```
;*****************************************************************
;                                                                *
;    Filename:      n1np_frequency_counter_6-digit.asm             *
;    Date:          02 February 2021                               *
;    File Version:  1                                              *
;                                                                *
;    Author:        Benjamin Gayle                                 *
;                   www.AntonomasiaProductions.org                 *
;                                                                *
;                                                                *
;*****************************************************************
;                                                                *
;    Files required:                                             *
;                                                                *
;                                                                *
;                                                                *
;*****************************************************************
;                                                                *
;    Notes:       1. TMR0 input is 5V pulse.                     *
;                 2. Count for one second                        *
;                 3. Convert count to six individual BCD digits. *
;                 4. Output BCD to 7447 for common anode displays.*
;                 5. Multiplex the displays, each on for 1ms.    *
;                                                                *
;                                                                *
;*****************************************************************


    list    p=16F84A            ; list directive to define processor
    #include <p16f84a.inc>        ; processor specific variable definitions

    __CONFIG    _CP_OFF & _WDT_OFF & _PWRTE_ON & _XT_OSC

; '__CONFIG' directive is used to embed configuration data within .asm file.
; The lables following the directive are located in the respective .inc file.
; See respective data sheet for additional information on configuration word.



;***** VARIABLE DEFINITIONS
w_temp      EQU     H'0C'           ; variable used for context saving
status_temp EQU     H'0D'           ; variable used for context saving

count_low   EQU     H'0E'           ; lower 8 bits of count total
count_high  EQU     H'0F'           ; upper 8 bits of count total
count_carry EQU     H'10'           ; overflow for count > 65535

delay_1     EQU     H'12'           ; time delay
delay_2     EQU     H'13'

display_index   EQU H'14'           ; which display is active

loop        EQU     H'15'           ; display loop counter
count_bin   EQU     H'16'           ; index for Bin2BCD
count_dec   EQU     H'17'           ; index for Bin2BCD
```

