

```

1. ;=====
2. ; Main.asm file generated by New Project wizard
3. ;
4. ; Created:   [Current Date]
5. ; Processor: AT89C52
6. ; Compiler:  ASEM-51 (Proteus)
7. ;=====
8.
9. $NOMOD51
10. $INCLUDE (80C52.MCU)
11.
12. ;=====
13. ; DEFINITIONS
14. ;=====
15.
16. TOP equ P2.7      ; Input A
17. BOTTOM equ P2.6    ; Input B
18. OUTPUT equ P1.7    ; Output C
19.
20. RS equ P1.0       ; Command/Data
21. RW equ P1.1       ; Read/Write
22. EN equ P1.2       ; Latch enable
23.
24.
25. ;=====
26. ; RESET and INTERRUPT VECTORS
27. ;=====
28.
29.     ; Reset Vector
30.     org 0000h
31.     jmp Start
32.
33. ;=====
34. ; CODE SEGMENT
35. ;=====
36.
37. Start: CLR TOP      ; Initially turn off P2.7
38.        CLR BOTTOM   ; Initially turn off P2.6
39.        CLR OUTPUT   ; Initially turn off Motor
40.        mov A, #38h ;set up 2 line, 5x7 matrix display
41.        Acall cmd
42.        mov A, #0Fh ;Display ON, Cursor ON, Cursor blinking
43.        Acall cmd
44.        mov A, #01h ;Clear the display
45.        Acall cmd
46.        mov A, #06h ;Cursor increment mode(left to right)
47.        Acall cmd
48.        mov A, #80h ;Cursor Home(line1, position1)
49.        Acall cmd
50.
51.
52.
53. MAIN_LOOP:
54.     JB TOP, STOP      ; If A (P1.0) is 1, jump to STOP
55.     JNB BOTTOM, RUN    ; If B (P1.1) is 0, jump to RUN
56.
57.     SJMP MAIN_LOOP    ; Continue Looping
58.

```

```
59. STOP:
60.     CLR OUTPUT          ; Turn off Motor
61.     Acall Full
62.     SJMP MAIN_LOOP      ; Return to main Loop
63.
64. RUN:
65.     SETB OUTPUT         ; Turn on Motor
66.     Acall Loww
67.     SJMP MAIN_LOOP      ; Return to main Loop
68.
69. Loww:  mov A, #'T'
70.     Acall disp
71.     mov A, #'A'
72.     Acall disp
73.     mov A, #'N'
74.     Acall disp
75.     mov A, #'K'
76.     Acall disp
77.     mov A, #' '
78.     Acall disp
79.     mov A, #'L'
80.     Acall disp
81.     mov A, #'O'
82.     Acall disp
83.     mov A, #'W'
84.     Acall disp
85.
86.
87.
88. On:  mov A, #01h ;Clear the display
89.     Acall cmd
90.     mov A, #80h ;Cursor Home(line1, position1)
91.     Acall cmd
92.     mov A, #'M'
93.     Acall disp
94.     mov A, #'O'
95.     Acall disp
96.     mov A, #'T'
97.     Acall disp
98.     mov A, #'O'
99.     Acall disp
100.    mov A, #'R'
101.    Acall disp
102.    mov A, #' '
103.    Acall disp
104.    mov A, #'O'
105.    Acall disp
106.    mov A, #'N'
107.    Acall disp
108.
109. loop:  jnb BOTTOM, loop
110. loop3: jnb TOP, loop3
111.     ret
112.
113. Full:  mov A, #01h ;Clear the display
114.     Acall cmd
115.     mov A, #80h ;Cursor Home(line1, position1)
116.     Acall cmd
```

```
117.     mov A, #'T'
118.     Acall disp
119.     mov A, #'A'
120.     Acall disp
121.     mov A, #'N'
122.     Acall disp
123.     mov A, #'K'
124.     Acall disp
125.     mov A, #' '
126.     Acall disp
127.     mov A, #'F'
128.     Acall disp
129.     mov A, #'U'
130.     Acall disp
131.     mov A, #'L'
132.     Acall disp
133.     mov A, #'L'
134.     Acall disp
135.
136. loop1:  jb TOP, loop1
137.
138. Off:     mov A, #01h ;Clear the display
139.     Acall cmd
140.     mov A, #80h ;Cursor Home(line1, position1)
141.     Acall cmd
142.     mov A, #'M'
143.     Acall disp
144.     mov A, #'O'
145.     Acall disp
146.     mov A, #'T'
147.     Acall disp
148.     mov A, #'O'
149.     Acall disp
150.     mov A, #'R'
151.     Acall disp
152.     mov A, #' '
153.     Acall disp
154.     mov A, #'O'
155.     Acall disp
156.     mov A, #'F'
157.     Acall disp
158.     mov A, #'F'
159.     Acall disp
160.
161. loop2:  jb BOTTOM, loop2
162.     mov A, #01h ;Clear the display
163.     Acall cmd
164.     mov A, #80h ;Cursor Home(line1, position1)
165.     Acall cmd
166.     ret
167.
168. cmd:     mov P3,A
169.     clr RS ; command
170.     clr RW ; write command
171.     setb EN ; set latch enable
172.     clr EN ; clear latch enable
173.     Acall delay
174.     ret
```

```
175.  
176. disp:  mov P3,A  
177.      setb RS ;data  
178.      clr RW ;write command  
179.      setb EN ;set latch enable  
180.      clr EN ;clear latch enable  
181.      Acall delay  
182.      ret  
183.  
184. delay:  
185.      mov R1, #0FFH  
186. delay1:  
187.      mov R2, #0FFh  
188. delay2:  
189.      djnz R2, delay2  
190.      djnz R1, delay1  
191.      ret  
192.  
193. ;=====
```

```
194.      END  
195.  
196.  
197.
```