析：

数字 0~9 点阵显示代码如图 4-26a 所示，假设显示数字“0”，形成的列代码为 00H，00H，3EH，41H，41H，41H，3EH，00H。只要把这些代码分别送到相应的列线上面，即可实现“0”的数字显示。

送显示代码过程：送第一列线代码到 P3 口，同时置第一行线为“0”，其他行线为“1”，延时 2ms 左右；送第二列线代码到 P3 口，同时置第二行线为“0”，其他行线为“1”，延时 2ms 左右；如此下去，直到送完最后一列代码，再从头开始送。

数字 1~9 的代码如下（图 4-26b~j）：

数字“1”代码为 00H，00H，00H，00H，21H，7FH，01H，00H。

数字“2”代码为 00H，00H，27H，45H，45H，45H，39H，00H。

数字“3”代码为 00H，00H，22H，49H，49H，49H，36H，00H。

数字“4”代码为 00H，00H，0CH，14H，24H，7FH，04H，00H。

数字“5”代码为 00H，00H，72H，51H，51H，51H，4EH，00H。

数字“6”代码为 00H，00H，3EH，49H，49H，49H，26H，00H。

数字“7”代码为 00H，00H，40H，40H，40H，4FH，70H，00H。

数字“8”代码为 00H，00H，36H，49H，49H，49H，36H，00H。

数字“9”代码为 00H，00H，32H，49H，49H，49H，3EH，00H。

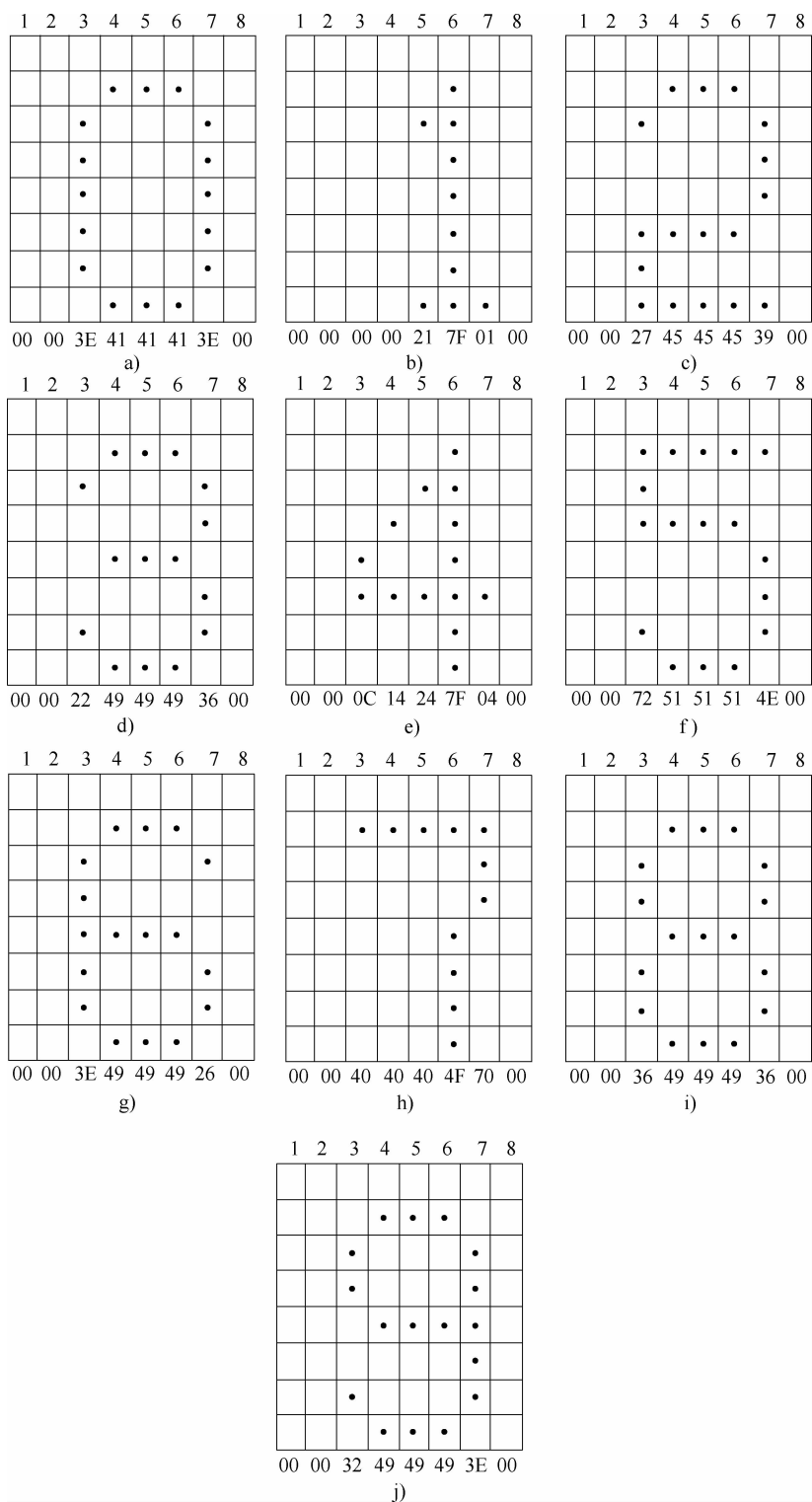


图 4-26 数字 0~9 代码

a) 数字0 b) 数字1 c) 数字2 d) 数字3 e) 数字4 f) 数字5 g) 数字6 h) 数字7 i) 数字8 j) 数字9

操作步骤:

- 1) 利用导线按表 4-9 中的对应关系将点阵 LED 和单片机连接起来。

表 4-9 点阵 LED 和单片机连线表

点阵 LED	单片机
H0 ~ H7	P0 口
L0 ~ L7	P2 口
	\overline{EA} 接 +5V

2) 运行 Keil μ Vision4 软件, 新建一个工程文件夹 (如 LED), 新建一个工程文件 LED. ASM。

- 3) 输入并编辑 LED. ASM 文件, 并且编译生成 HEX 文件。
- 4) 将生成的 HEX 文件下载到单片机中。
- 5) 运行实验程序, 观察实验现象, 点阵 LED 应显示数字 0 ~ 9。
- 6) 完成实训操作, 关闭电源, 拆除接线, 整理归位。
- 7) 完成任务工单中相关题目。

参考程序:

```

        TCOUNT EQU 30H
        R_CNT EQU 31H
        NUMB EQU 32H
        ORG 0000H
        LJMP START
        ORG 000BH
        LJMP INT_T0
START: MOV TCOUNT, #00H
        MOV R_CNT, #00H
        MOV NUMB, #00H
        MOV TMOD, #01H
        MOV TH0, # (65536 - 4000) / 256 ; 定时 4ms
        MOV TL0, # (65536 - 4000) MOD 256
        SETB TR0
        MOV IE, #82H
        SJMP $
INT_T0: MOV TH0, # (65536 - 1000) / 256
        MOV TL0, # (65536 - 1000) MOD 256
        MOV DPTR, #TAB ; 选中某列
        MOV A, R_CNT ; rcnt 为列计数器
        MOVC A, @ A + DPTR
        MOV P2, A
        MOV DPTR, #NUB ; 取此列的代码

```

```

MOV    A, NUMB                ; NUMB 为要显示的数据
MOV    B, #8
MUL    AB
ADD    A, R_CNT              ; 确定在第几行第几列
MOV    CA, @ A + DPTR
CPLA
MOV    P0, A                  ; 输出行码
INC    R_CNT
MOV    A, R_CNT
CJNE   A, #8, NEXT
MOV    R_CNT, #00H
NEXT:  INC    TCOUNT          ; 显示延时, 每个数字显示 1s
MOV    A, TCOUNT
CJNE   A, #250, NEX
MOV    TCOUNT, #00H
INC    NUMB
MOV    A, NUMB
CJNE   A, #10, NEX           ; 0~9 循环显示
MOV    NUMB, #00H
NEX:   RETI
TAB:   DB    0FEH, 0FDH, 0FBH, 0F7H, 0EFH, 0DFH, 0BFH, 7FH
NUB:   DB    00H, 00H, 3EH, 41H, 41H, 41H, 3EH, 00H
        DB    00H, 00H, 00H, 00H, 21H, 7FH, 01H, 00H
        DB    00H, 00H, 27H, 45H, 45H, 45H, 39H, 00H
        DB    00H, 00H, 22H, 49H, 49H, 49H, 36H, 00H
        DB    00H, 00H, 0CH, 14H, 24H, 7FH, 04H, 00H
        DB    00H, 00H, 72H, 51H, 51H, 51H, 4EH, 00H
        DB    00H, 00H, 3EH, 49H, 49H, 49H, 26H, 00H
        DB    00H, 00H, 40H, 40H, 40H, 4FH, 70H, 00H
        DB    00H, 00H, 36H, 49H, 49H, 49H, 36H, 00H
        DB    00H, 00H, 32H, 49H, 49H, 49H, 3EH, 00H
END

```

强化训练

将 4 块 8×8 LED 点阵合并成一个 16×16 显示屏, 并让其显示汉字“我”(图 4-27)。

任务分析:

为了方便调试, 可以把 4 块 8×8 LED 点阵组成的 16×16 显示屏的行信号扫描输出引脚和列信号显示输出引脚分别引到显示屏的两边, 原理图如图 4-28 所示。

图 4-28 中, S_i ($i = 1, 2, 3, \dots, 16$) 代表行扫描信号输出, D_i ($i = 1, 2, 3, \dots, 16$) 代表列显示信号输出。

