嵌入式Proteus实验三 实验报告

汪润中 515021910285

源代码

**文件名：Main.c**

1. /\* Main.c file generated by New Project wizard
2. \*
3. \* Created: 周三 4月 12 2017
4. \* Processor: 8086
5. \* Compiler: Digital Mars C
6. \*
7. \* Before starting simulation set Internal Memory Size
8. \* in the 8086 model properties to 0x10000
9. \*/
10. #include "header.h"
11. static unsigned int DelayShort **=** 40**;**
12. static unsigned int DelayLong **=** 4000**;**
13. static unsigned char SEGTAB**[**16**]={**
14. 3FH**,** // 7-Segment Tube, 共阴极类型的7段数码管示意图
15. 06H**,**
16. 5BH**,** // a a a
17. 4FH**,** // f b
18. 66H**,** // f b
19. 6DH**,** // f b
20. 7DH**,** // g g g
21. 07H**,** // e c
22. 7FH**,** // e c
23. 6FH**,** // e c
24. 77H**,** // d d d h h h
25. 7CH**,** // ----------------------------------
26. 39H**,** // b7 b6 b5 b4 b3 b2 b1 b0
27. 5EH**,** // DP g f e d c b a
28. 79H**,** //
29. 71H
30. **};**
31. static unsigned char studentNo **=** 5Dh**;**
32. static unsigned int displayPointer **=** 0**;**
33. static unsigned int toDisp**[**4**];**
34. void main**(**void**)**
35. **{**
36. // Write your code here
37. \_\_asm
38. **{**
39. IN AL**,** 0 //Simulation Patch for Proteus, please ignore this line
40. **}**
41. //service initiate
42. INT\_INIT**();**
43. INIT8255**();**
44. INIT8253**();**
45. //endless loop
46. **while** **(**1**)**
47. **{**
48. DISPLAY8255**();**
49. AccessPC**();**
50. FeedDog**();**
51. **}**
52. **}**

55. void INIT8255**(**void**)**
56. **{**
57. \_\_asm
58. **{**
59. // Init 8255 in Mode x, L8255PA xPUT, L8255PB xPUT, L8255PCU xPUT, L8255PCL xPUT
60. // 计算8255的控制字
61. // Init 8255 in Mode 0, L8255PA Output, L8255PB Output, L8255PC UP Input, L8255PC LOW Output
62. MOV DX**,** L8255CS
63. MOV AL**,** 81h
64. OUT DX**,** AL
65. **}**
66. **}**
67. void INIT8253**(**void**)**
68. **{**
69. \_\_asm
70. **{**
71. //设定Timer0的初始值
72. MOV DX**,** L8253CS
73. MOV AL**,** 36h //Select timer0, write two bytes(LSB first), mode 3
74. OUT DX**,** AL
75. MOV DX**,** L8253T0
76. MOV AL**,** 10h
77. OUT DX**,** AL
78. MOV AL**,** 27h
79. OUT DX**,** AL
80. //设定Timer1的初始值
81. MOV DX**,** L8253CS
82. MOV AL**,** 76h //Select timer1, write two bytes(LSB first), mode 3
83. OUT DX**,** AL
84. MOV DX**,** L8253T1
85. MOV AL**,** 64h
86. OUT DX**,** AL
87. MOV AL**,** 0
88. OUT DX**,** AL
89. //设定Timer2的初始值
90. MOV DX**,** L8253CS
91. MOV AL**,** 0B0h //Select timer2, write two bytes(LSB first), mode 0
92. OUT DX**,** AL
93. MOV DX**,** L8253T2
94. MOV AL**,** 5h //10ms period check
95. OUT DX**,** AL
96. MOV AL**,** 0
97. OUT DX**,** AL
98. **}**
99. **}**
100. void FeedDog**(**void**)**
101. **{**
102. //看门狗喂狗程序，周期30ms
103. \_\_asm
104. **{**
105. MOV DX**,** L8253T2
106. MOV AL**,**15
107. OUT DX**,**AL
108. MOV AL**,**0
109. OUT DX**,**AL
110. **}**
111. **}**
112. void AccessPC**(**void**)**
113. **{**
114. \_\_asm
115. **{**
116. MOV DX**,** L8255PC // DX指向C端口
117. IN AL**,** DX // 读取C口数值
118. MOV AH**,** AL // Save it
119. AND AH**,** 0BFh // 清除PC6
120. AND AL**,** 01h // Test bit 0
121. JZ APCOut
122. OR AH**,** 40h // 置位PC6
123. APCOut**:**
124. MOV AL**,** AH // Restore and Modify
125. OUT DX**,** AL // 更新PC6
126. **}**
127. **}**
128. void DISPLAY8255**(**void**)**
129. **{**
130. //Uncomment codes in this section to enable 3.3
131. //displayPointer = displayPointer % 4;
132. //switch(displayPointer)
134. //Comment following codes to disable 3.5
135. displayPointer **=** displayPointer **%** 10000**;**
136. toDisp**[**0**]** **=** displayPointer **%** 10**;**
137. toDisp**[**1**]** **=** **(**displayPointer **%** 100**)/**10**;**
138. toDisp**[**2**]** **=** **(**displayPointer **%** 1000**)/**100**;**
139. toDisp**[**3**]** **=** displayPointer **/** 1000**;**

