

浙江大学 实验报告

课程名称: 微机原理及应用实验
实验名称: 数字计时器

指导老师: 胡斯登

专业: 电子信息工程
姓名: 冯静怡
学号: 3220104119
地点: 紫金港东三 406
日期: 2024 年 5 月 12 日

1 实验目的

1. 学习使用单片机的各种接口
2. 学会按键扫描和数码管扫描显示
3. 学会使用定时器中断的状态机运行方式

2 实验过程

2.1 分析实验完成功能

1. 自定义倒计时时间;
2. 按“开始/暂停”键实现倒计时开始, 再次按“开始/暂停”键暂停倒计时;
3. 若当前计时为 00-00, 按“开始/暂停”开始正计时功能, 再次按“开始/暂停”, 可暂停正计时功能;
4. 按归零键, 可使计时归零;
5. 数码管显示第 4 8 位显示数字 00-00。

2.2 分配硬件功能

1. 确定按键及其功能 (#XXH 表示按键编号)
 - #00H 十秒位增 1; #01H 分钟个位增一; #02H 分钟十位增一
 - #08H 归零; #0BH 开始/暂停
2. 计时器状态, 使用 RAM 44H 位置记录当前状态 (#XXH 表示 44H 内存存储数字)
 - #0FH 暂停
 - #FFH 倒计时模式; #00H 正计时模式
 - #F0H 待机模式
 - #AAH 倒计时完成

实验中计时器一共可能处于以上 5 个状态, 使用 44H 记录方便在代码的任何位置确认当前计时器状态

3. 53H-57H 地址位置记录当前显示的数字
4. 22H 位地址位置记录当前显示是否被清零

- 0 → 清零
- 1 → 有数字

方便确认按开始键是进行正计时/倒计时

5. 计时器功能分配

- TIME0: 每隔 10ms 对键盘进行扫描；进行键盘按键的确定；但实际上为了 T0 的中断和 T1 的中断不冲突，所以设置非整数倍，为 9548us。

设置：

$$T0 = (65536 - 9548)D = (DA\ B4)H$$

- TIME1: 每隔 2ms 对数码管进行扫描显示。

所以设置：

$$T1 = (65536 - 2000)D = (F8\ 30)H$$

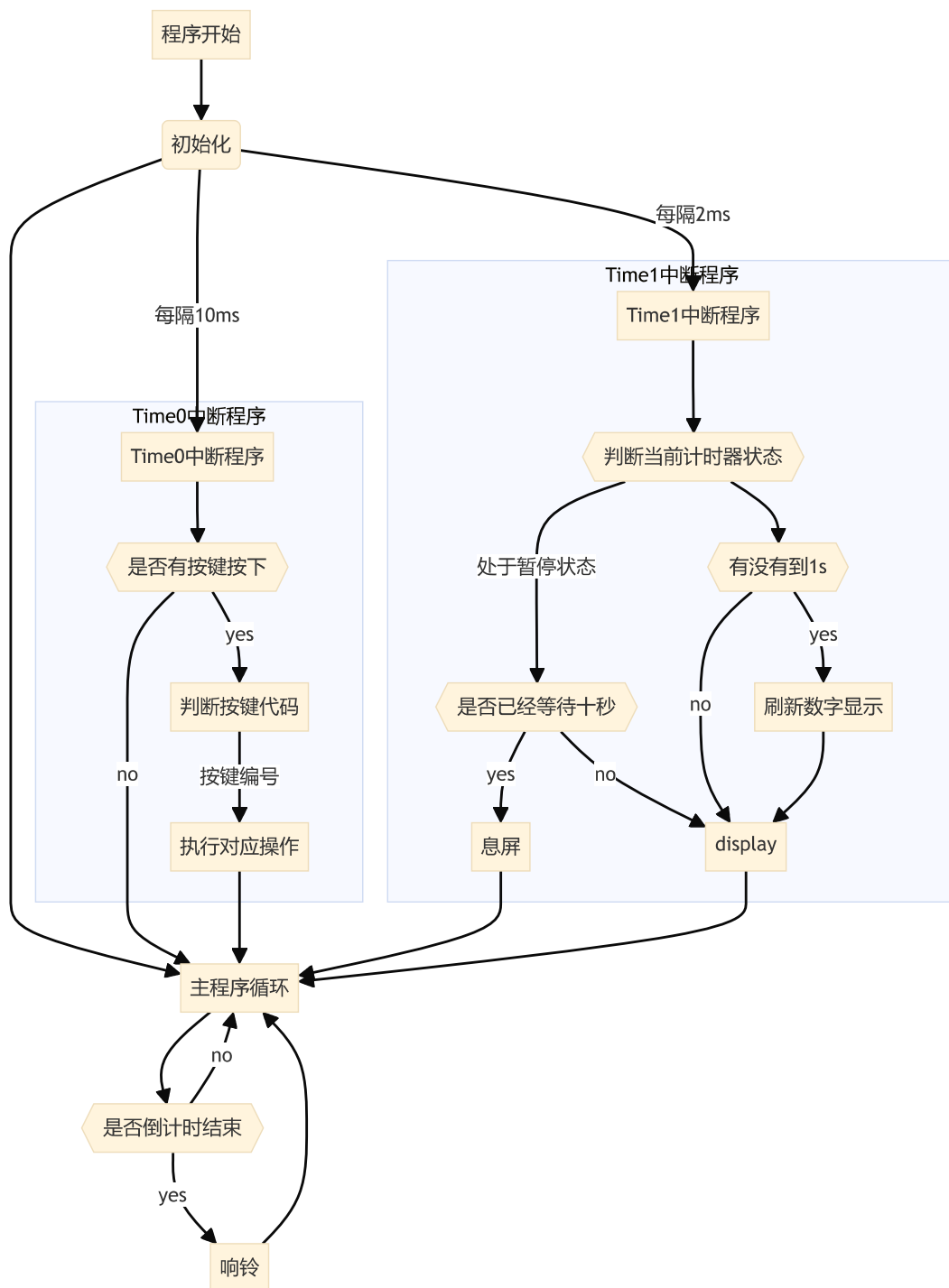
4AH&4BH 地址位置记录 1s 计数器，用于计时器的计时。

$$4BH\&4AH=10000H - 500D = (FE\ 0C)H$$

45H 地址位置记录暂停状态下的 10s 计数器, 通过 4AH&4BH 地址位置记录的 1s 计数器标志计数。

6. 30H 地址位置记录按键按下的次数。30H=2 时，开始判断按键的值（按键防抖）。当按键松开以后（30H 被清零），开始执行按下按键后的对应操作。

2.3 代码流程



3 代码编写

3.1 主程序初始化

通过下面的初始化代码，也可以分析得到各个存储单元在接下来的复杂代码中启到的作用。

1	MAIN:	15	MOV P0,#0FFH;P0口的显示值	器1	
2	; 初始化显示的数字	16	; 计时器初始化	26	MOV TH1,#0F8H
3	MOV R1,#53H	17	MOV 30H,#00H; 按键次数初始	27	MOV TL1,#050H
4	LOOP_MAIN:	化		28	; 设定定时器状态为待机状态
5	MOV @R1,#00H	18	MOV TMOD,#11H; 设定定时器	29	MOV 44H,#0F0H
6	INC R1	0、1工作方式		30	
7	CJNE R1,#58H,LOOP_MAIN	19	MOV TH0,#0DAH; 设定定时器	31	MOV 45H,#00H; 暂停->待机状
8	MOV R1,#53H	0初值			态的10s计数器
9	MOV 42H,#0F7H	20	MOV TL0,#0B4H	32	MOV 46H,#0FFH; 报警状态计
10	MOV 55H,#0DH	21	; 启动定时器0		数器
11	; 此时为零状态	22	SETB EA	33	MOV 4AH,#0CH; 1s计数器, 采
12	CLR 22H	23	SETB ET0		用32位正计数方式
13	; 八位数码管显示状态为灭灯状态	24	SETB TR0	34	MOV 4BH,#0FEH
14	MOV P2,#0FFH; 灯全灭	25	; 设定定时器1初值, 但不启动定时	35	MOV DPTR,#TABLE; 查表

3.2 主程序循环代码

主要为了在计时器结束之后能够响铃，如果计时器结束则调用响铃子程序；在执行这段代码的过程中会被定时器中断不断打断，但不影响主要功能。

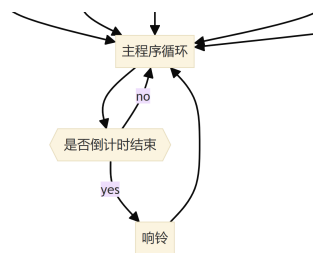


图 1: 主程序循环代码示意图

1	LOOP_SJMP:	6	NEXT_SJMP:	11	LLA:
2	MOV A,44H	7	SJMP LOOP_SJMP; 循环	12	DJNZ R7,LLA
3	CJNE A,#0AAH,NEXT_SJMP	8	DELAY: ;延时子程序	13	RET
4	CPL P3.4; 取反小喇叭(p3.3)	9	MOV R7,#00H		
5	LCALL DELAY; 调用延时				

3.3 定时器 T0 中断子程序

中断程序主要用于扫描按键，根据按键的情况进行相应的操作。

TIME0 中断主程序 调用子函数 KS，判断是否有按键按下；同时记录按键按下的次数，当按键按下次数大于 2 次，且此时按键已经松开，跳转到 OPERATE 子程序。

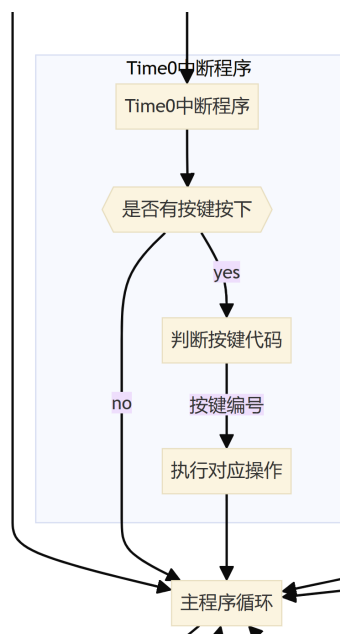


图 2: TIME0 运行逻辑图

```

1      ;定时器0中断，扫描按键          10      MOV A,30H
2      TIME0:                          11      JZ T0_END1 ;如果30H为0，
3      PUSH A                            返回中断
4      PUSH PSW                          12      CJNE A,#01H,OPERATE ;
5      MOV TH0,#0DAH                    如果30H不等于1，跳转
6      MOV TL0,#0B4H                    到操作NEXT1_TIME0
7      LCALL KS                          13      ;如果30H等于1，则将30H清
8      JNZ K1 ;如果有按键按下，跳      零，返回中断
9      ;如果没有按键按下，判断是什      14      MOV 30H,#00H
      么情况下没有按下                  15      LJMP T0_END1

```

K1 判断按键值 K1 通过将 P1 口分别输出 #0F0H 和 #0FH，得到具体按键。存储为如下形式: 0001 0001B, 代表第一行第一列的按键。

```

16     K1:                               24     ANL A,#0FH
17     INC 30H                           25     PUSH A
18     MOV A,30H                         26
19     CJNE A,#02H,T0_END1 ;            27     MOV P1,#0F0H
      如果30H没两次，则返回            28     MOV A,P1
      中断                             29     CPL A
20     ;30H判断按下两次，开始判断      30     ANL A,#0F0H
      是哪一个被按下                  31     MOV R4,A
21     MOV P1,#0FH                       32     POP A
22     MOV A,P1                           33     ORL A,R4
23     CPL A

```

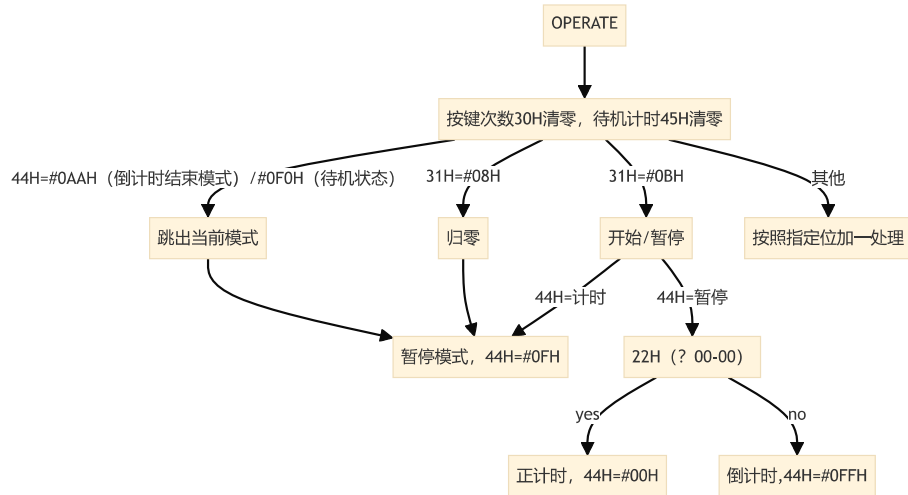
LK 按键值转换 将形如 0010 0001B 的代码转化位 4D 的十进制数，方便后续操作，具体代码略。

```

34     LK:                               44     MOV B,#04H          54     JC NEXT2
35     MOV R4,#00H                       45     MUL AB              55     INC R4
36     PUSH A                             46     MOV R5,A            56     CJNE R4,#04H,LOOP2
37     LOOP:                              47     POP ACC             57
38     RRC A                              48                       58     NEXT2:MOV A,R5
39     JC NEXT1                            49     POP ACC            59     ADD A,R4
40     INC R4                              50     SWAP A             60     MOV 31H,A
41     CJNE R4,#04H,LOOP                  51     MOV R4,#0H        61     ; 返回中断
42     NEXT1:PUSH ACC                     52     LOOP2:
43     MOV A,R4                           53     RRC A

```

OPERATE 按键操作 根据按键的值，进行相应的操作。逻辑如下图所示：

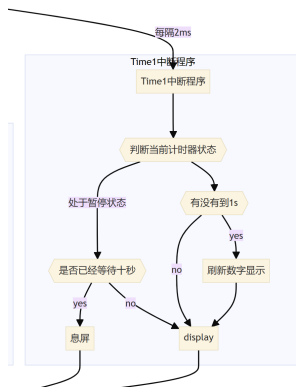


具体代码如下：

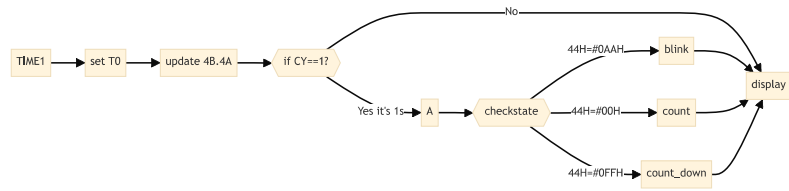
68	OPERATE:		LOOP_OPERATE	121	SETB 22H
69	MOV 30H,#00H	96	MOV R0,#53H	122	CJNE A,#00H,OPERATE3
70	MOV A,44H	97	; MOV 52H,#0DH	123	MOV A,56H
71	MOV 45H,#00H	98	MOV 55H,#0DH	124	INC A
72	CJNE A,#0AAH,	99	LJMP T0_END	125	MOV 56H,A
	NEXTXX_OPERATE	100		126	CJNE A,#06H,T0_END
73	MOV 44H,#0FH	101	; 0BH 开始/暂停	127	MOV 56H,#00H
74	LJMP T0_END	102	OPERATE1:CJNE A,#0BH,	128	SJMP
	NEXTXX_OPERATE:		OPERATE2		JUDGEM1_OPERATE
75	CJNE A,#0F0H,	103	MOV A,44H		
	NEXTX_OPERATE	104	CJNE A,#0FH,	129	OPERATE3:CJNE A,#01H,
77	; 息屏模式下		NEXT1_OPERATE1		OPERATE4
78	MOV 44H,#0FH ;调整为暂停	105	; 如果是暂停状态下	130	JUDGEM1_OPERATE:
	模式	106	JB 22H,	131	MOV A,54H
79	SETB ET1		NEXT2_OPERATE1	132	INC A
80	SETB TR1	107	MOV 44H,#00H	133	MOV 54H,A
81	MOV 42H,#0F7H ;P2口的值	108	SJMP	134	CJNE A,#0AH,T0_END
82	MOV R1,#53H		CONTINUE_OPERATE1	135	MOV 54H,#00H
83	LJMP T0_END			136	SJMP
84	NEXTX_OPERATE:	109	NEXT2_OPERATE1:		JUDGEM2_OPERATE
85	MOV A, 31H	110	MOV 44H,#0FFH		
86	;判断31H的值	111	CONTINUE_OPERATE1:	137	OPERATE4:CJNE A,#02H,
87	; 08H 归零	112	MOV TH1,#0F8H		T0_END
88	CJNE A,#08H,OPERATE1	113	MOV TL1,#030H	138	JUDGEM2_OPERATE:
89	CLR 22H	114	LJMP T0_END	139	MOV A,53H
90	MOV 44H,#0FH	115	; 如果是计时状态下	140	INC A
91	MOV R0,#53H	116	NEXT1_OPERATE1:	141	MOV 53H,A
92	LOOP_OPERATE:	117	MOV 44H,#0FH	142	CJNE A,#0AH,T0_END
93	MOV @R0,#00H	118	LJMP T0_END	143	MOV 53H,#00H
94	INC R0	119			
95	CJNE R0,#58H,	120	OPERATE2:		