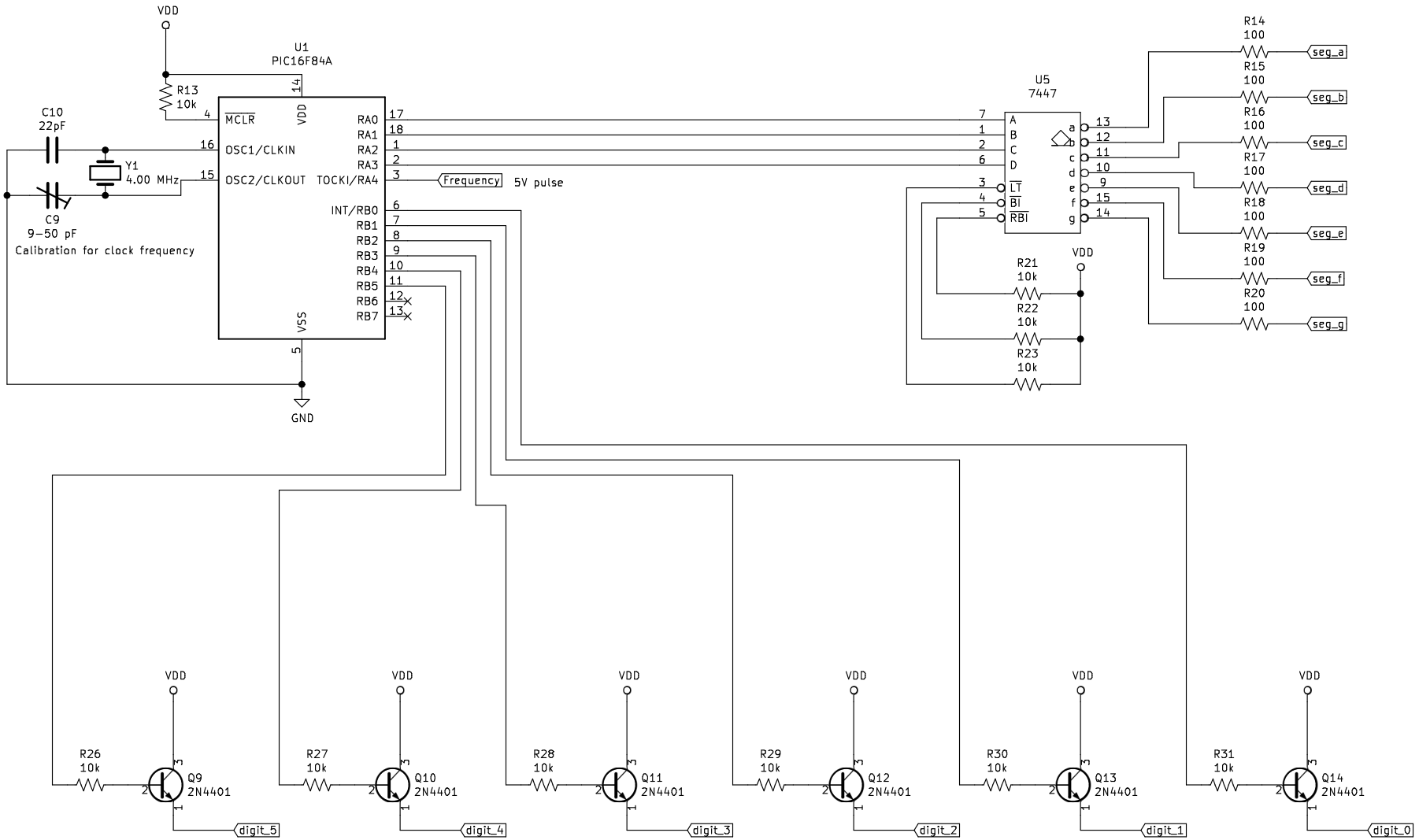
```
count_temp  EQU     H'18'

    ; variables for the display outputs
    ; will hold BCD to output on PORTA
digit_0     EQU     H'20'           ; ones
digit_1     EQU     H'21'           ; tens
digit_2     EQU     H'22'           ; hundreds
digit_3     EQU     H'23'           ; thousands
digit_4     EQU     H'24'           ; ten thousands
digit_5     EQU     H'25'           ; hundred thousands
digit_6     EQU     H'26'           ; placeholder for BCD conversion
digit_7     EQU     H'27'           ; placeholder for BCD conversion

;*****************************************************************
    ORG     H'0'                    ; processor reset vector

    CLRF    PORTA                   ; initialise PORTA
    CLRF    PORTB                   ; initialise PORTB
    BSF     STATUS, 5               ; set bank select to bank 1
    MOVLW   B'10000'                ; set RA0-RA3 as outputs
    MOVWF   TRISA
    MOVLW   B'00000000'             ; set RB0-RB7 as outputs
    MOVWF   TRISB

    CLRWDT                          ; Set prescaler for TMR0
    BSF     OPTION_REG, 5           ; OPTION_REG,5 is T0CS, 1 sets counter mode
    BSF     OPTION_REG, 4           ; OPTION_REG,4 is T0SE, 0 sets rising edge
    BSF     OPTION_REG, 3           ; OPTION_REG,3 is PSA, 0 assigns prescaler to TMR0
    BCF     OPTION_REG, 0           ; OPTION_REG,0 is PS0, prescaler value bit 0
    BCF     OPTION_REG, 1           ; OPTION_REG,1 is PS1, prescaler value bit 1
    BCF     OPTION_REG, 2           ; OPTION_REG,2 is PS2, prescaler value bit 2

    BCF     STATUS, 5               ; switch back to bank 0

    MOVLW   D'8'                    ; lamp test for first Display loop
    MOVWF   digit_5
    MOVWF   digit_4
    MOVWF   digit_3
    MOVWF   digit_2
    MOVWF   digit_1
    MOVWF   digit_0
    CLRF    count_low               ; initialise for safety
    CLRF    count_high
    CLRF    count_carry
    CLRF    TMR0

    GOTO    Start                   ; go to beginning of program

;*****************************************************************
; subroutines

Update_Count
    CLRF    count_temp
    MOVF    TMR0, 0                 ; get the current low byte of the count
    MOVWF   count_temp
    BTFSS   count_temp, 7           ; if zero, might have rolled over
    BTFSS   count_low, 7            ; if one, did roll over
```

```
    GOTO    C1
    INCFSZ  count_high, 1           ; record the roll-over in the high byte
    GOTO    C1                      ; if count_high is not zero
    BTFSS   count_carry, 0          ; if already set, do not increment more than once
    INCF    count_carry, 1          ; if count_high is zero, record the overflow
C1
    MOVF    count_temp, 0
    MOVWF   count_low               ; save the new low byte of the count

    RETLW   0
; END Update_Count

Delay_1ms
    MOVLW   D'195'                  ; time delay 1ms
    MOVWF   delay_1                 ; this sets how long each display will stay on
    NOP                             ; padding for loop timing
    NOP
    NOP
    NOP
Delay_5u                            ; 5 micro-second loop
    NOP
    NOP
    DECFSZ  delay_1, 1
    GOTO    Delay_5u
