7 Segment Display & Multiplexing Code and Schematic

by

MagiDavid

On the following pages you'll find the code used in the 7 segment display video tutorials along with the schematic used to wire up the display. If you have any questions feel free to contact me via youtube or at www.neoloch.com

Notes:

ICSP

It's beyond the scope of this datasheet to cover all the ins and outs of ICSP, but I'll cover the few points that relate to the second video and usage of ICSP.

Since most often the microcontroller being programmed will be under circuit power (otherwise the PICkit will have to power the circuit), ICSP requires that proper isolation circuitry be employed to prevent the signals for the programming device being corrupted by signals for the circuit the programming device is plugged into. In this case a PICkit™ 2 is the programmer. During the first video implementing the PICkit 2 didn't seem practical since the LED display is connected directly to +5V. That in itself would probably cause interference and prevent ICSP form operating correctly.

In the second video the two transistors add a buffer between the display and V+ and we have the addition of a 1K ohm resistor between RB7 and digit 2's transistor provided enough isolation so that programming is possible. It's best, and I practice this when possible, to keep ICSP signals seperate from the circuit, but in this case it's not possible because of the limited number of I/O bits.

MCLR – VERY IMPORTANT

When dealing with microcontrollers that require an external pull-up resistor on MCLR, you MUST include an isolation diode to prevent programming voltage generated by the programmer from feeding back into the rest of the circuit. Programming a microcontroller without the diode in place can result in damage to the rest of the circuit.

Suggested reading:
http://ww1.microchip.com/downloads/en/DeviceDoc/30277d.pdf

Copyright and Distribution Notice

The contents of this file are being distributed under the:

You are free:

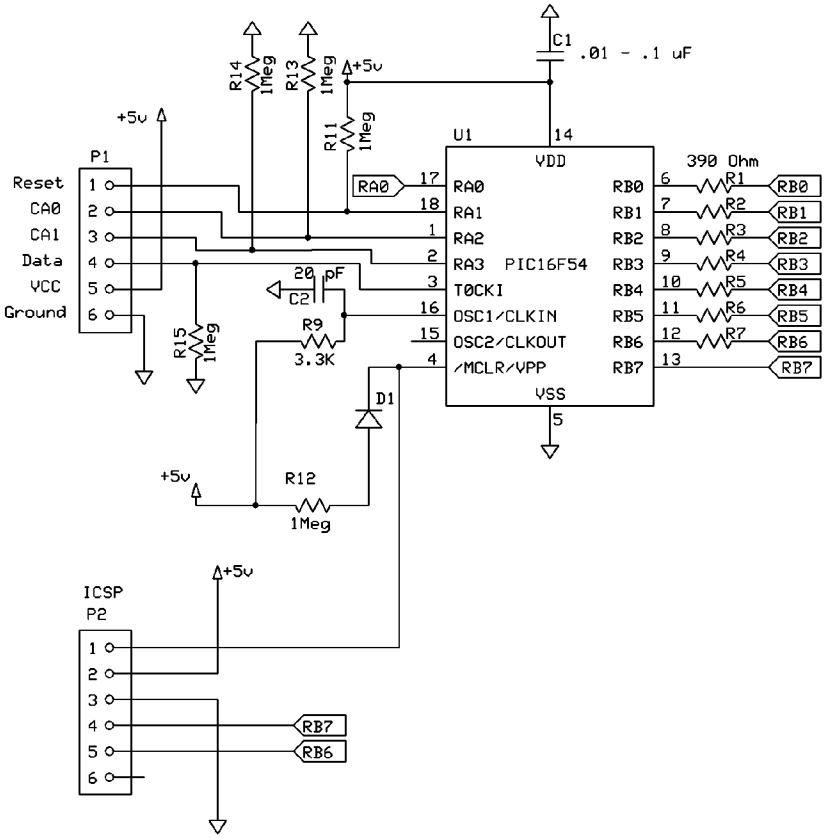
- **to Share** — to copy, distribute and transmit the work
- **to Remix** — to adapt the work

