

```

1      NAME DISPLAY
2      ;
3      ;
4      ;
5      ;
6      ;
7      ;
8      ;
9      ;
10     ;
11     ;
12     ;
13     ; DisplayTimerInit
14     ;
15     ; Description:      This function initializes the display timer event handler
16     ;                  and display_string.
17     ; Operation:       It zeroes out all 8 chars in display_string and
18     ;                  sets the timer event handler to the function pointer of
19     ;                  DisplayTimerEventHandler.
20     ;
21     ; Arguments:       None.
22     ;
23     ; Return Value:    None.
24     ;
25     ; Local Variables: None.
26     ;
27     ; Shared Variables: display_string
28     ;                  currentIndex
29     ;
30     ; Global Variables: None.
31     ;
32     ; Input:           None.
33     ;
34     ; Output:          Display
35     ;
36     ; Error Handling:  None.
37     ;
38     ; Algorithms:      None.
39     ;
40     ; Data Structures: None.
41     ;
42     ; Registers Used:  None.
43     ;
44     ; Stack Depth:     .
45     ;
46     ; Author:          Archan Luhar
47     ; Last Modified:   11/04/2013
48     ;
49     ;
50     ; Pseudo Code
51     ; -----
52     ;     currentIndex = 0
53     ;     display_string = 0 dup 8
54     ;     Set Timer 2 registers to interrupt every 1/1000 second.
55     ;     Interrupt Table [ timer 2 ] = DisplayTimerEventHandler
56     ;
57     ;

```

```

58
59 ; DisplayTimerEventHandler
60 ;
61 ; Description:      This function is handles timer interrupts and outputs
62 ;                  to the display what is stored at display_string shared
63 ;                  memory.
64 ;
65 ; Operation:
66 ;
67 ; Arguments:        display_string shared variable
68 ;
69 ; Return Value:     None.
70 ;
71 ; Local Variables:  None.
72 ;
73 ; Shared Variables: display_string
74 ;                  currentIndex
75 ;
76 ; Global Variables: None.
77 ;
78 ; Input:            None.
79 ;
80 ; Output:           Display
81 ;
82 ; Error Handling:   None.
83 ;
84 ; Algorithms:       None.
85 ;
86 ; Data Structures:  None.
87 ;
88 ; Registers Used:   None.
89 ;
90 ; Stack Depth:      .
91 ;
92 ; Author:           Archan Luhar
93 ; Last Modified:    11/04/2013
94 ;
95 ;
96 ; Pseudo Code
97 ; -----
98 ;     FIRST_DISPLAY_UNIT = 0
99 ;     NUM_DISPLAY_DIGITS = 8
100 ;
101 ;     OUT (FIRST_DISPLAY_UNIT + currentIndex), display_string[currentIndex]
102 ;     currentIndex = (currentIndex + 1) mod NUM_DISPLAY_DIGITS
103 ;
104 ;
105 ;
106 ; Display
107 ;
108 ; Description:      This function is used to display a <null> terminated string
109 ;                  to the LED display on the target board.
110 ;
111 ; Operation:        This function goes through each character of the string
112 ;                  located at ES:SI by checking if it is a letter or digit
113 ;                  and then writing the corresponding led display bits from a
114 ;                  array to a space in memory dedicated to storing the current

```

```

115 ;           display. A timer interrupt handler will take care of
116 ;           actually OUTputting to the several display pieces the
117 ;           letter and digit "codes".
118 ;
119 ; Arguments:      SI - the offset from ES which is the location of the the
120 ;                 string.
121 ;
122 ; Return Value:   None.
123 ;
124 ; Local Variables: None.
125 ;
126 ; Shared Variables: SEGTAB14 - LED pattern codes table for ASCII characters
127 ;                     display_string - a portion of the memory dedicated to
128 ;                                     storing the exact representation of the
129 ;                                     currently displayed characters. The timer
130 ;                                     event handler will display what is in this
131 ;                                     location.
132 ;
133 ; Global Variables: None.
134 ;
135 ; Input:          None.
136 ; Output:         A timer interrupt handler will output to the LED display
137 ;                 the ASCII-display binary translated digits and letters.
138 ;
139 ; Error Handling:  None.
140 ;
141 ; Algorithms:     None.
142 ;
143 ; Data Structures: Array.
144 ;
145 ; Registers Used:  None.
146 ;
147 ; Stack Depth:    .
148 ;
149 ; Author:         Archan Luhar
150 ; Last Modified:  11/04/2013
151 ;
152 ;
153 ; Pseudo Code
154 ; -----
155 ;     i = 0;
156 ;     while true:
157 ;         char = string[i]
158 ;         if char == ASCII_NULL:
159 ;             display_string[i] = DISPLAY_STRING_END_CHAR
160 ;             break;
161 ;
162 ;         display_string[i] = SEGTAB14[char];
163 ;
164 ;         i++;
165 ;
166 ;
167 ;
168 ;
169 ; DisplayNum
170 ;
171 ; Description:     This function is used to display a decimal number to

```

```

172 ;           the LED display.
173 ;
174 ; Operation:   This function simply calls Dec2String to get the ASCII
175 ;             representation of the number and then call Display
176 ;             to display the ascii representation of the number.
177 ;
178 ; Arguments:   AX - number to display.
179 ;
180 ; Return Value: None.
181 ;
182 ; Local Variables: None.
183 ;
184 ; Shared Variables: None.
185 ; Global Variables: None.
186 ;
187 ; Input:       None.
188 ; Output:      None.
189 ;
190 ; Error Handling: None.
191 ;
192 ; Algorithms:  None.
193 ;
194 ; Data Structures: None.
195 ;
196 ; Registers Used: None.
197 ;
198 ; Stack Depth: .
199 ;
200 ; Author:      Archan Luhar
201 ; Last Modified: 11/04/2013
202 ;
203 ;
204 ; Pseudo Code
205 ; -----
206 ;   Display(Dec2String(n))
207 ;
208 ;
209 ;
210 ; DisplayHex
211 ;
212 ; Description: This function is used to display a hexadecimal number to
213 ;             the LED display.
214 ;
215 ; Operation:   This function simply calls Hex2String to get the ASCII
216 ;             representation of the number and then call Display
217 ;             to display the ascii representation of the number.
218 ;
219 ; Arguments:   AX - number to display.
220 ;
221 ; Return Value: None.
222 ;
223 ; Local Variables: None.
224 ;
225 ; Shared Variables: None.
226 ; Global Variables: None.
227 ;
228 ; Input:       None.

```

```
229 ; Output:          None.
230 ;
231 ; Error Handling:   None.
232 ;
233 ; Algorithms:       None.
234 ;
235 ; Data Structures:  None.
236 ;
237 ; Registers Used:   None.
238 ;
239 ; Stack Depth:      .
240 ;
241 ; Author:           Archan Luhar
242 ; Last Modified:    11/04/2013
243 ;
244 ;
245 ; Pseudo Code
246 ; -----
247 ;   Display(Hex2String(n))
248
```