    RETLW   0
; END Delay_1ms

Display
    MOVLW   D'166'                  ; run for 1 second
    MOVWF   loop
Repeat
    MOVF    digit_5, 0
    MOVWF   PORTA                   ; output BCD
    MOVLW   B'100000'
    MOVWF   display_index
    MOVF    display_index, 0
    MOVWF   PORTB                   ; select which display to turn on
    CALL    Delay_1ms
    CLRF    PORTB                   ; turn all of the displays off
    RRF     display_index, 1        ; go to the next display
    CALL    Update_Count

    MOVF    digit_4, 0
    MOVWF   PORTA                   ; output BCD
    MOVF    display_index, 0
    MOVWF   PORTB                   ; select which display to turn on
    CALL    Delay_1ms
    CLRF    PORTB                   ; turn all of the displays off
    RRF     display_index, 1        ; go to the next display
    CALL    Update_Count

    MOVF    digit_3, 0
    MOVWF   PORTA                   ; output BCD
    MOVF    display_index, 0
    MOVWF   PORTB                   ; select which display to turn on
    CALL    Delay_1ms
    CLRF    PORTB                   ; turn all of the displays off
```

```
    RRF     display_index, 1        ; go to the next display
    CALL    Update_Count

    MOVF    digit_2, 0
    MOVWF   PORTA                   ; output BCD
    MOVF    display_index, 0
    MOVWF   PORTB                   ; select which display to turn on
    CALL    Delay_1ms
    CLRF    PORTB                   ; turn all of the displays off
    RRF     display_index, 1        ; go to the next display
    CALL    Update_Count

    MOVF    digit_1, 0
    MOVWF   PORTA                   ; output BCD
    MOVF    display_index, 0
    MOVWF   PORTB                   ; select which display to turn on
    CALL    Delay_1ms
    CLRF    PORTB                   ; turn all of the displays off
    RRF     display_index, 1        ; go to the next display
    CALL    Update_Count

    MOVF    digit_0, 0
    MOVWF   PORTA                   ; output BCD
    MOVF    display_index, 0
    MOVWF   PORTB                   ; select which display to turn on
    CALL    Delay_1ms
    CLRF    PORTB                   ; turn all of the displays off
    CALL    Update_Count

    NOP                             ; padding for loop timing
    DECFSZ  loop, 1
    GOTO    Repeat
    RETLW   0
; END Display

Bin2BCD
; 24-bit binary to BCD coversion
; from piclist Steve Teal
; ported from 18F to 16F by Benjamin Gayle, 21 December 2019

    CLRF    digit_7
    CLRF    digit_6
    CLRF    digit_5
    CLRF    digit_4                 ; clean up for the new values
    CLRF    digit_3
    CLRF    digit_2
    CLRF    digit_1
    CLRF    digit_0

    ; NB. BINn and FSR are trashed after this routine
    MOVLW   D'24'                   ; Initiate bit loop
    MOVWF   count_bin
BITLOOP
    RLF     count_low, 1            ; Every iteration of this loop will copy the next
    RLF     count_high, 1           ; bit of the bin value, starting with the MSB,
    RLF     count_carry, 1          ; to the carry flag
    MOVLW   digit_0
```

```
    MOVWF   FSR                     ; Initiate DECn pointer and counter
    MOVLW   D'8'
    MOVWF   count_dec               ; The following is executed 8 times per bit
DECLOOP
    RLF     INDF, 1                 ; Multiply DECn by two with carry, DECn * 2 + C
    MOVLW   D'10'
    SUBWF   INDF, 0
    BTFSC   STATUS, 0               ; DECn has overflowed (>>9) if carry is set
    MOVWF   INDF                    ; If carry is set DECn = DECn - 10
    INCF    FSR, 1                  ; Carry is CARRIED over to next multiply
    DECFSZ  count_dec, 1
    GOTO    DECLOOP                 ; Multiply next DECn
    DECFSZ  count_bin, 1
    GOTO    BITLOOP                 ; Do next bit
    RETLW   0

; END Bin2BCD

Blank                               ; turn off the leading displays if zero
    MOVLW   H'F'                    ; only need to check the lower nybble
    ANDWF   digit_5, 0              ; check if leading digit is zero
    BTFSS   STATUS, 2
    RETLW   0                       ; zero flag not set, we are done blanking
    MOVLW   H'F'                    ; zero flag set, blank the digit
    MOVWF   digit_5

    MOVLW   H'F'                    ; only need to check the lower nybble
    ANDWF   digit_4, 0              ; check if leading digit is zero
    BTFSS   STATUS, 2
    RETLW   0                       ; zero flag not set, we are done blanking
    MOVLW   H'F'                    ; zero flag set, blank the digit
    MOVWF   digit_4                 ; H'F' is D'15', turns off all segments

    MOVLW   H'F'                    ; only need to check the lower nybble
    ANDWF   digit_3, 0              ; check if leading digit is zero
    BTFSS   STATUS, 2
    RETLW   0                       ; zero flag not set, we are done blanking
    MOVLW   H'F'                    ; zero flag set, blank the digit
    MOVWF   digit_3                 ; H'F' is D'15', turns off all segments

    MOVLW   H'F'                    ; only need to check the lower nybble
    ANDWF   digit_2, 0              ; check if leading digit is zero
    BTFSS   STATUS, 2
    RETLW   0                       ; zero flag not set, we are done blanking
    MOVLW   H'F'                    ; zero flag set, blank the digit
    MOVWF   digit_2                 ; H'F' is D'15', turns off all segments

    MOVLW   H'F'                    ; only need to check the lower nybble
    ANDWF   digit_1, 0              ; check if leading digit is zero
    BTFSS   STATUS, 2
    RETLW   0                       ; zero flag not set, we are done blanking
    MOVLW   H'F'                    ; zero flag set, blank the digit
    MOVWF   digit_1                 ; H'F' is D'15', turns off all segments

    RETLW   0

; END Blank
```

```
Convert_Count
    CALL    Bin2BCD                 ; convert 3 bytes to 6 BCD digits
    CALL    Blank                   ; blank leading zeroes
    CLRF    count_low               ; reset for the next count
    CLRF    count_high
    CLRF    count_carry
    CLRF    TMR0                    ; re-start the count
    RETLW   0
; END Convert_Count

; END subroutines section

;*****************************************************************
Start
    CALL    Display                 ; timed loop, duration of the count
    CALL    Convert_Count
    GOTO    Start

    END                             ; directive 'end of program'
```



