

华中科技大学  
人工智能与自动化学院

## **微机原理拓展实验：键盘扫描**

彭杨哲

U201914634

2021 年 12 月 21 日

## 1 实验目的

- 实验仪器上包含一组 16 位的键盘，编写代码，实现对 16 位键盘的扫描。

## 2 实验内容

- 结合前面的几次实验，实现对实验仪器上 16 位键盘的扫描；
- 分别按下 F1 F16 按键，数码管上显示不同的数值，例如按下 F1，数码管显示 1，按下 F2，数码管上显示 2 按下 F16，数码管上显示 16

## 3 实验结果

### 3.1 实验代码

```
1 ;数码管显示
2 ; ?目标:
3
4 ; - A口和B口每个时钟周期都输出，B口一位0像流水灯一样循环扫描，
   A口对应的0那位输出要显示的段码
5
6 ; 编写:
7
8 ; 要有时钟周期的话，就得有delay，不需要用到8253，
   只要定时重新给8255写数据就行，先写数据，然后delay一下，
   然后再开始下一个循环
9
10 ; 每次需要写入的数据：A口：待显示的段码，B口：0循环移位(以8为周期)
11
12 ; 可以写成一个buffer，按位排序，并显示，从零开始依次移位
13
14 ; **注意:** 显示的时候应该是像BCD码一样，最多只显示到9
15
16 ;键盘读取
17
18 ; ?目标:
19
20 ; - C口随周期读入数据
21 ; - 检测是否有按键按下，检测按下的按键是哪个
22
23 ; 编写:
24
25 ; 和数码管的显示结合起来，放在一个主函数里，有等待过程，
   在等待按键输入的时候，数码管的扫描还是要继续进行
26
```

```

27 ; C口输入, 只读PC0和PC1
28
29 CTRL_WORD EQU 10001001B;8255的控制字
30 PORT_CTRL EQU 0273H;8255的控制端口地址
31 PORT_A EQU 0270H;A口地址
32 PORT_B EQU 0271H;B口地址
33 PORT_C EQU 0272H;C口地址
34
35
36 _STACK SEGMENT STACK ;堆栈段
37     DW 100 DUP(?) ;定义100个字空间
38 _STACK ENDS
39
40 DATA SEGMENT
41     ;定义数据缓冲区, 格式是十进制的格式, 显示范围为0-9,
42     ;一个数字对应一个七段数码管
43     Buffer DB 16,16,16,16
44             DB 16,16,16,16
45     ;数码管的段码, 从0-9, 为1的位熄灭(需要取个反)
46     Table_1 DB 0C0H,0F9H,0A4H,0BOH,099H
47             DB 92H,82H,0F8H,80H,98H
48             DB 88H,83H,0C6H,0A1H,86H
49             DB 8EH,0FFH
50     cnt DB 0;记录当前显示到第几个数码管
51 DATA ENDS
52
53 CODE SEGMENT
54 MAIN PROC NEAR
55     ASSUME CS:CODE, DS:DATA, SS:_STACK;定义堆栈段地址
56     MOV AX,DATA;加载数据段地址
57     MOV DS,AX;将数据段地址加载到DS
58     MOV ES,AX;设置ES为DATA的地址
59     MOV AX,_STACK;加载堆栈段地址
60     MOV SS,AX;设置SS为STACK的地址
61     CALL Display_
62 MAIN ENDP
63
64 ;数码管显示
65 ;@IN: Buffer里存的要显示的数码管的值(十进制)
66 ;@OUT: 无
67 Display_ PROC NEAR
68     ;A口方式0输出, B口方式0输出, C口输入
69     MOV AL, CTRL_WORD;控制字
70     MOV DX,PORT_CTRL;8255控制字地址
71     OUT DX,AL;写控制字
72
73     ;读行, 看是否所有的键均松开(如果有键按下的话,
74     ;则所有的数码管全部亮起(8888...))
75     MOV DX,PORT_B;B口地址
76     MOV AL,00H;全部送零

```

```

76     OUT DX,AL
77     MOV DX,PORT_A;A口地址
78     MOV AL,00H;段码全部送0, 可以改成全部送1, 那就是全都不亮, 都是灭的
79     OUT DX,AL
80     MOV DX,PORT_C;C口地址
81     WAIT_OPEN:
82     IN AL,DX;键盘状态读入C口
83     AND AL,03H;只看后两位
84     CMP AL,03H;是否都为1(各键均松开)
85     JNE WAIT_OPEN;否, 继续查
86     ;直到所有的键都松开了才开始循环
87
88 LOOP_1:
89     MOV AX,0000H
90     MOV SI,AX;把AX当做周期为8的扫描计数器
91
92 LABEL_1:
93     MOV BX,OFFSET Buffer;将Buffer首地址给AX
94     MOV BL,SI[BX];取对应的要写的数值到BL, 也是输入函数的参数
95     MOV CX,SI;函数的参数
96     CALL Write_AB;写A口和B口
97     ;这里AX,BX,DX都可以随便用, 但SI不能
98
99     CALL Scan;扫描按键, 同时处理扫描的延时的问题
100
101     ; MOV CX,5;写第5位数码管
102     ; MOV BX,10;全不显示
103     ; CALL Write_AB;消隐
104
105     ; MOV CX,6;写第5位数码管
106     ; MOV BX,10;全不显示
107     ; CALL Write_AB;消隐
108
109     INC SI;扫描计数器加一
110     CMP SI,8;到8的时候重新开始计数
111     JE LOOP_1;从零重新开始计数
112     JMP LABEL_1;加一
113     RET
114 Display_ ENDP
115
116 ;扫描按键
117 ;@IN: CX: 扫描码的第几位为零
118 ;@OUT: Buffer里存相应的扫描到的数字(只写最高两位);
119 Scan PROC NEAR
120     PUSH AX;保护AX, 后面要用
121     PUSH BX;保护BX, 后面要用
122     PUSH DX;保护DX, 后面要用与读写端口
123     PUSH SI;保护SI, 后面要用
124
125     ;查列是否有0, 即是否有键压下
126     MOV DX,PORT_C;

```

```

127     IN AL,DX;读C口
128     AND AL,03H;只看后两位
129     CMP AL,03H;是否有键压下
130     JE DELAY_RETURN;无,先延时一下然后直接跳出函数
131     ;有键压下,延时一段时间,消除抖动
132     CALL Delay_Longer
133     ;再查列,看键盘是否仍被压下
134     IN AL,DX;读C口
135     AND AL,03H;只看后两位
136     CMP AL,03H;是否有键压下
137     JE RETURN;已松开,只是一次小抖动,不是按键压下,因为已经延时过了,
        所以直接跳到返回

138
139     ;消隐所有位,方式二有用
140     ;MOV BX,8
141     ;LOOP_2:
142     ;    MOV BYTE PTR Buffer[BX-1],16;消隐所有段码,因为偏移地址是0-7,
        所以要减一
143     ;    DEC BX;减一
144     ;    JNZ LOOP_2
145
146     ;键仍被压着,确定哪一个键被压着
147     MOV BX,CX
148     TEST AL,01H;KL1那两行
149     JE RESULT
150     ADD CX,8;KL2那两行
151     RESULT:
152     NOP;结果就是BX里是哪一位为零(也就是哪一位数码管亮),
        CX里是实际的要显示的数字

153
154     ;方式一:只在前两位显示,显示成0-9
155     ;MOV BX,OFFSET Buffer;将Buffer首地址给BX
156     ;CMP CX,9;如果大于9就要用两位数码管来显示
157     ;JLE LABEL_2;如果不大于9,十位放0,个位放1
158     ;MOV BYTE PTR 7[BX],1;大于9的话,先把buffer的十位放一
159     ;SUB CX,10;buffer的个位
160     ;JMP LABEL_3
161     ;LABEL_2:
162     ;MOV BYTE PTR 7[BX],10;小于9的话,也把buffer的十位放10(不显示)
163     ;LABEL_3:
164     ;MOV BYTE PTR 6[BX],CL;写入buffer的个位
165     ;JMP RETURN;不延时,直接返回
166
167     ;方式二:按下哪一位在哪位显示0-F
168     ;MOV Buffer[BX],CL
169
170     ;方式三:按下一位后就多显示一个数码管
171     XOR BX,BX;BX清零
172     MOV BL,cnt;读入全局显示计数器
173     CMP BL,8
174     JB NORMAL