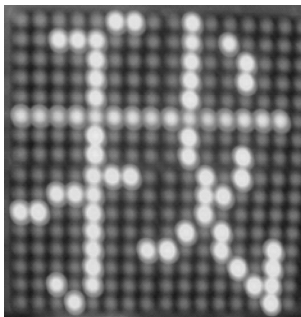
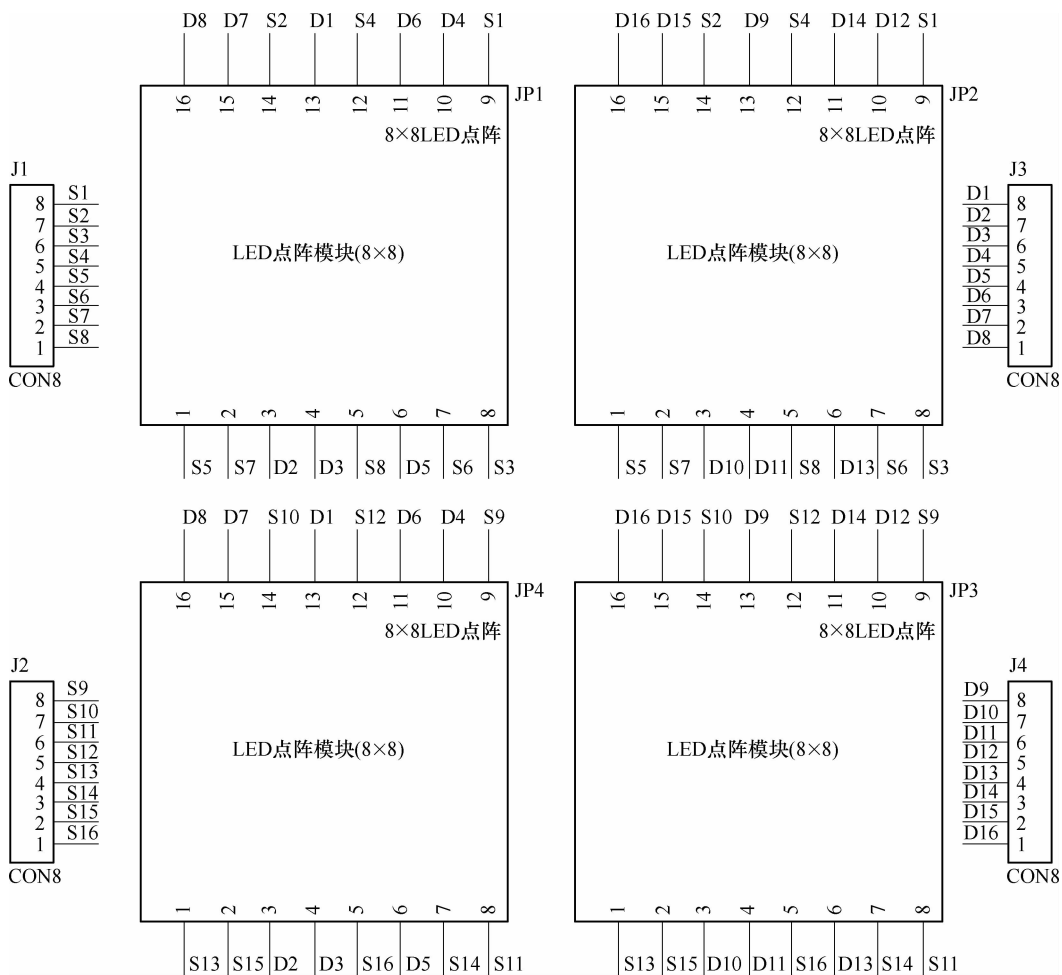


图 4-27 显示效果图


 图 4-28 4 块 8×8 LED 点阵组成 16×16 LED 点阵屏的原理图

习 题

1. LED 发光灯由哪四部分组成？其核心部分是什么？
2. 什么是人眼的视觉惰性？
3. 目前广为采用的扫描驱动方式是利用什么原理设计的？

项目四 电子时钟

任务 电子时钟设计

知识目标

1. 掌握时钟芯片 DS1302 的工作原理及编程方法。
2. 会灵活使用单片机的定时器和中断。
3. 掌握 LED 数码管动态显示任意字符的方法。

能力目标

1. 能根据项目要求，进行编程算法设计。
2. 能设计出电子时钟的硬件电路。
3. 能控制 LED 数码管按照 1s 的间隔变化数字。
4. 能用定时器定时 1s。

任务相关知识

1. DS1302 简介

DS1302 是美国 DALLAS 公司推出的涪流充电时钟芯片，内含一个实时时钟/日历电路和 31B 静态 RAM。通过简单的串行接口与单片机进行通信。实时时钟/日历电路提供秒、分、时、日、日期、月、年的信息，每月的天数和闰年的天数可以自动调整，时钟操作可通过 AM/PM 来决定采用 24 小时或 12 小时格式。DS1302 与单片机之间能简单地采用同步串行的方式进行通信，仅需用到三个口线：RES（复位）、I/O（数据线）和 SCLK（串行时钟）。时钟/RAM 的读/写数据以 1B 或者多达 31B 的字符组方式通信。DS1302 广泛应用于电话、传真、便携式仪器以及电池供电的仪器仪表等产品领域，主要的性能指标如下。

1) 实时时钟具有能计算 2100 年之前的秒、分、时、日、日期、星期、月、年的能力，还有闰年调整的能力。

- 2) 31 × 8 位暂存数据存储 RAM。
- 3) 串行 I/O 口方式使得引脚数量最少。
- 4) 宽范围工作电压：2.0 ~ 5.5V。
- 5) 工作电流：2.0V 时小于 300nA。
- 6) 读/写时钟或 RAM 数据时，有单字节传送和多字节传送两种传送方式。
- 7) 8 引脚 DIP 封装或可选的 8 引脚 SOIC 封装（根据表面装配）。
- 8) 简单三线接口。
- 9) 与 TTL 兼容（ $V_{CC} = 5V$ ）。

10) 可选工业级温度范围: $-40 \sim 85^{\circ}\text{C}$ 。

11) 在 DS1202 基础上增加的特性: 对 V_{CC1} 有可选的涓流充电能力; 双电源用于主电源和备份电源供应; 备份电源引脚可由电池或大容量电容输入; 附加 7B 暂存存储器。

2. 引脚功能及结构

DS1302 的引脚图如图 4-29 所示, 其中 V_{CC1} 为备份电源, V_{CC2} 为主电源。在主电源关闭的情况下, 备份电源能保持时钟的连续运行。DS1302 由 V_{CC1} 或 V_{CC2} 两者中的电压较大者供电。当 V_{CC2} 大于 $V_{\text{CC1}} + 0.2\text{V}$ 时, V_{CC2} 给 DS1302 供电; 当 V_{CC2} 小于 V_{CC1} 时, DS1302 由 V_{CC1} 供电。X1 和 X2 是振荡源, 外接 32.768kHz 晶振。RST 是复位/片选线, 通过把 RST 输入驱动置高电平来启动所有的数据传送。

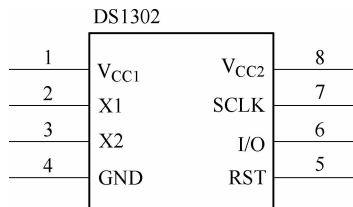
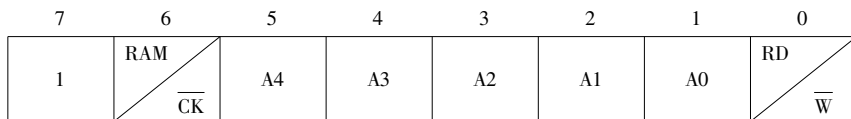


图 4-29 DS1302 引脚图

RST 输入有两种功能: 首先, RST 接通控制逻辑, 允许地址/命令序列送入移位寄存器; 其次, RST 提供终止单字节或多字节数据的传送手段。当 RST 为高电平时, 所有的数据传送被初始化, 允许对 DS1302 进行操作。如果在传送过程中 RST 置为低电平, 则会终止此次数据传送, I/O 引脚变为高阻态。上电运行时, 在 $V_{\text{CC}} \geq 2.5\text{V}$ 之前, RST 必须保持低电平。只有在 SCLK 为低电平时, 才能将 RST 置为高电平。I/O 为串行数据输入/输出端 (双向), 后面有详细说明。SCLK 始终是输入端。

3. DS1302 的控制字节

DS1302 控制字节的最高有效位 (位 7) 必须是逻辑 1, 如果它为 0, 则不能把数据写入 DS1302 中; 位 6 如果为 0, 则表示存取日历时钟数据, 为 1 表示存取 RAM 数据; 位 5 ~ 位 1 指示操作单元的地址; 最低有效位 (位 0) 为 0 表示要进行写操作, 为 1 表示进行读操作。DS1302 控制字节总是从最低位开始输出, 如下所示:



4. 数据输入/输出 (I/O)

在控制指令字输入后的下一个 SCLK 时钟的上升沿, 数据被写入 DS1302, 数据输入从低位 (即位 0) 开始。同样, 在紧跟 8 位的控制指令字后的下一个 SCLK 脉冲的下降沿读出 DS1302 的数据, 从低位 0 位到高位 7 依次读出。

5. DS1302 的寄存器

DS1302 寄存器表见表 4-10。

表 4-10 DS1302 寄存器表

寄存器	命令字节		取值范围	寄存器内容						
	写	读		7	6	5	4	3	2	1
秒寄存器	80H	81H	00 ~ 59	CH	10S			SEC		
分寄存器	82H	83H	00 ~ 59	0	10MIN			MIN		
时寄存器	84H	85H	00 ~ 23 或 00 ~ 12	12/24	0	10A/P	HR	HR		
日寄存器	86H	87H	01 ~ 28, 29, 30, 31	0	0	10DATE		DATE		
月寄存器	88H	89H	01 ~ 12	0	0	0	10M	MONTH		
周寄存器	8AH	8BH	01 ~ 07	0	0	0	0	0	DAY	
年寄存器	8CH	8DH	00 ~ 99	10YEAR			YEAR			

初步训练

利用 DS1302 实现电子时钟设计。

任务分析：

电子时钟电路图如图 4-30 所示。

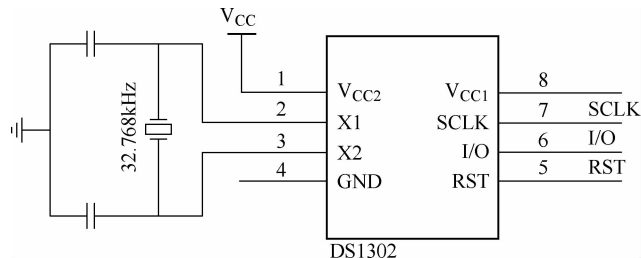


图 4-30 电子时钟电路图

根据 DS1302 的工作原理，将其与单片机对应引脚相连，见表 4-11。

表 4-11 DS1302 与单片机连线对应表

DS1302	单片机	DS1302	单片机
RST	P1.0	动态显示数码管	单片机
SCLK (CLK)	P1.1	8P 插头	P0 口
I/O	P1.2	8P 插头	P2 口
	\overline{EA} 引脚接 +5V		

操作步骤：

- 1) 将单片机 V_{CC} 接 +5V，GND 接地， \overline{EA} 接 +5V，按照 DS1302 与单片机连线对应表连接硬件电路。
- 2) 运行 Keil μ Vision4 软件，新建一个工程文件夹（如 ds1302），新建一个工程文件 ds1302.ASM。
- 3) 输入并编辑 ds1302.ASM 文件，并且编译生成 HEX 文件。
- 4) 将生成的 HEX 文件下载到单片机中。
- 5) 运行实验程序，观察实验现象，并记录在任务工单中。
- 6) 完成实训操作，关闭电源，拆除接线，整理归位。
- 7) 完成任务工单中相关题目。

习 题

1. DS1302 与单片机之间能简单地采用同步串行的方式进行通信，仅需用到几个口线？分别是什么？
2. DS1302 所需晶振频率为多少？
3. DS1302 各引脚的功能是什么？

项目五 键盘接口技术

任务一 独立按键控制设计

知识目标

1. 掌握使用单片机读取按键状态的方法。
2. 掌握消除按键抖动的方法。
3. 掌握检测按键状态的方法。

能力目标

1. 能根据项目要求, 进行编程算法设计。
2. 能设计出按键与单片机连接的硬件电路。
3. 能完成按键检测的程序设计。
4. 能实现按键去抖的硬件电路设计和软件编程设计。

任务相关知识

单片机按键及其检测方法如下。

键盘由一组按键组成, 一个按键实际上是一个开关元件。在单片机系统中实现向单片机输入数据、传送命令等功能, 是人工干预单片机的主要手段。键盘分为非编码键盘和编码键盘, 由软件完成对按键闭合状态识别的称为非编码键盘, 由专用硬件实现对按键闭合状态识别的称为编码键盘。

(1) 独立式键盘结构

独立式键盘相互独立, 每个按键占用一根 I/O 口线, 每根 I/O 口线上的按键工作状态不会影响其他按键的工作状态。这种按键软件程序简单, 但占用 I/O 口线较多 (一根口线只能接一个键), 适用于按键应用数量较少的系统中。

1) 独立式键盘的特点如下:

- ①各按键相互独立, 电路配置灵活。
- ②按键数量较多时, I/O 端线耗费较多, 电路结构繁杂。
- ③软件结构简单。

2) 键盘输入信息的主要过程如下:

- ①单片机判断是否有键按下, 如有键按下, 再去处理键值程序。
- ②确定按下的是哪一个键。
- ③把此步骤代表的信息翻译成单片机所能识别的代码, 如 ASCII 码或其他特征码。

该类键盘内的独立按键连接电路如图 4-31 所示。

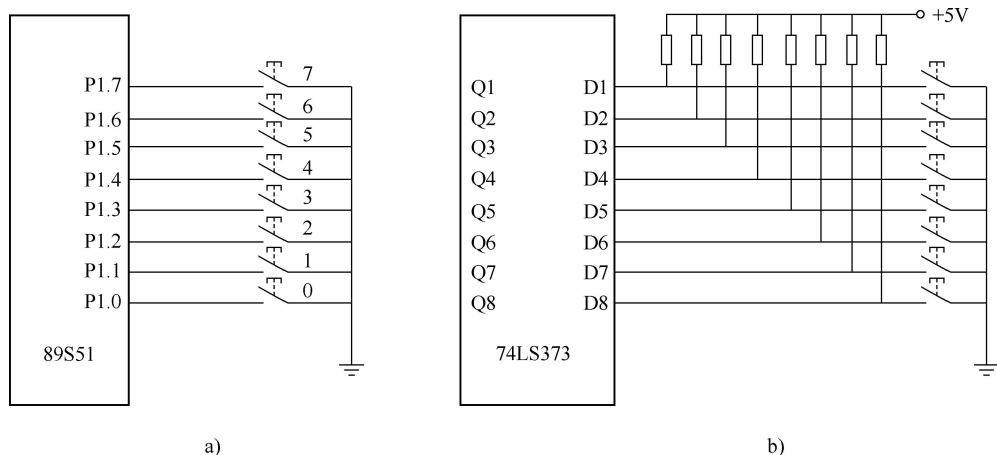


图 4-31 独立按键连接电路图
a) 芯片内有上拉电阻 b) 芯片内无上拉电阻

(2) 按键抖动及消抖

按键一般为机械开关结构，因机械触点的弹性及电压突跳等原因，往往在触点闭合或断开的瞬间会出现电压抖动，由于电压抖动造成一次按键多次处理的现象就称为按键抖动，如图 4-32 所示。

消除按键抖动的最常用的方法为软件延时消抖法。具体方法如下：

在第一次检测到有键按下时，执行一段延时 10ms 的子程序，之后，再次检测该键的电平状态，如果该键电平仍保持闭合状态，则确认为真正有键按下；否则，认为无键按下。同理，在检测到该键释放后，也应采用相同的步骤进行确认，从而可消除抖动的影响。

初步训练

查询式键盘实验。具体要求：单片机与 8 个独立式按键连接，用一个数码管显示键值，当按键按下时，数码管显示相应的数字。

任务分析：

P0 口接共阳极数码管，P3 口接查询式键盘。查询到 P3 口第 1 个按键按下时，数码管显示“1”，第 2 个按键按下时，数码管显示“2”，依次到第 8 个按键按下时，数码管显示“8”。