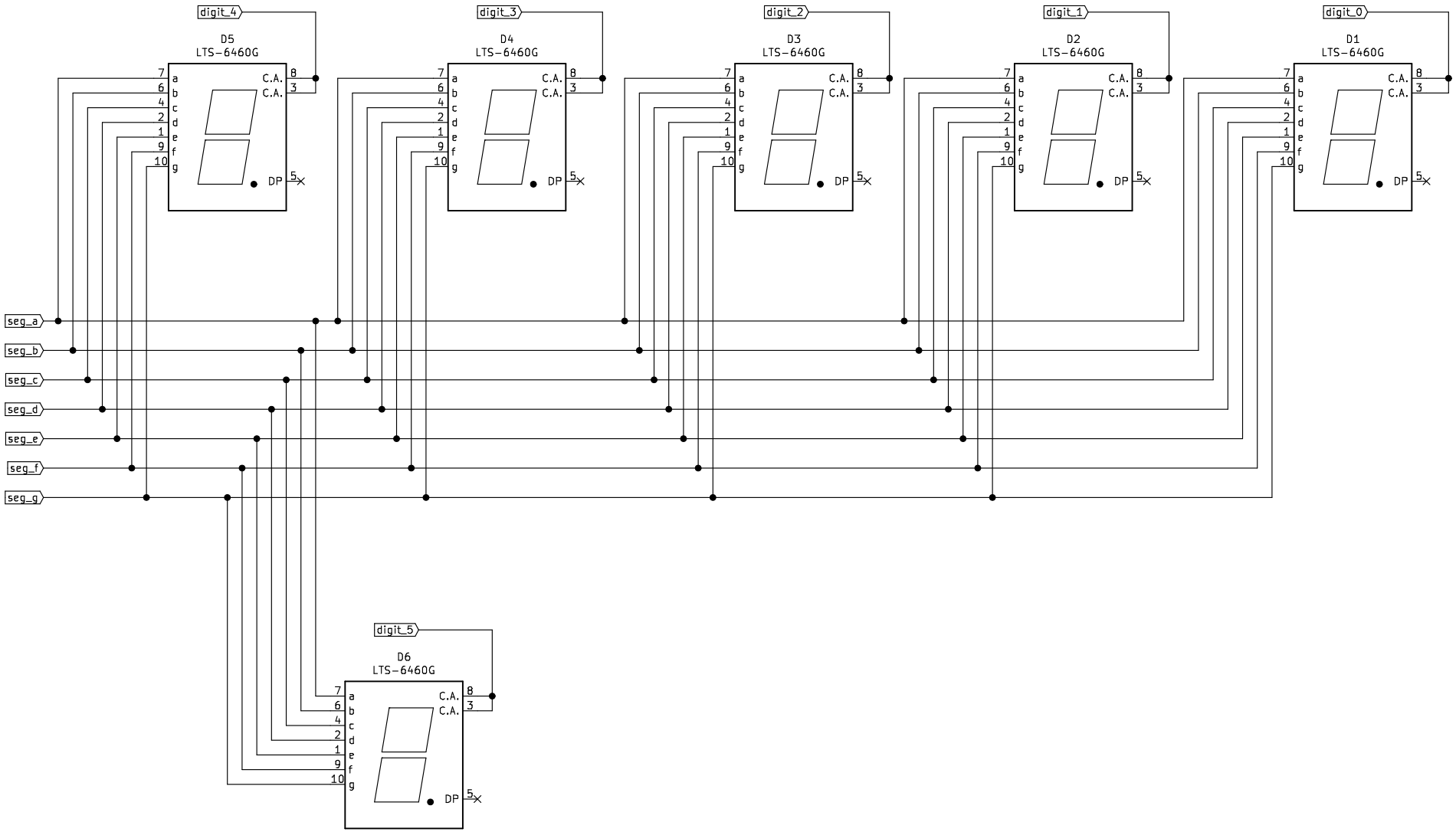
VDD = 5V

Sheet 1 of 2
Frequency Counter and 6–Digit Display Driver

Sheet: /
File: n1np_frequency_counter_6–digit_p1.sch

Title: N1NP Frequency Counter

Size: A	Date: 2021–03–07	**Rev: 1**
KiCad E.D.A. kicad 5.1.5+dfsg1–2build2		Id: 1/1



Sheet 2 of 2
7-Segment Displays

Sheet: /
File: n1np_frequency_counter_6-digit_p2.sch

Title: N1NP Frequency Counter

Size: A	Date: 2021-02-02	**Rev: 1**
KiCad E.D.A. eeschema 5.1.5+dfsg1-2build2		Id: 1/1