```
;*****************************************
;**** lab5 up down switch to 7 segment -> debounce
;*****************************************
        PROCESSOR PIC16F628
        #include <P16F628.INC>
        __CONFIG    _CP_OFF & _MCLRE_OFF & _HS_OSC & _LVP_OFF & _WDT_OFF

        cblock  0x20
            temp
            temp1
            count
            count0
            count1
            count2
        endc

        ORG 0x00        ;reset vector
        goto    main    ;vector to main program


main:
        call    Init

L1:
        btfsc   PORTA,0     ; is UP pressed?
        goto    L2

        movlw   .10
        call    DelayMS

        btfsc   PORTA,0
        goto    L2

        incf    temp,f

L2:     btfsc   PORTA,1     ; is DOWN pressed?
        goto    L3

        movlw   .10
        call    DelayMS

        btfsc   PORTA,1
        goto    L3

        decf    temp,f

L3      movlw   .16
        subwf   temp,w
        btfss   STATUS,Z    ;check if temp=16?
        goto    L4      ;No,check zero

        clrf    temp        ;Yes, clear 'temp' back to zero
        goto    L5          ;Repeat the infinite loop

L4:
        movlw   .255
        subwf   temp,w
        btfss   STATUS,Z    ;check if temp=255?
        goto    L5          ;No, go back and do it again

        movlw   .15
        movwf   temp        ;yes, set temp = 16
        goto    L5          ;Repeat the infinite loop


L5:     movf    temp,w      ;use [Temp] to call 'Table7seg'
        call    Table7seg
        movwf   PORTB       ;Send the obtain 7 seg pattern to PORTB
```

```
69
70          movlw   .250
71          call    DelayMS
72
73          goto    L1
74
75
76  ;Loopup table for 7segments LED Patterns
77  Table7seg:
78          addwf   PCL,F
79          ;Segments   .GFEDBA
80          retlw   B'00111111'     ;Number0
81          retlw   B'00000110'     ;Number1
82          retlw   B'01011011'     ;Number2
83          retlw   B'01001111'     ;Number3
84          retlw   B'01100110'     ;Number4
85          retlw   B'01101101'     ;Number5
86          retlw   B'01111101'     ;Number6
87          retlw   B'00000111'     ;Number7
88          retlw   B'01111111'     ;Number8
89          retlw   B'01101111'     ;Number9
90          retlw   B'01110111'     ;A
91          retlw   B'01111100'     ;B
92          ;retlw  B'01011000'     ;C little
93          retlw   B'00111001'     ;C big
94          retlw   B'01011110'     ;D
95          retlw   B'01111001'     ;E
96          retlw   B'01110001'     ;F
97          retlw   B'10000000'     ;dot-point
98
99  DelayMS:
100         movwf   count2
101         incf    count2,f
102         decfsz  count2,f
103         goto    $+2
104         goto    $+3
105         call    Delay1MS
106         goto    $-4
107         return
108
109 Delay1MS:
110         movlw   .50             ; 1 cyc
111         movwf   count1          ; 1 cyc
112 outterloop:
113         movlw   .5              ; 1 cyc * count1
114         nop                     ; 1 cyc * count1
115         movwf   count0          ; 1 cyc * count1
116 innerloop:
117         decfsz  count0,F        ; 1 cyc * count1 * count0
118         goto    innerloop       ; 2 cyc * count1 * count0
119         decfsz  count1,F        ; 1 cyc * count1
120         goto    outterloop      ; 2 cyc * count1
121         return                  ; 1 cyc
122         ; total = 3 + (6+3.count0).count1
123         ; count0 = 5 , count1 = 50, total = 1053 cyc
124
125 ; Time delay subroutine for 1.[W] seconds by calling DelayMS subroutine
126 DelayS:
127         movwf   temp1
128 delays_1:
129         movlw   .250
130         call    DelayMS
131         movlw   .250
132         call    DelayMS
133         movlw   .250
134         call    DelayMS
135         movlw   .250
136         call    DelayMS
```

```
137             decfsz  temp1,f
138             goto    delays_1
139             return
140
141     Init:
142             movlw   .7
143             banksel CMCON
144             movwf   CMCON       ; Disable analog comparator
145             banksel TRISB
146             movlw   0x00
147             movwf   TRISB       ; Set PORTB as output ports
148             movlw   0xFF
149             movwf   TRISA       ; Set PORTA as input ports
150             banksel PORTB
151             clrf    PORTB
152             clrf    temp
153             return
154
155             END
156
157
158
159
160
161
```