142. **{**
143. //case 0:
144. \_\_asm
145. **{**
146. **;** 点亮第一个七段管
147. MOV DX**,** L8255PA **;** 指向PA
148. MOV AL**,** 0FEh **;** 设定第一个七段管的PA值
149. OUT DX**,** AL **;** 输出到PA
150. MOV BL**,** toDisp**+**6
151. MOV BH**,** 0
152. ADD BX**,** OFFSET SEGTAB
153. MOV AL**,** **[**BX**]**
154. MOV DX**,** L8255PB **;** 指向PB
155. OUT DX**,** AL **;** 输出字模到PB口，点亮
156. CALL DELAY **;** 延时
157. **}**
158. //break;
159. //case 1:
160. \_\_asm
161. **{**
162. **;** 点亮第二个七段管
163. MOV DX**,** L8255PA **;** 指向PA
164. MOV AL**,** 0FDh **;** 设定第一个七段管的PA值
165. OUT DX**,** AL **;** 输出到PA
166. MOV BL**,** toDisp**+**4
167. MOV BH**,** 0
168. ADD BX**,** OFFSET SEGTAB
169. MOV AL**,** **[**BX**]**
170. MOV DX**,** L8255PB **;** 指向PB
171. OUT DX**,** AL **;** 输出字模到PB口，点亮
172. CALL DELAY **;** 延时
173. **}**
174. //break;
176. //case 2:
177. \_\_asm
178. **{**
179. **;** 点亮第三个七段管
180. MOV DX**,** L8255PA **;** 指向PA
181. MOV AL**,** 0FBh **;** 设定第一个七段管的PA值
182. OUT DX**,** AL **;** 输出到PA
183. MOV BL**,** toDisp**+**2
184. MOV BH**,** 0
185. ADD BX**,** OFFSET SEGTAB
186. MOV AL**,** **[**BX**]**
187. MOV DX**,** L8255PB **;** 指向PB
188. OUT DX**,** AL **;** 输出字模到PB口，点亮
189. CALL DELAY **;** 延时
190. **}**
191. //break;
193. //case 3:
194. \_\_asm
195. **{**
196. **;** 点亮第四个七段管
197. MOV DX**,** L8255PA **;** 指向PA
198. MOV AL**,** 0F7h **;** 设定第一个七段管的PA值
199. OUT DX**,** AL **;** 输出到PA
200. MOV BL**,** toDisp
201. MOV BH**,** 0
202. ADD BX**,** OFFSET SEGTAB
203. MOV AL**,** **[**BX**]**
204. MOV DX**,** L8255PB **;** 指向PB
205. OUT DX**,** AL **;** 输出字模到PB口，点亮
206. CALL DELAY **;** 延时
207. **}**
208. //break;
209. //default:break;
210. **}**
211. **}**
212. void DELAY**(**void**)**
213. **{**
214. \_\_asm
215. **{**
216. PUSH CX
217. MOV CX**,** DelayShort
218. D1**:** LOOP D1
219. POP CX
220. **}**
221. **}**
222. void INT\_INIT**(**void**)**
223. **{**
224. \_\_asm
225. **{**
226. CLI // Disable interrupt
227. MOV AX**,** 0
228. MOV ES**,** AX // 准备操作中断向量表
230. //!!!!!!!!!!!!!!!!!!!!!!!!!!!
231. MOV AX**,** OFFSET MYIRQ //! Incorrect if using LEA!
232. //！ dont know why
233. //!!!!!!!!!!!!!!!!!!!!!!!!!!!
235. MOV BL**,** studentNo //calculate interrupt offset
236. MOV BH**,** 0
237. SHL BX**,** 1
238. SHL BX**,** 1
239. MOV ES**:[**BX**],** AX //write interrupt service address into memory, IP first
240. MOV AX**,** CS
241. INC BX
242. INC BX
243. MOV ES**:[**BX**],** AX //then CS
244. STI //enable interrupt
245. **}**
246. **}**
247. void \_interrupt \_far MYIRQ**(**void**)**
248. **{**
249. \_\_asm**{**
250. //LED display
251. MOV DX**,** L8255PC
252. IN AL**,**DX
253. XOR AL**,** 80h
254. OUT DX**,** AL
256. INC displayPointer //upload display content
257. **}**
258. **}**