根据以上分析，画出电路图，如图 4-33 所示。

程序流程图如图 4-34 所示。

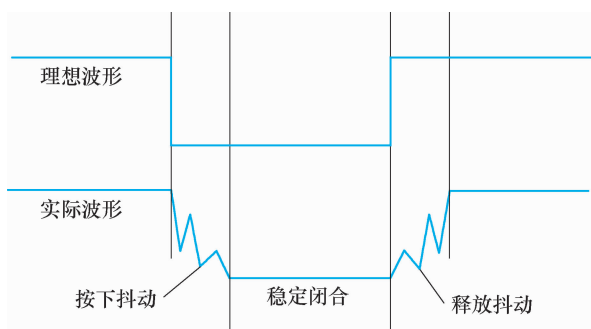


图 4-32 键闭合和释放时的电压抖动

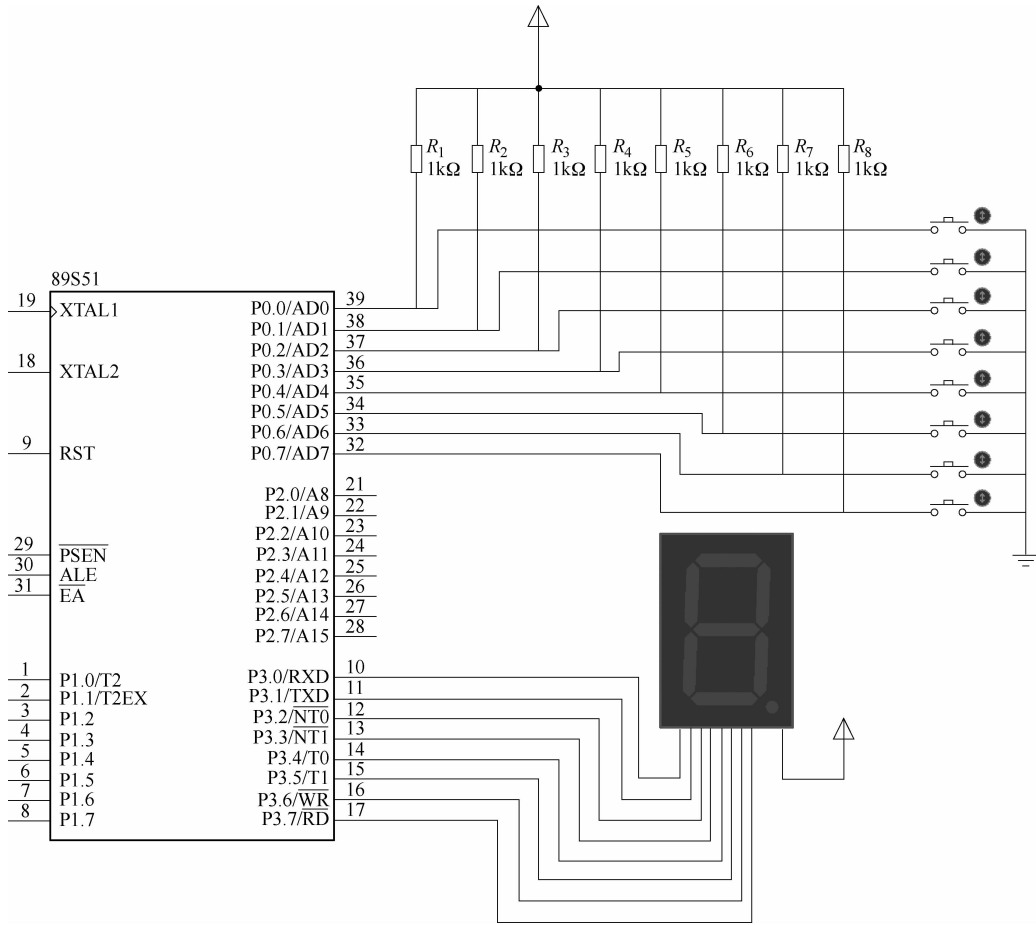


图 4-33 单片机与查询式键盘、数码管显示器连接电路图

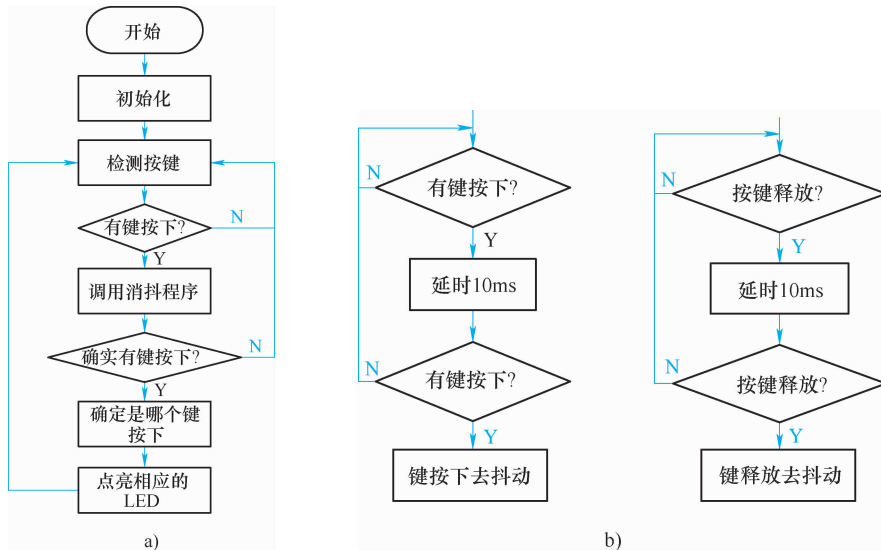


图 4-34 程序流程图

a) 主程序 b) 软件去抖动服务子程序

操作步骤：

- 1) 将单片机 V_{CC} 接 +5V, GND 接地, \overline{EA} 接 +5V。
- 2) 将 P0 口接共阳极数码管电路, P3 口接键盘 K1 ~ K8, 打开电源。
- 3) 打开 Keil μ Vision4 软件, 根据流程图进行编程, 并调试通过, 生成 HEX 文件。
- 4) 将生成的 HEX 文件下载到单片机中。
- 5) 依次按下 K1 ~ K8, 观察数码管显示情况并记录在任务工单中。
- 6) 完成实训操作, 关闭电源, 拆除接线, 整理归位。
- 7) 完成任务工单中相关题目。

参考程序：

```

                ORG    0000H                                AJMP    L0
                AJMP  START                                L2:    MOV    P0, #0A4H
                ORG    0030H                                AJMP    L0
START:  MOV    P3, #0FFH                                L3:    MOV    P0, #B0H
                MOV    P2, #0FFH                            AJMP    L0
                CLR    P2.0                                L4:    MOV    P0, #99H
                MOV    P0, #03H                            AJMP    L0
L0:     JNB    P3.0, L1                                L5:    MOV    P0, #92H
                JNB    P3.1, L2                            AJMP    L0
                JNB    P3.2, L3                                L6:    MOV    P0, #82H
                JNB    P3.3, L4                            AJMP    L0
                JNB    P3.4, L5                                L7:    MOV    P0, #0F8H
                JNB    P3.5, L6                            AJMP    L0
                JNB    P3.6, L7                                L8:    MOV    P0, #80H
                JNB    P3.7, L8                            AJMP    L0
                AJMP  L0                                    END
L1:     MOV    P0, #0F9H

```

强化训练

用独立按键控制智能车双闪灯。具体要求：该电路由两部分组成：一部分是单片机最小系统电路；另一部分是 P1.0 引脚和 P3.0 引脚分别接一个 LED, P2.0 引脚接一个按键。任务要求：模拟智能车双闪灯, 当按键按下奇数次时, LED 闪烁; 当按键按下偶数次时, LED 熄灭。试设计硬件电路并编写程序, 实现以上功能。

任务分析：

如图 4-35 所示, 在系统断电情况下, 连接电路: 用两条连接线将单片机最小系统模块的 P1.0 引脚和 P3.0 引脚与 LED 模块连接起来, P2.0 引脚与一个独立按键相连接, 将主机模块的 +5V 电源、显示模块的 +5V 电源、按键模块的 +5V 电源和电源模块的 +5V 电源连接, 同时将主机模块、显示模块、按键模块的 GND 和电源模块的 GND 连接在一起。

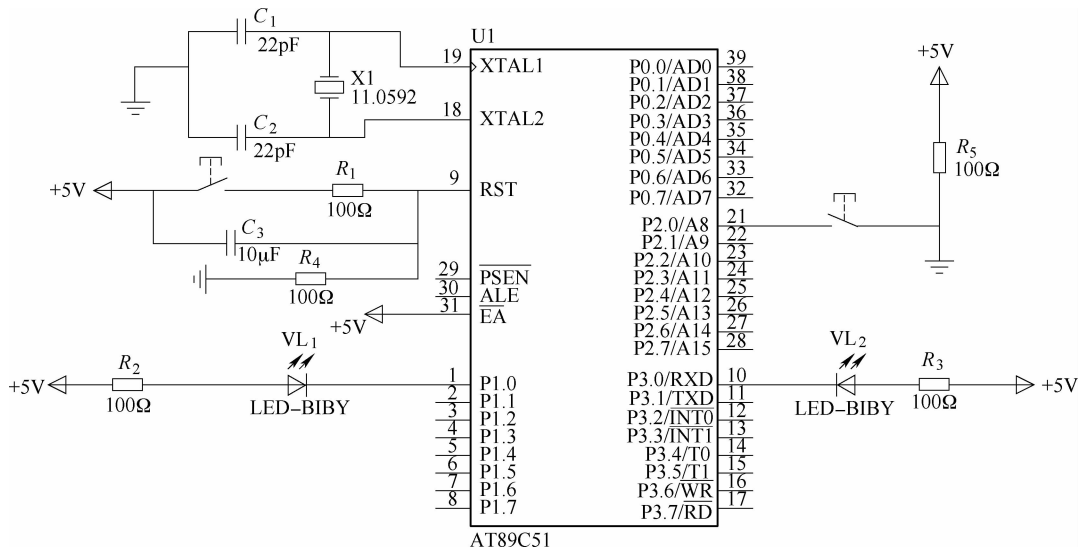


图 4-35 单片机与 LED 电路连接原理图

习 题

1. 键盘可分为_____和_____。
2. 独立式键盘的特点有哪些？
3. 键盘检测输入信息的主要过程有哪些？
4. 按键抖动的消除方法有哪几种？
5. 阐述独立式键盘的应用场合及优缺点。
6. 阐述按键去抖动的含义。

◆ 任务二 矩阵键盘控制设计

📖 知识目标

1. 掌握使用单片机读取矩阵按键状态的方法。
2. 掌握消除按键抖动的方法。
3. 掌握检测按键状态的方法。

🔧 能力目标

1. 能根据项目要求，进行编程算法设计。
2. 能设计出矩阵按键与单片机连接的硬件电路。
3. 能完成矩阵按键检测的程序设计。



任务相关知识

当键盘中按键数量较多时,为了减少对 I/O 口线的占用,通常将按键排列成矩阵形式(这是一种常见的连接方式),这种键盘称为矩阵键盘,也称为行列键盘。矩阵键盘采用行列式结构,由行线和列线组成,按键位于行、列的交叉点上。当键被按下时,其交点的行线和列线接通,相应的行线或列线上的电平发生变化,单片机通过检测行或列线上的电平变化即可确定哪个按键被按下。其特点是键多时占用 I/O 口线少,硬件资源利用合理,但判键速度慢,多用于设置数字键,适用于键数多的场合。

4×4 矩阵键盘的原理电路如图 4-36 所示,矩阵键盘的行线通过电阻接 +5V 电源,当键盘上没有键闭合时,所有的行线与列线是断开的,行线 D7~D4 均为高电平。当某键闭合时,则该键所对应的行线与列线短路。利用这一点,采用扫描方法检测键盘有无键按下,然后再判断键号。先将列线全送 0,若行线不全为 1,则有键按下,否则无键按下;若有键按下,再将列线逐列置低电平,检查行线状态来判断键盘中哪个键按下。逐行逐列地检查键盘状态的过程称为对键盘的一次扫描。

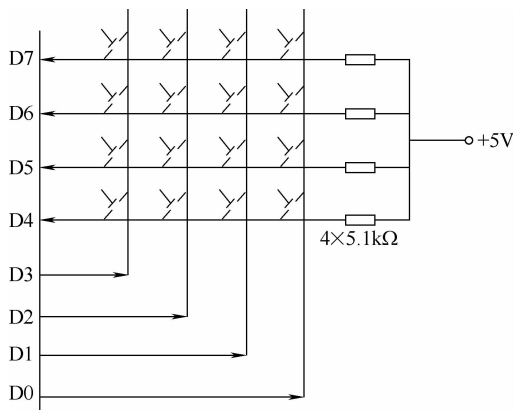


图 4-36 4×4 矩阵键盘的原理电路

在实际系统中,键盘扫描只是 CPU 的工作内容之一。CPU 在忙于各项工作任务时,既要兼顾键盘扫描,又不能过多占用 CPU 时间。为此,要根据实际情况,选择键盘的工作方式。

(1) 编程扫描方式

编程扫描方式是 CPU 在工作空余调用键盘扫描子程序,响应键输入要求。当 CPU 执行键功能程序时,不再响应键输入要求。

(2) 定时扫描方式

定时扫描方式是利用定时器产生定时(10ms)中断,CPU 响应中断后对键盘进行扫描,并在有键闭合时转入该键的功能处理程序。定时扫描的键盘电路与编程方式相同。定时扫描的优点是能及时响应键输入;缺点是无论有无键闭合,CPU 都要定时扫描,浪费 CPU 时间。

(3) 中断扫描方式

在编程扫描和定时扫描方式中,CPU 可能空扫描或不能及时响应键输入。为了克服这一缺点,可以采用中断扫描方式:如果键盘中无键闭合,CPU 执行当前程序;当有键闭合时,发出中断请求,CPU 在中断服务程序中完成键扫描并执行键功能程序。中断扫描既能及时处理键输入,又能提高 CPU 运行效率。

矩阵键盘的识别有行扫描法和线反转法两种方法。

(1) 行扫描法

所谓行扫描法,就是当行线发出低电平信号时,如果该行线所连接的键没有被按下,则列线所连接的输出端口得到的是全“1”信号;如果有键被按下,则列线所连接的输出端口得到的是非全“1”信号,如图 4-37 所示。

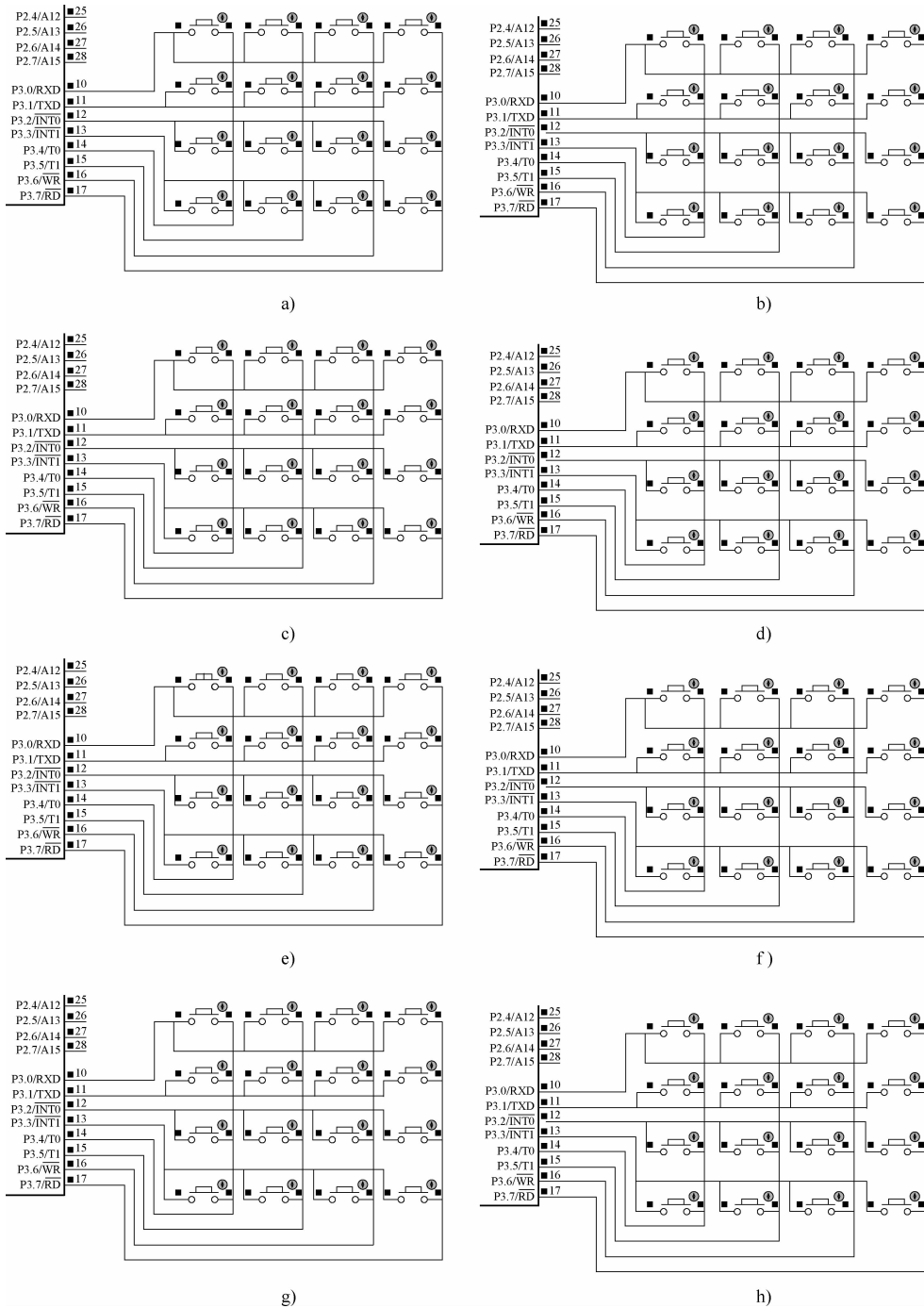


图 4-37 行扫描法按键检测图

- a) 扫描第 1 行，第 1 行无键按下
- b) 扫描第 1 行，第 1 行有键按下
- c) 扫描第 2 行，第 2 行无键按下
- d) 扫描第 2 行，第 2 行有键按下
- e) 扫描第 3 行，第 3 行无键按下
- f) 扫描第 3 行，第 3 行有键按下
- g) 扫描第 4 行，第 4 行无键按下
- h) 扫描第 4 行，第 4 行有键按下

具体过程如下：

首先，为了提高效率，一般先快速检查整个键盘中是否有键被按下，然后再确定按下的是哪一个键。

其次，用逐行扫描的方法来确定闭合键的具体位置。方法是：先扫描第 0 行，即输出 1110（第 0 行为“0”，其余 3 行为“1”），然后读入列信号，判断是否为全“1”。

(2) 线反转法

具体过程如下：先将行线作为输出线，列线作为输入线，行线输出全“0”信号，读入列线的值，然后将行线和列线的输入输出关系互换，即列线全部输出“0”信号，行线作为输入线，那么在闭合键所在的行线上值必为 0，如图 4-38 所示。这样，当一个键被按下时，必定可读到一对唯一的行列值。线反转法也是识别闭合键的一种常用方法，该方法比行扫描法速度快，但在硬件电路上要求行线与列线均需有上拉电阻，故比行扫描法稍复杂些。

注意：如果矩阵键盘接在没有接上拉电阻的 P0 口上，此方法失效。

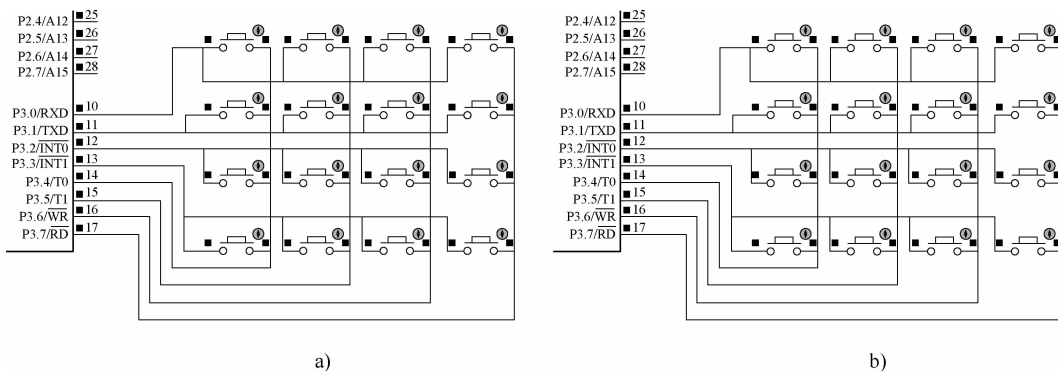


图 4-38 线反转法检测按键

a) 行键全 0，读取列键 b) 列键全 0，读取行键

以此便可判断出是第几行第几列的键被按下，可以计算键值。

键盘编码为：第 1 行 0、1、2、3，第 2 行 4、5、6、7，第 3 行 8、9、10、11，第 4 行 12、13、14、15。

现在第 2 行第 2 列的按键被按下，LED 显示结果为 5。测试结果如图 4-39 所示。

初步训练

了解 4×4 矩阵键盘的识别原理和编程方法。

任务分析：

1) 每个按键都有它的行值和列值，行值和列值的组合就是识别这个按键的编码。键盘处理程序的任务是：确定有无键按下，判断哪一个键按下，键的功能是什么；消除按键在闭合或断开时的抖动。两个并行口中，一个输出扫描码，使按键逐行动态置高电平，另一个并行口输入按键状态，由行扫描值和回馈信号共同形成键编码而识别按键，通过软件查表查出该键的功能。

2) 按键按下时，在数码管显示相应的值 0~F。

根据以上分析，画出电路图，如图 4-40 所示。程序流程图如图 4-41 所示。

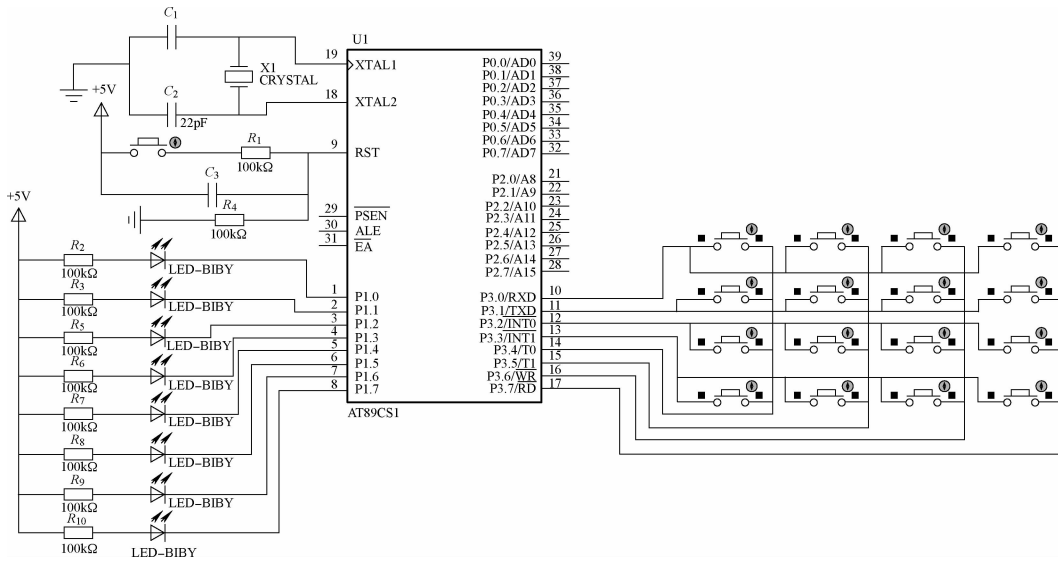


图 4-39 第 2 行第 2 列的按键按下时 proteus 仿真图

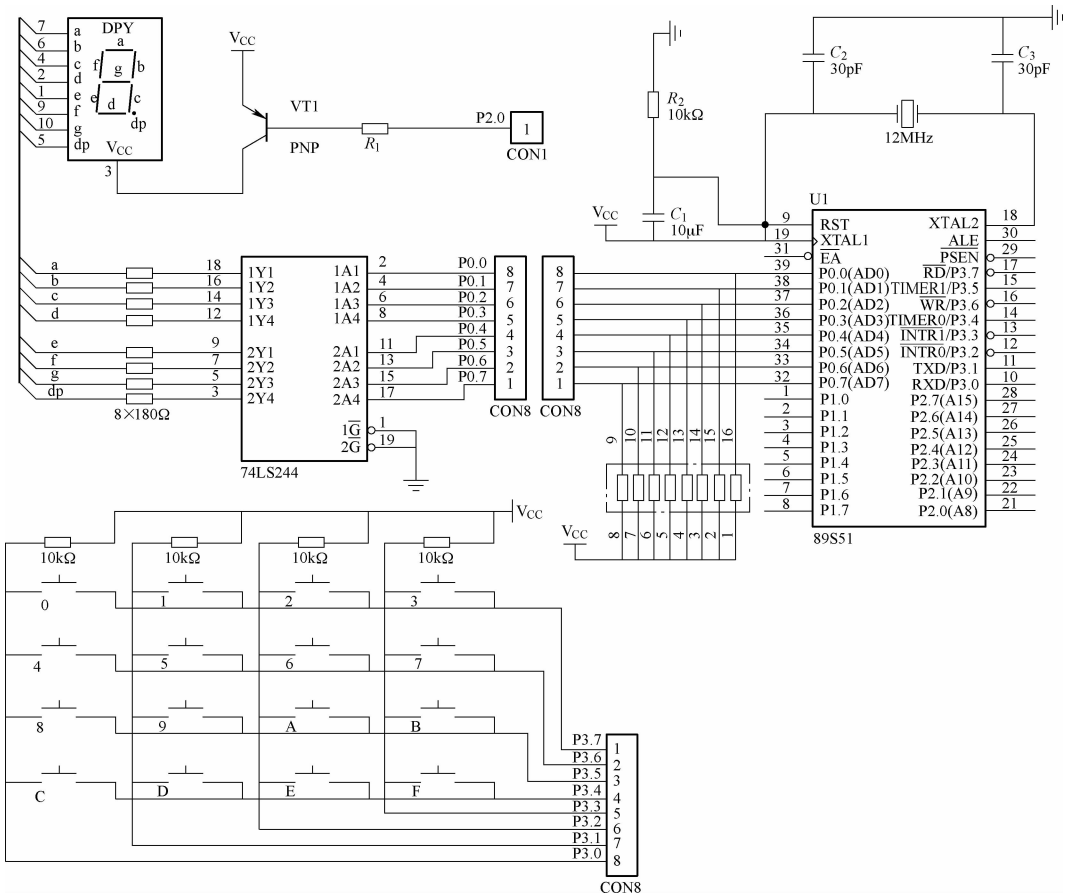


图 4-40 矩阵键盘识别电路原理图

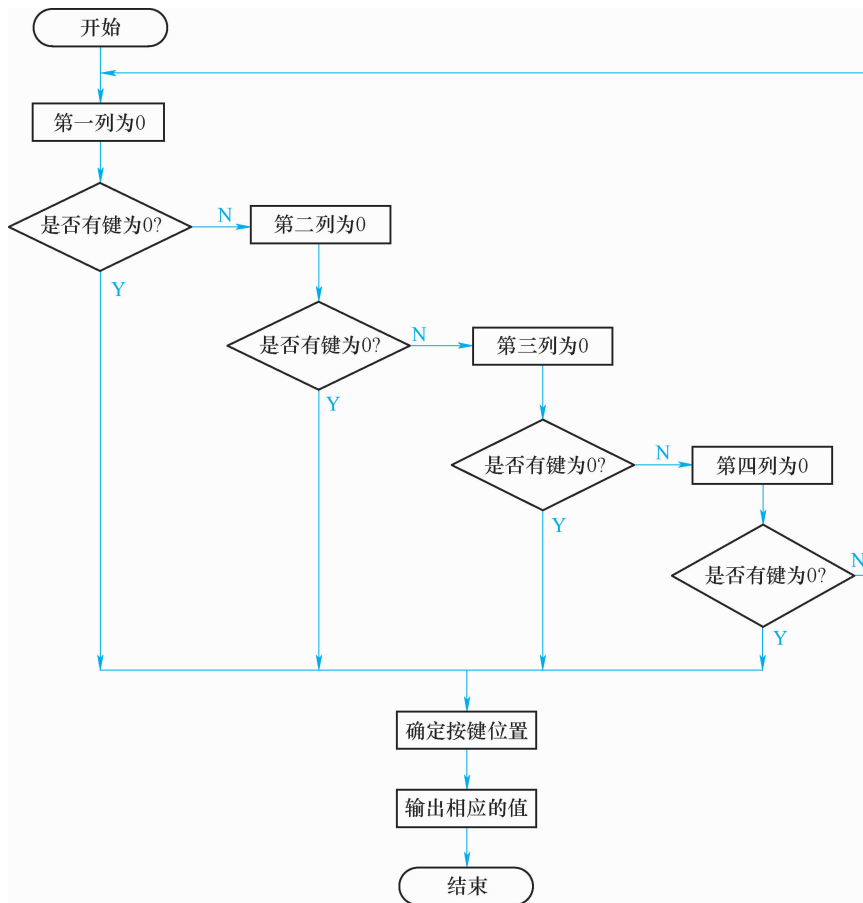


图 4-41 程序流程图

操作步骤：

- 1) 将单片机 V_{CC} 接 +5V，GND 接地， \overline{EA} 接 +5V。
- 2) 将 P3 口接矩阵键盘，P0 口接数码管段选线，P2.0 口接数码管位选线，打开电源。
- 3) 打开 Keil μ Vision4 软件，根据流程图进行编程，并调试通过，生成 HEX 文件。
- 4) 将生成的 HEX 文件下载到单片机中。
- 5) 依次按下按键，观察数码管显示情况，并记录在任务工单中。
- 6) 完成实训操作，关闭电源，拆除接线，整理归位。
- 7) 完成任务工单中相关题目。

参考程序：

可扫二维码获取。

强化训练

4×4 矩阵键盘计算器设计。具体要求：分别识别 16 个键值；设定前 10 个键值为 0~9，后面键值分别是 A、B、C、D、—，最后一个键值为确认键；利用 LED 数码管实现 255 内的减法运算；对 4×4 矩阵键盘先进行全扫描，再进行行扫描；在 LED 数码管中动态显示扫描的内容。

任务分析：

利用 89S51 进行数据处理，利用 4×4 矩阵键盘进行按键扫描，利用 LED 数码管进行动态显示。该设计首先是运用矩阵键盘的按键扫描，然后把扫描得到的键值进行译码，其次把译码所得的数值进行处理，最后将处理后的值进行动态显示。矩阵键盘和 LED 数码管分别接在 P1 口和 P0 口，这样可大大节约单片机的 I/O 端口。

1. 硬件原理图设计

参考思路：利用 proteus 仿真电路（电路见图 4-42），矩阵键盘的四列接在 P1 口的低 4 位，矩阵键盘的四行接在 P1 口的高 4 位，LED 数码管的段选码接口与 P0 口相连，LED 数码管的位选码接口与 P2.5~P2.7 相连。

