

Embedded Systems – Assignment 2

Problem Description:

Use MPLAB X IDE and the PIC18F45K50's assembly language to display d"00" through d"59" with a second between each number on two 7-segment LED displays. The source code must use TIMERO, an oscillator, and look-up tables.

Pseudocode:

```
ones_digit = 0; // Ones digit LED
tens_digit = 0; // Tens digit LED

while(true) {
    timer0 = 0;    // PIC1845K50's TIMERO
    counter1 = 0;
    counter2 = 0;

    // 256 * 256 * 61 ≈ 4,000,000
    while(counter2 != 61) {
        while(counter1 != 255) {
            while(timer0 != 255) {
                ++timer0;
            }

            ++counter1;
        }

        ++counter2;
    }

    ++ones_digit;

    if (ones_digit == 10) {
        ++tens_digit;
        ones_digit = 0;

        if (tens_digit == 7)
            tens_digit = 0;
    }
}

} // while(true) closing bracket
```

Assembly Code:

Code from "ES_A2.asm"; included in .zip file.

```
1  #include <p18F45K50.inc>
2
3  CONFIG WDTCN = OFF      ; Disable the watchdog timer.
4  CONFIG MCLRE = ON       ; MCLR pin is on.
5  CONFIG DEBUG = ON       ; Enable debug mode.
6  CONFIG LVP = ON         ; Low-voltage programming is on.
7  CONFIG PBDEN = OFF      ; RB[5:0] will be configured as digital inputs (datasheet, pg. 133)
8  CONFIG FOSC = INTOSCIO  ; Internal oscillator (port function on RA6)
9
10 ; Using 16 MHz oscillator, which means 4,000,000 instruction cycles / sec
11 ; I use three counters in total: TIMER0, COUNT1, and COUNT2.
12 ; Since 1 second == 4,000,000 instruction cycles and TIMER0 uses 1 instruction cycle to
13 ; increment itself, a counter with at minimum 22 bits is needed to count up to 4,000,000.
14
15 ; The 24-bit counter I'm using is as follows:
16 ; [23:16] == COUNT2, [15:8] == COUNT1, [7:0] == TIMER0
17 COUNT1 EQU 0x00
18 COUNT2 EQU 0x01
19
20 ONES_DIGIT EQU 0x02
21 TENS_DIGIT EQU 0x03
22
23 org 0      ; Start code at 0.
24
25 GOTO Start
26
27 Lookup_ONES:
28     ADDWF PCL, 1
29     RETLW b'10000001' ; 0
30     RETLW b'11001111' ; 1
31     RETLW b'10010010' ; 2
32     RETLW b'10000110' ; 3
```