3.4 定时器 TIME1 中断子程序

TIME1 中断主程序 扫描显示数码管，一个灯亮 2ms，具有四种显示状态，分别为暂停显示、正计时、倒计时、倒计时结束显示，具体逻辑如下图所示：



(a) TIME1 运行逻辑图



(b) TIME1 具体执行方式

1	;扫描显示数码管	33	;P0显示R1指向的值	64	CLR TR1
2	TIME1:	34	INC R1	65	CLR ET1
3	PUSH A	35	CJNE R1,#58H,NEXT_TIME1	66	AJMP T1_END
4	PUSH PSW	36	MOV R1,#53H	67	NEXT3_TIME1:
5	;扫描显示数码管, 一个灯亮2ms	37		68	SJMP DISPLAY
6	MOV TH1,#0F8H	38	NEXT_TIME1:	69	;当前状态为倒计时
7	MOV TL1,#030H	39	MOV A,44H	70	NEXT2_TIME1:
8		40	CJNE A,#0AAH,	71	POP A
9	MOV A,4AH		NEXT6_TIME1	72	;如果没进位
10	ADD A,#01H	41	POP A	73	JZ DISPLAY
11	MOV 4AH,A	42	JZ NEXT7_TIME1	74	MOV A,44H
12	MOV A,4BH	43	MOV A,46H	75	;当前状态为正计时
13	ADDC A,#00H	44	CPL A	76	JZ ZHENG_OPERATE
14	MOV 4BH,A	45	MOV 46H,A	77	;当前状态为倒计时
15	;如果进位了, 说明1S已经到了	46	NEXT7_TIME1:	78	SJMP FU_OPERATE
16	MOV A,#00H	47	MOV A,46H	79	
17	MOV ACC.0,C	48	JZ NEXTTY_TIME1	80	ZHENG_OPERATE:
18	PUSH A	49	MOV P2,#0FFH	81	LCALL ZHENG_TIME1
19	;此时栈顶是标志位C	50	NEXTTY_TIME1:	82	SJMP DISPLAY
20	JNC NEXT8_TIME1	51	LJMP DISPLAY	83	
21	MOV 4AH,#0CH	52	NEXT6_TIME1:	84	FU_OPERATE:
22	MOV 4BH,#0FEH	53	CJNE A,#0FH,NEXT2_TIME1	85	LCALL FU_TIME1
23		54	POP A	86	
24	NEXT8_TIME1:	55	JZ NEXT3_TIME1	87	DISPLAY:
25	;数码管左移	56	MOV A,45H	88	MOV A,@R1
26	MOV A,42H	57	INC A	89	MOVC A,@A+DPTR
27	RL A	58	MOV 45H,A	90	MOV P0, A
28	CJNE A,#0FEH,	59	CJNE A,#0AH,	91	
	NEXT1_TIME1		NEXT3_TIME1	92	T1_END:
29	MOV A,#0F7H	60	MOV 45H,#00H	93	POP PSW
30	NEXT1_TIME1:	61	MOV 44H,#0F0H	94	POP A
31	MOV 42H,A	62	MOV P0,#0FFH	95	RET
32	MOV P2,A	63	MOV P2,#0FFH		

ZHENG_TIME1 正计时 对存储在 53-57H 地址位置记录当前显示的数字进行加一操作, 具体代码略。

1	ZHENG_TIME1:	6	MOV 57H,#00H	11	CJNE A,#0AH,ZHENG_RET
2	MOV A,57H	7 ;逐位加一, 具体代码略	12	MOV 53H,#00H
3	INC A	8	MOV A,53H	13	ZHENG_RET:
4	MOV 57H,A	9	INC A	14	RET
5	CJNE A,#0AH,ZHENG_RET	10	MOV 53H,A		

FU_TIME1 倒计时 对存储在 53-57H 地址位置记录当前显示的数字进行减一操作，具体代码略。

1	FU_TIME1:	10	MOV 53H,A	19	MOV 42H,#0F7H
2	MOV A,57H	11	CJNE A,#0FFH,FU_RET	20	CLR P3.4
3	DEC A	12	MOV 53H,#00H	21	MOV 44H,#0AAH
4	MOV 57H,A	13	MOV 54H,#00H	22	CLR 22H
5	CJNE A,#0FFH,FU_RET	14	MOV 55H,#0DH	23	MOV 46H,#0FFH
6	MOV 57H,#09H	15	MOV 56H,#00H	24	FU_RET:
7 ;逐位减一，具体代码略	16	MOV 57H,#00H	25	RET
8	MOV A,53H	17	DINGDINGDING:		
9	DEC A	18	MOV R1,#53H		