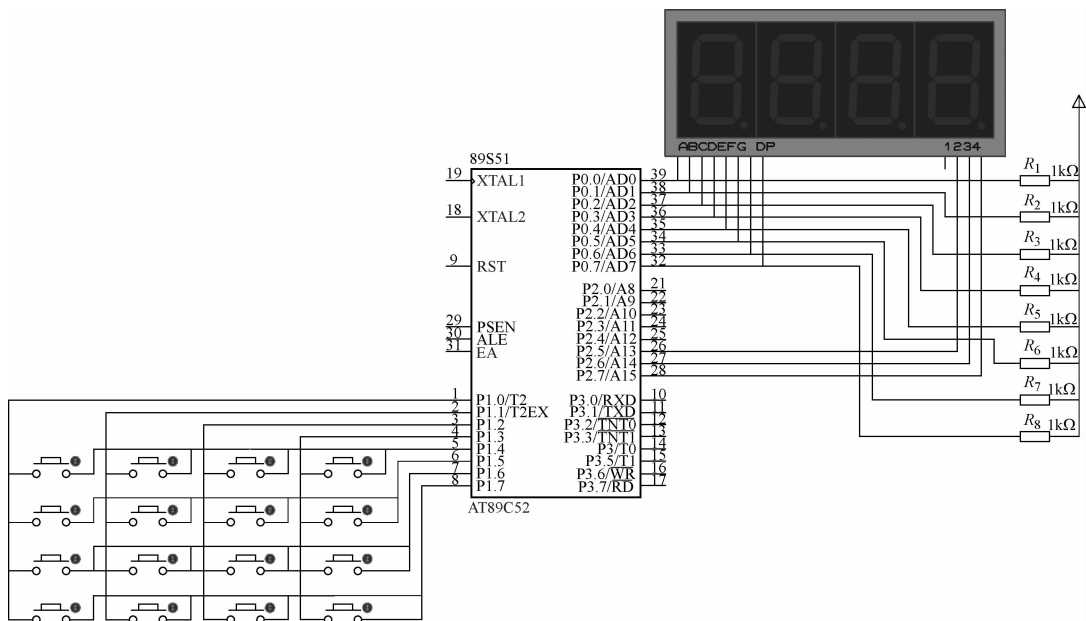


图 4-42 proteus 仿真电路图

2. 软件设计

初始化后，程序开始进入主程序，进行按键判断，如果有键按下，继续向下执行，否则重新扫描；如果有键按下，延时 5ms；再次进行按键判断，如果有键按下，继续向下执行，否则重新扫描；如果有键按下，确定按键位置；再次进行按键判断，如果有键按下，继续向下执行，否则，表示出错，重新扫描；如果有键按下，对按键译码；判断是否按键释放，如果释放，继续向下执行，否则等待；对送来的译码值进行数据处理；把处理的数据，放到

LED 数码管中动态显示。

根据以上分析，程序流程图如图 4-43 所示。

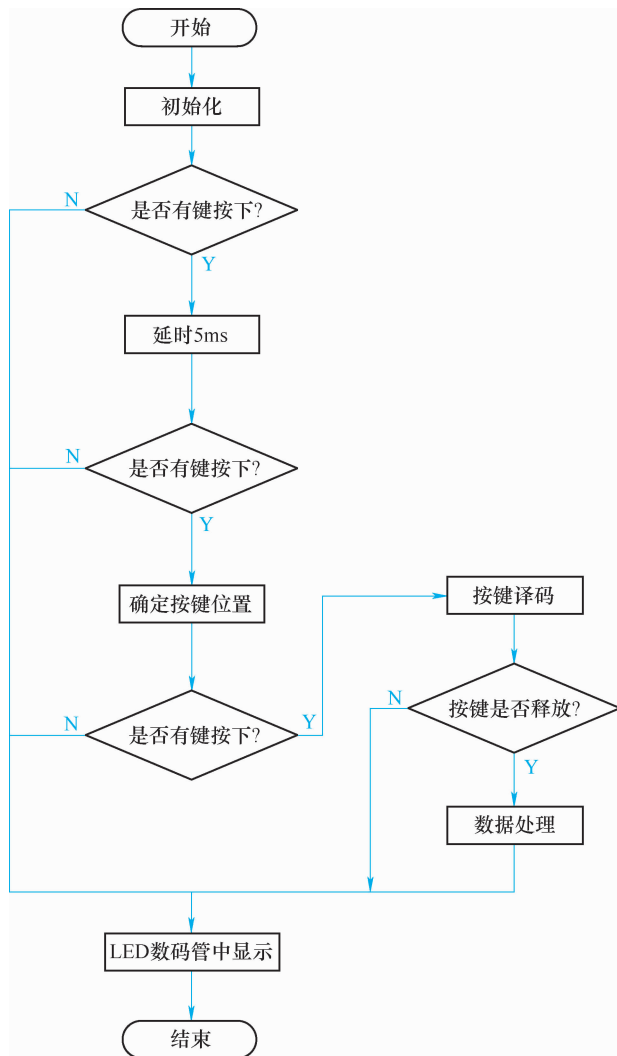


图 4-43 程序流程图

操作步骤：

- 1) 将单片机 V_{CC} 接 +5V，GND 接地， \overline{EA} 接 +5V。
- 2) 将 P1 口接矩阵式键盘，低 4 位接列线，高 4 位接行线；P0 口接数码管段选线，P2.5 ~ P2.7 引脚接数码管位选线，打开电源。
- 3) 打开 Keil μ Vision4 软件，根据流程图进行编程，并调试通过，生成 HEX 文件。
- 4) 将生成的 HEX 文件下载到单片机中。
- 5) 依次进行计算，观察数码管显示情况，并记录在任务工单中。
- 6) 完成实训操作，关闭电源，拆除接线，整理归位。
- 7) 完成任务工单中相关题目。

习 题

1. 简述矩阵键盘的按键扫描方法。
2. 简述矩阵键盘的扫描原理。
3. 独立式键盘和矩阵键盘分别具有什么特点？各适用于什么场合？

任务三 蜂鸣器设计

知识目标

1. 回顾定时器的使用方法。
2. 掌握蜂鸣器的工作原理。
3. 掌握蜂鸣器发出不同音调声音的原理。

能力目标

1. 能完成蜂鸣器、键盘与单片机的连接电路。
2. 能完成用键盘控制蜂鸣器发出不同音调的程序设计。

任务相关知识

本任务在前面学习的基础上，综合应用定时器、蜂鸣器和矩阵键盘的知识。定时器相关知识参考“流水灯设计”的相关内容；蜂鸣器相关知识参考“汽车喇叭模拟控制”和“电子音乐设计”两部分内容；矩阵键盘相关知识参考“矩阵键盘控制”的相关内容。

初步训练

利用定时器和蜂鸣器发出 1 哆、2 来、3 咪、4 发、5 嗦、6 拉、7 西音调来。

任务分析：

音调由不同频率的方波产生，音调与频率的关系见表 4-12。要产生音频方波，只要算出某一音频的周期（1/频率），然后将此周期除以 2（即为半周期的时间）即可。利用定时器对此半周期时间计时，每当计时到后就将输出方波的 I/O 口反相，然后重复计时此半周期时间，再对 I/O 口反相，就可在 I/O 口得到此频率的方波。

每个音调相应的定时器初值 X 可按下法计算：

$$\frac{1}{2} \times \frac{1}{f} = \frac{12}{f_{\text{osc}}} \times (2^{16} - X)$$

$$\text{则} \quad X = 2^{16} - \frac{f_{\text{osc}}}{24 \times f}$$

当晶振 $f_{\text{osc}} = 12\text{MHz}$ 时，各音调相应的定时器初值 X 见表 4-12。

表 4-12 音调与频率的关系

音阶	低频 f/Hz	X	中频 f/Hz	X	高频 f/Hz	X
1	262	F88CH	523	FC44H	1047	FE22H

(续)

音阶	低频 f/Hz	X	中频 f/Hz	X	高频 f/Hz	X
2	294	F95BH	587	FCACH	1175	FE56H
3	330	FA15H	659	FD09H	1319	FE85H
4	349	FA67H	698	FD34H	1397	FE9AH
5	392	FB04H	784	FD82H	1568	FEC1H
6	440	FB90H	880	FDC8H	1760	FEE4H
7	494	FC0CH	988	FE06H	1976	FF03

根据以上分析, 画出电路图, 如图 4-44 所示。

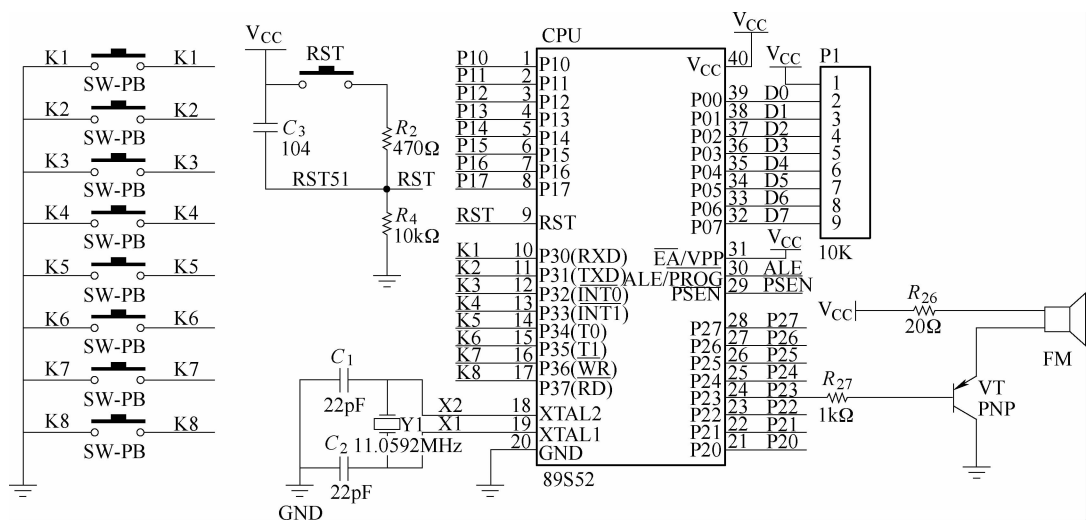


图 4-44 电子琴发出 7 个音调电路原理图

程序流程图如图 4-45 所示。

操作步骤:

- 1) 连线: 使用连接线, 在断电状态下按照原理图连接硬件电路。
- 2) 运行 Keil $\mu\text{Vision4}$ 软件, 新建一个工程文件夹 (如 BJDJ), 新建一个工程文件 BJDJ. ASM。
- 3) 输入并编辑 BJDJ. ASM 文件, 并且编译生成 HEX 文件。
- 4) 用编程器将生成的 HEX 文件下载到单片机中, 或用仿真器来执行程序, 将程序下载到仿真器中。
- 5) 运行实验程序, 按下按钮, 将会听到扬声器发出一定频率的声音。按下不同的按键, 将听到不同的声音。
- 6) 完成实训操作, 关闭电源, 拆除接线, 整理归位。
- 7) 完成任务工单中相关题目。

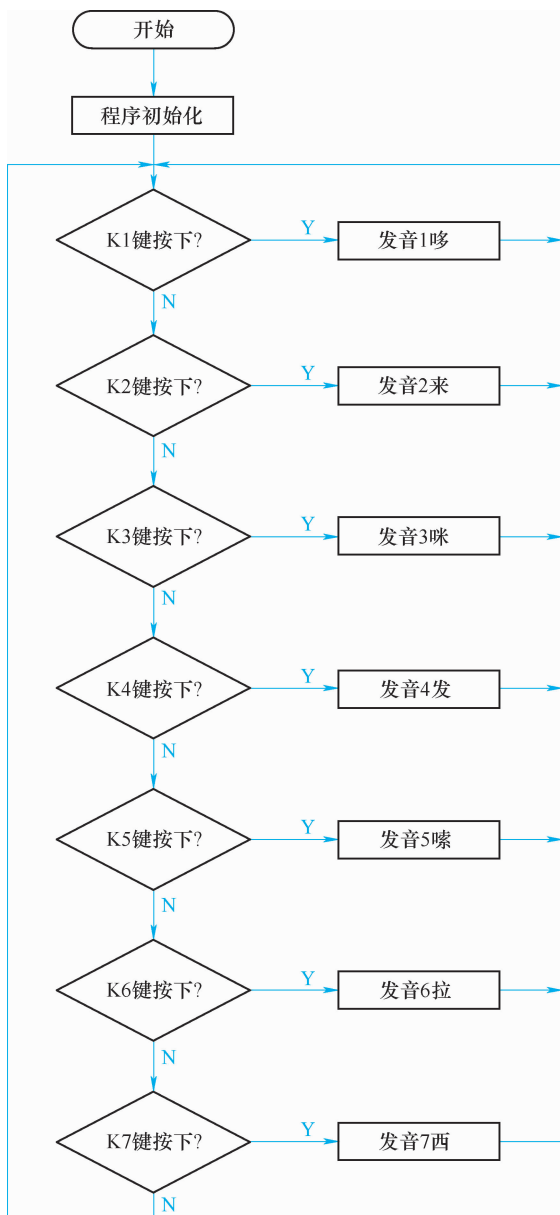


图 4-45 程序流程图

强化训练

电子琴模拟实验。具体要求：设计出蜂鸣器与单片机连接的硬件电路，利用定时器改变蜂鸣器发音频率，进一步学习矩阵键盘的编程方法。

任务分析：

音乐产生的方法：一首音乐是许多不同的音阶组成的，而每个音阶对应着不同的频率，可以利用不同频率的组合构成想要音乐。可以利用单片机的定时/计数器 T0 来产生不同的频率，只要把音阶对应频率的关系弄正确即可。表 4-13 为以单片机 12MHz 晶振为例，列出的音符、频率、简谱码对照表。

表 4-13 音符、频率、简谱码对照表

音 符	频率/Hz	简谱码 (T 值)	音 符	频率/Hz	简谱码 (T 值)
低 1D0	262	63628	#4 FA#	740	64860
#1D0#	277	63731	中 5 SO	784	64898
低 2RE	294	63835	#5 SO#	831	64934
#2RE#	311	63928	中 6 LA	880	64968
低 3M	330	64021	#6	932	64994
低 4FA	349	64103	中 7 SI	988	65030
#4FA#	370	64185	高 1 DO	1046	65058
低 5SO	392	64260	#1 DO#	1109	65085
#5SO#	415	64331	高 2 RE	1175	65110
低 6LA	440	64400	#2 RE#	1245	65157
#6	466	64463	高 3 M	1245	65157
低 7SI	494	64524	高 4 FA	1397	65178
中 1 DU	523	64580	#4 FA#	1480	65198
#1 DU#	554	64633	高 5 SO	1568	65217
中 2 RE	587	64684	#5 SO#	1661	65235
#2 RE#	622	64732	高 6 LA	1760	65252
中 3M	659	64777	#6	1865	65268
中 4FA	698	64820	高 7 SI	1967	65283

此实验键盘所对应的音符见表 4-14。

表 4-14 按键与音符对照表

3̣	4̣	5̣	6̣
7̣	1	2	3
4	5	6	7
1̣	2̣	3̣	4̣

根据以上分析，画出电路图，如图 4-46 所示。

操作步骤：

- 1) 矩阵键盘区 8P 插槽连到单片机最小系统 P3 端口的 8P 插槽，单片机的 EA 脚接 +5V。
- 2) 音频电路区的 IN 端口连到单片机最小系统 1 脚端口。
- 3) 运行 Keil μ Vision4 软件，根据流程图进行编程，并调试通过，生成 HEX 文件。
- 4) 将生成的 HEX 文件下载到单片机中。
- 5) 运行实验程序，按下按钮，将会听到扬声器发出一定频率的声音。按下不同的按键，将听到不同的声音。

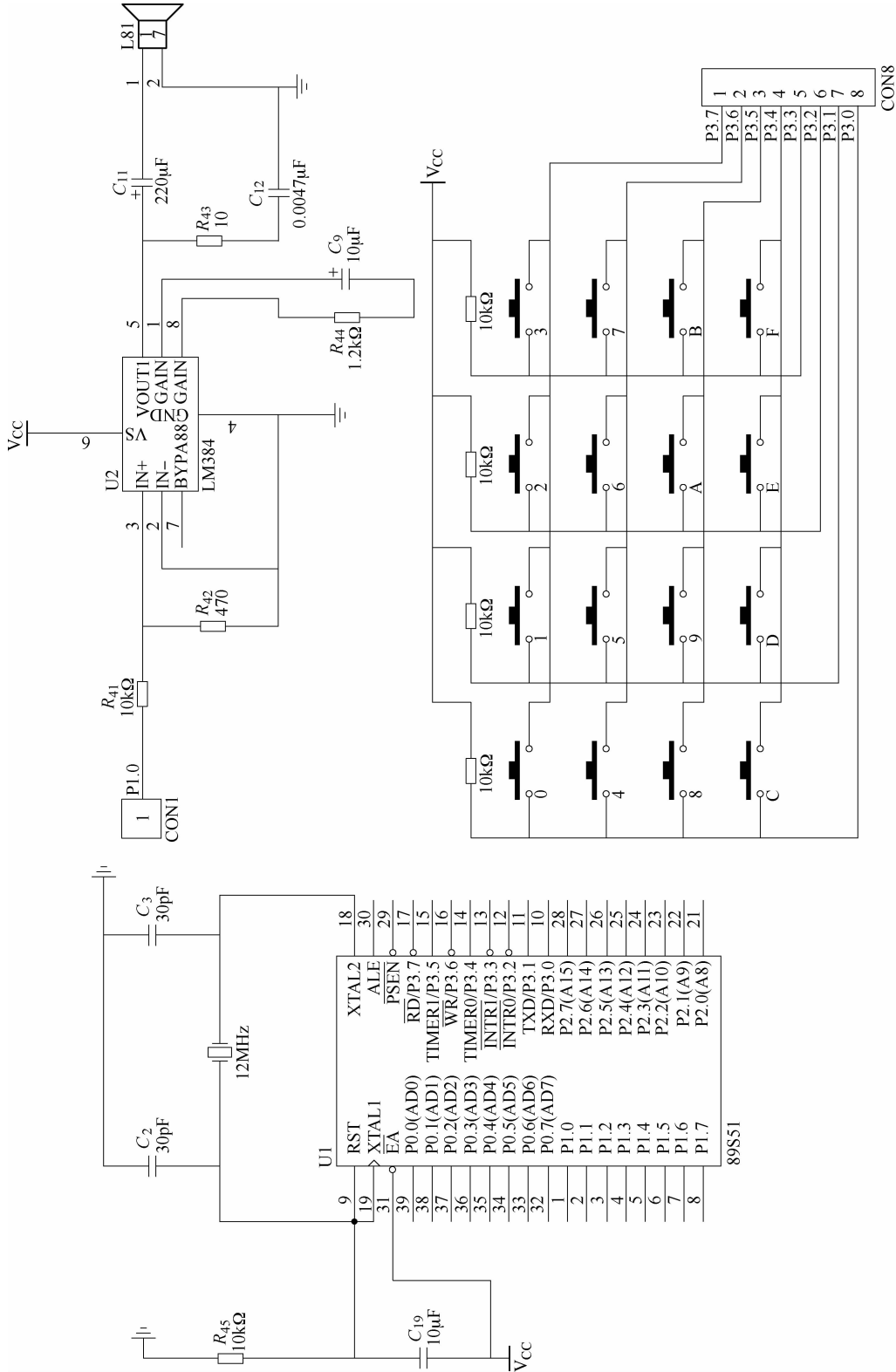


图4-46 电子琴实验电路图

- 6) 完成实训操作, 关闭电源, 拆除接线, 整理归位。
- 7) 完成任务工单中相关题目。

习 题

1. 定时/计数器的实质是_____。
2. MCS-51 单片机有_____个 16 位可编程定时/计数器, 有_____种工作方式。
3. MCS-51 单片机的最大计数值为_____, 此时工作于工作方式_____。
4. 若定时/计数器工作于工作方式 0 时, 为_____位的定时/计数器, TL 的低 5 位计满溢出时, 向高_____位的 TH 进位。
5. 处于不同工作方式时, 初值的计算方法分别是什么?