```

```

175     MOV Buffer[0],CL
176     JMP NEXT_STEP
177     NORMAL:
178     MOV Buffer[BX],CL
179
180     NEXT_STEP:
181     INC BL;加一
182     CMP BL,9;如果小于9,就是正常的,保存下就好
183     JB STORE_CNT
184
185
186     ;消隐所有位
187     MOV BX,8
188     LOOP_2:
189         MOV BYTE PTR Buffer[BX-1],16;消隐所有段码,因为偏移地址是0-7,
            所以要减一
190         DEC BX;减一
191     JNZ LOOP_2
192
193
194     MOV BL,0;显示到了最高一个数码管,重新从最低位数码开始显示
195     STORE_CNT:
196     MOV cnt, BL
197
198     ;等待键全部松开之后才再次进入主循环,才能再次进入scan函数
199     ;显示过程
200     MOV AH,1;有键按下的标志,1是按下
201     LOOP_1_:
202
203     CMP AH,1;全部扫描过程中都没有键按下,才跳出去
204     JNE RETURN
205
206     MOV AH,0;先假设全部键都没按下
207     MOV SI,0000H;把si当做周期为8的扫描计数器
208
209     LABEL_1_:
210     MOV BX,OFFSET Buffer;将Buffer首地址给AX
211     MOV BL,SI[BX];取对应的要写的数值到BL,也是输入函数的参数
212     MOV CX,SI;函数的参数
213     CALL Write_AB;写A口和B口
214
215     INC SI;扫描计数器加一
216     CMP SI,9;到8的时候重新开始计数
217     JE LOOP_1_;从零重新开始计数
218
219     CALL Delay;
220
221     IN AL,DX;键盘状态读入C口
222     AND AL,03H;只看后两位
223     CMP AL,03H;是否都为1(各键均松开)
224     JE NEXT;这两个按键里没有键按下,就不用管,AH不做改变

```

```

225     MOV AH,1;不等于11, 有键按下了
226     NEXT:
227     JMP LABEL_1_;加一
228
229     DELAY_RETURN:
230     CALL Delay;相当于是扫描的延时
231
232     RETURN:
233     POP SI;恢复SI
234     POP DX;恢复DX
235     POP BX;恢复BX
236     POP AX;恢复AX
237     RET
238 Scan ENDP
239
240
241 ;写A口和B口的数据
242 ;@IN: CX: 要写第几位数码管(高位置零); BX: 要写的数字(十进制,高位置零)
243 ;@OUT: 无
244 Write_AB PROC NEAR
245     PUSH AX;保护AX, 后面要用
246     PUSH SI;保护SI, 后面要用
247     PUSH DX;保护DX, 后面要用
248
249     ;AL: A口要写的的数据, AH: B口要写的的数据
250     MOV SI,BX;函数输入
251     CALL Offset2Seg;调用函数
252     MOV AH,0FEH;第一个扫描码
253     ROL AH,CL;注意是CL, 不是CX
254
255     ;写A口
256     MOV DX,PORT_A;A口地址
257     OUT DX,AL;输出段码
258     ;写B口
259     MOV DX,PORT_B;B口地址
260     XCHG AH,AL;把B口要写的的数据换到AL来
261     OUT DX,AL;输出扫描码
262
263
264     POP DX;恢复DX
265     POP SI;恢复SI
266     POP AX;恢复AX
267     RET
268 Write_AB ENDP
269
270 ;把偏移地址转成对应的数码管的段码
271 ;@IN: SI: 偏移地址
272 ;@OUT: AL: 段码(AH不动); SI: 同输入
273 Offset2Seg PROC NEAR
274     PUSH BX;保护一下bx
275     MOV BX,OFFSET Table_1;将七段数码管代码表首地址给BX

```

```

276     MOV AL,SI[BX];取对应的段码到AL
277     POP BX;弹出BX
278     RET
279 Offset2Seg ENDP
280
281 ;延时一段较长的时间，用来防抖动
282 ;@IN:无
283 ;@OUT:无
284 Delay_Longer PROC NEAR
285     PUSH CX;保存CX值
286     MOV CX,1000H;一段稍长的时间
287     LOOP $ ;在此行循环
288     POP CX;恢复CX值
289     RET;返回
290 Delay_Longer ENDP
291
292 ;延时一段时间，当做一个时间周期
293 ;@IN:无
294 ;@OUT:无
295 Delay PROC NEAR
296     PUSH CX;保存CX值
297     MOV CX,100H;一段比较短的时间
298     LOOP $ ;在此行循环
299     POP CX;恢复CX值
300     RET;返回
301 Delay ENDP
302
303 CODE ENDS
304 END MAIN

```

## 4 实验扩展及思考

1. 绘制本实验的详细软件流程图  
答: 如图1、图2、图3所示
2. 结合前面几次的实验内容，给出本次实验的连线表  
答: 如表1所示
3. 扼要注释每指令的功能  
答: 见实验代码部分



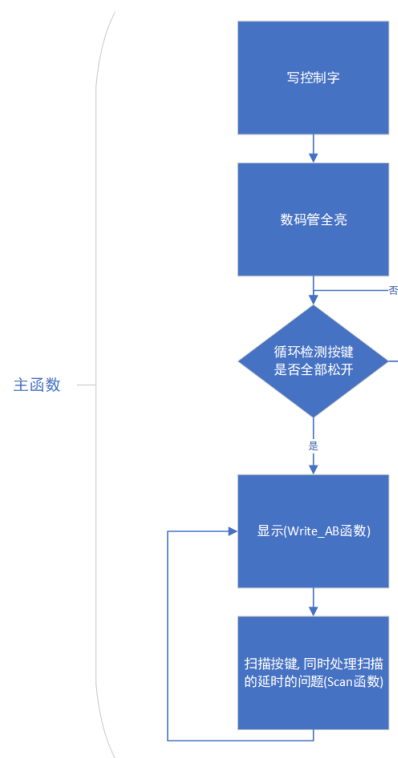


Figure 1: 主函数流程图

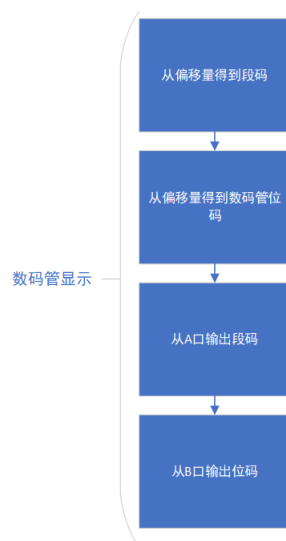


Figure 2: 数码管显示流程图

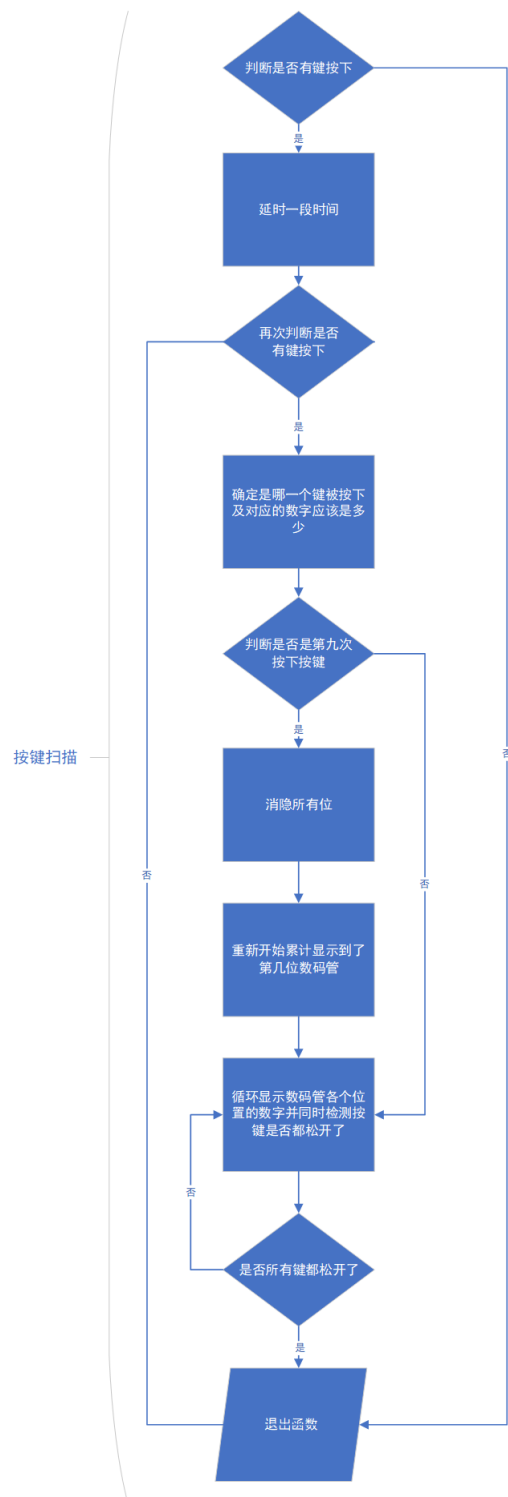


Figure 3: 按键扫描流程图

功能连线	<b>F5 区</b>	<b>D3 区</b>
键盘	D(按键行线, 要求读入 KL1 和 KL2)	JP24(键盘行输入 PC0 和 PC1)
	A(按键列线, 要求就是一位 0 像流水灯一样扫描, 要是输出)	JP20(键盘列输出, PB 口直接输出, PB 输出一位 0 像流水灯一样扫描)
数码管	B(数码管段码, 要求是输出的段码)	B(数码管段码, 相当于 PA 口的输出取反)
	C(数码管选择脚, 要求就是一位 0 像流水灯一样扫描, 要是输出)	C(数码管段选, 就是 PB 口的直接输出 (这里画的有点问题, 244 不取反))
CPU 与 8255 通讯连线	<b>D3 区</b>	<b>A3 区</b>
	CS	CS1
	A0	A0
	A1	A1

Table 1: 连线表