**文件名：Header.h**

1. #define L8253T0 100h //Even Timer0's address in I/O space, 108H as well.
2. //101H, 109H for high order byte access.
3. #define L8253T1 102h // Timer1's address in I/O space
4. // 103H, 10BH for high order byte access.
5. #define L8253T2 104h // Timer2'saddress in I/O space
6. // 105H, 10DH for high order byte access.
7. #define L8253CS 106h // 8253 Control Register's address in I/O space
8. // 107H, 10FH for high order byte access.
9. // 8255芯片端口地址 （Port Address):
10. #define L8255PA 121h // Odd Port A's address in I/O space, 129H as well.
11. // 120H, 128H for low order byte access.
12. #define L8255PB 123h // Odd Port B's address in I/O space, 12BH as well.
13. // 122H, 12AH for low order byte access.
14. #define L8255PC 125h // Odd Port C's address in I/O space, 13DH as well.
15. // 124H, 12CH for low order byte access.
16. #define L8255CS 127h // 8255 Control Register's port number in I/O space, 12FH as well
17. // 126H, 12EH for low order byte access.
18. // 中断矢量号定义
19. #define IRQNum 20h // 中断矢量号,要根据学号计算得到后更新此定义。
20. void DELAY**(**void**);**
21. void INIT8255**(**void**);**
22. void INIT8253**(**void**);**
23. void AccessPC**(**void**);**
24. void DISPLAY8255**(**void**);**
25. void INT\_INIT**(**void**);**
26. void \_interrupt \_far MYIRQ**(**void**);**
27. void FeedDog**(**void**);**

**文件名：rtl.asm**

1. ;====================================================================
2. ; rtl.asm file generated by New Project wizard
3. ;
4. ; Created: 周三 4月 12 2017
5. ; Processor: 8086
6. ; Compiler: Digital Mars C
7. ;====================================================================
8. .MODEL **SMALL**
9. .8086
10. .stack
11. .code
12. extern \_main**:near**
13. .startup
14. **cli** ; interrupt disable
15. **call** **near** ptr \_main
16. endless**:**
17. **jmp** endless
18. .data
19. public \_\_acrtused ; trick to force in startup
20. \_\_acrtused **=** 9876h ; funny value not easily matched ; ; in SYMDEB
21. END

实验感想

* 1. C & assembly混编流程
     1. 检查编译器Digital Mars C是否正确安装
     2. 如果当前编译器是MASM32，先备份代码；之后在Source code界面中，右键project名称，删除当前代码
     3. 回到schematic界面，在8086芯片上右键，选择edit source code，在界面中选择Digital Mars C作为编译器
     4. 在main.c文件中修改代码
     5. 在.c文件中，任意处可以插入\_\_asm{ ::code:: }。如果在汇编代码中更改了寄存器值，可能需要PUSH和POP
  2. 所有变量建议设置成static全局变量，在汇编中可以直接使用变量名
  3. unsigned char可认为是byte
  4. c & assembly混编时，二进制表示如00001111b编译不通过；十六进制0FFh和0xFF都通过
  5. c语言中，中断语法为
     1. 使用汇编语言设置中断矩阵

void set\_int(unsigned char int\_no, void \* service\_proc)

{ \_asm

{ push es

xor ax, ax

mov es, ax

mov al, int\_no

xor ah, ah

shl ax, 1

shl ax, 1

mov si, ax

mov ax, service\_proc

mov es:[si], ax

inc si

inc si

mov bx, cs

mov es:[si], bx

pop es

}

}

* + 1. 中断服务程序声明使用void \_interrupt \_far FUNCTION\_NAME(void)。FUNCTION\_NAME可以作为service\_proc变量输入set\_int
  1. C中的函数名可以直接使用，利用OFFSET可以获得IP偏移量