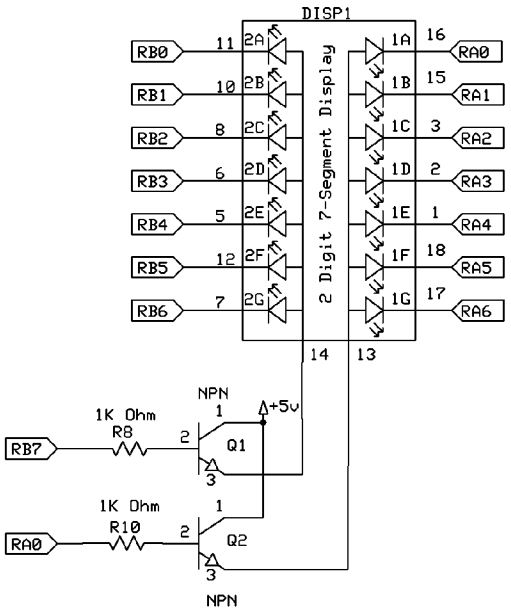
Under the following conditions:

- **Attribution** — You must attribute this work to NeoLoch, LLC (with link).

  What does "Attribute this work" mean?
  The page you came from contained embedded licensing metadata, including how the creator wishes to be attributed for re-use. You can use the HTML here to cite the work. Doing so will also include metadata on your page so that others can find the original work as well.
- **Noncommercial** — You may not use this work for commercial purposes.

If you have any questions regarding the licensing of this product please contact NeoLoch, LLC for clarification.





NeoLoch, LLC		
WP1 - 2 Digit 7 Seg Display		
D. Hoffman	Rev 1.0 9/13/2010	1 of 1

Assembly Code Listing

Below is the assembly code as it appears at the final stage of the second video. Feel free to edit this code for use in your own projects.

```
;=================================================================

    list      p=16F54           ; list directive to define processor
    #include <p16F5x.inc>       ; processor specific variable definitions

    __CONFIG   _CP_OFF & _WDT_OFF & _RC_OSC

; '__CONFIG' directive is used to embed configuration word 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
DISP_7SEG           UDATA
DISP_COUNTER     RES 1
DISP_TEMP           RES 1
DISP_LOOP           RES 1
DISP_FREQ           RES 1







;*****************************************************************
RESET_VECTOR   CODE   0x1FF       ; processor reset vector
        GOTO START

        ORG             0x000

START
        CLRF            DISP_COUNTER
        CLRF            DISP_TEMP
        CLRF            DISP_LOOP
        CLRW
        TRIS            PORTB
        TRIS            PORTA
```

```
MAIN
        CALL        DISP_UPDATE_D1
        CALL        DISP_PAUSE
        CALL        DISP_UPDATE_D2
        CALL        DISP_PAUSE
        INCFSZ      DISP_FREQ
        GOTO        MAIN
        INCF        DISP_COUNTER,F
        GOTO        MAIN


DISP_UPDATE_D1
        BCF         PORTB,7             ;TURN OFF DIGIT 2.
        SWAPF       DISP_COUNTER,W ;SWAP NIBBLES AND STORE IN W.
        MOVWF       DISP_TEMP
        MOVLW       0X0F
        ANDWF       DISP_TEMP,W         ;GET LOW NIBBLE FOR DISPLAY.
        CALL        SEVENSEG_LOOKUP
        MOVWF       PORTB               ;PUT DATA ON PORTB.
        BSF         PORTA,0             ;TURN ON DIGIT 1.
        RETURN

DISP_UPDATE_D2
        BCF         PORTA,0             ;TURN OFF DIGIT 1.
        MOVLW       0X0F
        ANDWF       DISP_COUNTER,W
        CALL S      EVENSEG_LOOKUP
        MOVWF       PORTB
        BSF    P    ORTB,7
        RETURN

DISP_PAUSE
        CLRF        DISP_TEMP
        CLRF        DISP_LOOP
DISP_PAUSE2
        INCFSZ      DISP_TEMP,F
        GOTO        DISP_PAUSE2
        RETURN


;-------------------------------------------------------------------------
; NUMBERIC LOOKUP TABLE FOR 7 SEG
;-------------------------------------------------------------------------
SEVENSEG_LOOKUP
        ADDWF       PCL,F
        RETLW       0X40
        RETLW       0X79
        RETLW       0X24
```

```
        RETLW   0X30
        RETLW   0X19
        RETLW   0X12
        RETLW   0X02
        RETLW   0X78
        RETLW   0X00
        RETLW   0X18
        RETLW   0X08
        RETLW   0X03
        RETLW   0X46
        RETLW   0X21
        RETLW   0X06
        RETLW   0X0E

; remaining code goes here

    END                 ; directive 'end of program'
```