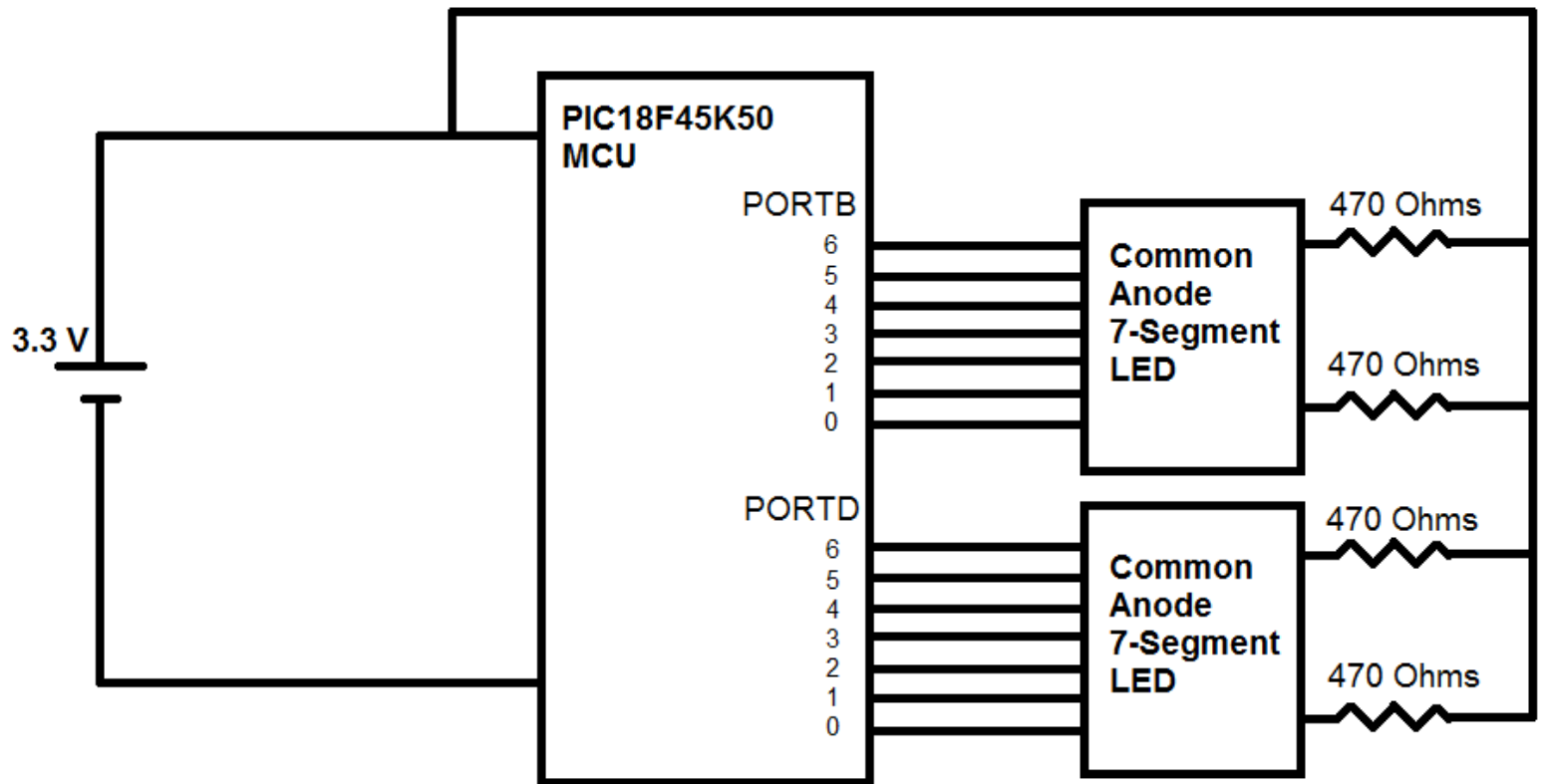
```
32      RETLW b'10000110'    ; 3
33      RETLW b'11001100'    ; 4
34      RETLW b'10100100'    ; 5
35      RETLW b'10100000'    ; 6
36      RETLW b'10001111'    ; 7
37      RETLW b'10000000'    ; 8
38      RETLW b'10000100'    ; 9
39
40      Lookup_TENS:
41          ADDWF PCL
42          RETLW b'10000001'    ; 0
43          RETLW b'11001111'    ; 1
44          RETLW b'10010010'    ; 2
45          RETLW b'10000110'    ; 3
46          RETLW b'11001100'    ; 4
47          RETLW b'10100100'    ; 5
48          RETLW b'10100000'    ; 6
49
50      Start:
51          MOVLB 0xF          ; Move low nibble to the bank select register.
52
53          CLRF PORTB         ; PORTB is the ones digit.
54          CLRF LATB
55          CLRF TRISB         ; Clear PORTB and use it for output.
56
57          CLRF PORTD         ; PORTD is the tens digit.
58          CLRF LATD
59          CLRF TRISD         ; Clear PORTD and use it for output.
60
61          MOVLW 0x08
62          MOVWF TOCON         ; TOCON = b"0000_1000"
63
```

```
63
64     MOVLW 0x78
65     MOVWF OSCCON      ; OSCCON = b"0111_1000"
66
67     CLRF COUNT1
68     CLRF COUNT2
69
70     CLRF ONES_DIGIT
71     CLRF TENS_DIGIT
72
73 Main:
74     CLRF TMROL
75     CLRF TMR0H          ; Set the timer/counter to 0; range of 0x00 through 0xFF.
76
77     BCF INTCON, TMR0IF
78     BSF TOCON, TMR0ON
79
80     MOVLW 0xC3
81     MOVWF COUNT2        ; COUNT2 starts at 0xC3 since only 61 increments are needed.
82
83     MOVLW 0x81
84     MOVWF PORTB
85     MOVWF PORTD
86
87 Loop:
88     BTFSS INTCON, TMR0IF
89     BRA Loop
90     INCF COUNT1, 1
91
92     ; Reset timer/counter overflow flag.
93     BCF TOCON, TMR0ON
94     CLRF TMROL
```

```
94      CLRF TMROL
95      CLRF TMROH
96      BCF INTCON, TMROIF
97      BSF TOCON, TMROON
98
99      BNZ Loop
100
101      BCF STATUS, Z
102      INCF COUNT2, 1
103
104      BNZ Loop
105
106      BCF STATUS, Z
107      INCF ONES_DIGIT, 1 ; Increment the ones digit for the lookup table.
108      INCF ONES_DIGIT, 1
109      BRA Set_COUNT2
110
111 Set_COUNT2:
112     MOVLW 0xC3
113     MOVWF COUNT2
114     MOVLW 0x14
115     CPFSEQ ONES_DIGIT
116     BRA Set_ONES
117     BRA Reset_ONES
118
119 Set_ONES:
120     MOVFF ONES_DIGIT, WREG
121     CALL Lookup_ONES
122     MOVWF PORTB
123     BRA Loop
124
125 Reset_ONES:
```

```
125 Reset_ONES:
126     CLRF ONES_DIGIT
127     MOVLW 0x00
128     CALL Lookup_ONES
129     MOVWF PORTB
130     INCF TENS_DIGIT, 1 ; Increment the tens digit for the lookup table.
131     INCF TENS_DIGIT, 1
132     MOVLW 0x0C
133     CPFSEQ TENS_DIGIT
134     BRA Set_TENS
135     BRA Reset_TENS
136
137 Set_TENS:
138     MOVFF TENS_DIGIT, WREG
139     CALL Lookup_TENS
140     MOVWF PORTD
141     BRA Loop
142
143 Reset_TENS:
144     CLRF TENS_DIGIT
145     MOVLW 0x00
146     CALL Lookup_TENS
147     MOVWF PORTD
148     BRA Loop
149
150 end
```

Wiring Diagram:



System's Picture:

