NeoLoch FLS-4D7S-1010

Fireloch™ 4 Digit 7 Segment Programmable Display Module

Features:

- 3 to 11 wire operation.
- Breadboard compatible.
- Compact design.
- Count up / down.
- Count in Hex / Dec.
- Two character sets, display up to 30 different characters.
- 4 counter / display memories.
- Counter value can be preloaded.
- Scrolling display.
- ICSP for programming custom code.
- Maximum Count Frequency 26khz (approximate.)
- PIC16F57 Microcontroller controlled.

1.0 Device Overview

Designed to be a useful tool both on the breadboard and in finished projects, the Fireloch™ FLS-4D7S-1010 display is a quick and easy tool to add a display to your projects. Even more, since the code is open to the community you can customize how the display functions to your specific needs.

To ease integration into breadboard circuits, all inputs are tied low with 1Mohm pull down resistors to keep inputs stable when not in use. This also has the benefit of only requiring three wires to enable the count up mode (Power, Ground, and Count.) Default configuration of the display is count up in hexadecimal.

The counter module has four built in memories that can be accessed using the pins DA1 and DA0 (default configuration = counter 0, DA1 = 0 & DA0 = 0.) This feature allows a count / display to be loaded into the first counter and stored for later use or recall. For example: If counter 0 counts up to FF and then the active counter is switched to counter 1 (DA1 = 0 & DA0 = 1), the value contained in counter 0 will not change since it is not the active counter, nor will counters 2 and 3. Only the active counter's displayed memory will be altered by input signals. This functionality allows for the storage and later retrieval of a counter value.

The display module has four main operational modes:

Mode1	Mode0	Mode of operation
0	0	Count up (default)
0	1	Count Down
1	0	Shift display left one digit and load LS digit with the value contained on D4-D0 using character set 1.
1	1	Shift display left one digit and load LS digit with the value contained on D4-D0 using character set 2.

Module Pin Descriptions

Pin Number	Pin Name	Pin Type	Description
1	D0 Clr	I	In add / sub mode this pin will clear the display. In display load mode: D0 of character to load.
2	D1 Hex/Dec	I	In add / sub mode: 0 = count hex, 1 = count decimal. In display load mode: D1 of character to load.
3	D2	I	D2 of character to load.
4	D3	I	D3 of character to load.
5	DA0	I	Address 0 of counter to display (default is 0.)
6	DA1	I	Address 1 of counter to display (default is 0.)
7	Mode0	I	Low bit of counter display mode.
8	Mode1	I	High bit of counter display mode.
9	Count / Load	I	Trigger the display to count up / down or load the character on D3-D0.
10	NC	-	Not Connected.
11	+5V	-	Positive voltage supply - +5V
12	Ground	-	Ground.

1.1 **Counting up in Hexadecimal / Decimal (see section 1.3 for decimal.)**

Using the counter to count up in Hexadecimal requires only three connections. +5V, Ground, and Count. Due to the pull down resistors, all other pins are pre-configured to the proper settings for counting up in Hex. The counter's maximum count is FFFF in this mode.

1.2 **Count Down in Hexadecimal / Decimal (see section 1.4 for decimal.)**

Counting down requires bringing Mode0 high. This will put the counter into count down mode.

1.3 **Count up in Decimal.**

To count up/down in decimal, D1 must be brought high (D1 = 1.)

1.4 **Count Down in Decimal.**

To count down in decimal, D1 and Mode0 must be brought high (D1=1 & Mode0=1)

1.5 **Loading Character Set 1.**

Character set 1 contains the hexadecimal character codes, 1-F. This is the character set the display uses to display digits when in counting mode. To load a character value directly into the display: first place the counter into character load mode by bringing Mode1 high and Mode 0 low.

Next, place the value of the character on the data lines D3 – D0. The value placed on the data lines directly corresponds to the value the display will load. For example: placing 0010 on the data lines will load the value 2 into the display, placing 1111 will load the value F into the display.

The last step is to pulse Load high to load the character. This will shift the contents of the display to the left and load the value located on the data lines into the LS 7 segment display (digit on the right.) Repeat this process to load the desired display value.

1.6 **Loading Character Set 2.**

A second character set has been incorporated into the display to add versatility and fun. Below is a list of the characters contained in the second character set:

Hexadecimal	D3-D0 Value	Character
0	0000	Space
1	0001	H
2	0010	L
3	0011	N
4	0100	1 – Segments E&F
5	0101	P
6	0110	U
7	0111	]
8	1000	-
9	1001	= – Segments G & C
A	1010	Square upper seg.
B	1011	Square lower seg.
C	1100	Funky 'R'
D	1101	Alien Char 1
E	1110	Alien Char 2
F	1111	Alien Char 3

By bringing both Mode1 and Mode0 high the counter will be placed into the load character using the second character set. Internally, the display module keeps track of each digits character set requirement. This allows characters from both character sets to be displayed at the same time.

A typical message display process would be to load 4 spaces to clear the display. Then load in the appropriate values from both character sets to display the desired message. For words that are longer than four letters or phrases a scrolling technique can be employed by simply pausing in between character loads. To see an example of this, place the counter in add mode in hex and count up to 9. Then bring D2 high, this will activate the "NEOLOCH" easter egg which scrolls "NEOLOCH" across the display. As long as D2 remains high, the egg will continue to operate.

1.7 **Creating Custom Characters**

First, a little background on how a digit is actually displayed is in order. The Fireloch display unit is composed of a common anode 7 segment multiplexed display. Each digit on the display has a corresponding common anode pin. Each of the transistors is connected to one of the anodes to control when that particular digit is on or off. PORTA is used to control the transistors.

Bit #	7	6	5	4	3	2	1	0
Segment	DOT	G	F	E	D	C	B	A

To update the display, the module will first turn off the previous digit's transistor, load the data for the next digit to be displayed, put that data on PORTC, and then turn on the next transistor. Using this method there isn't any ghosting carried over from the previous digit since all digits are off before PORTC is updated.

Now, since the display is common anode, to actually turn on a segment a 0 has to be placed on the corresponding I/O pin on PORTC.

Each pin on PORTC corresponds to a segment on the LED display, they are:
To display 0 on a digit, the byte would equal "11000000" = C0. Only segment DOT and G are off, so bits 6 & 7 would be set to 1 while the other bits are all 0. To display a 2, segments A,B,D,E, & G need to be on and that would be a value of "10100100" = A4.

So, to create a custom character you first need to determine which segments you need on and then calculate the correct hex value. Once you have that, the code in SEVENSEG_LOOKUP or SEVENSEG_LOOKUP2 table located in FLS-4D7S-1010 – Bank 0.asm needs to be changed. Then recompile the code and program the display using the ICSP port.

1.8 **PORTA**

PORTA is used to control the transistors that handle turning the individual digits on or off. Bringing an I/O pin high will turn on the corresponding transistor and bringing the I/O pin low will turn of the transistor. The display digits are numbered 1 – 4 left to right.

PORTA Bit	0	1	2	3
Display Digit #	1	2	3	4

2.0 In Circuit Serial Programming (ICSP™)

Special considerations have to be observed when using ICSP to program the PIC™ 16F57. The circuits design gives priority to ICSPDAT (RB7) and ICSPCLK (RB6), this means that when a programmer (PICkit™ 2 or PICkit™ 3) is plugged into the ICSP port, signals from the PICKit will override signals from pins 7 & 8 on the header connector.

This interference means that signals to RB6 & RB7 will not be read correctly and the display will not operate correctly until the PICkit is removed from the ICSP header and the display is reset. If you are writing your own firmware to use in the Fireloch display, it is highly recommended to avoid using RB6 and RB7 if at all possible. Doing so will allow the PICkit to remain connected to the Fireloch display module while degugging the custom firmware.

2.1 **Programming with PICkit 2**

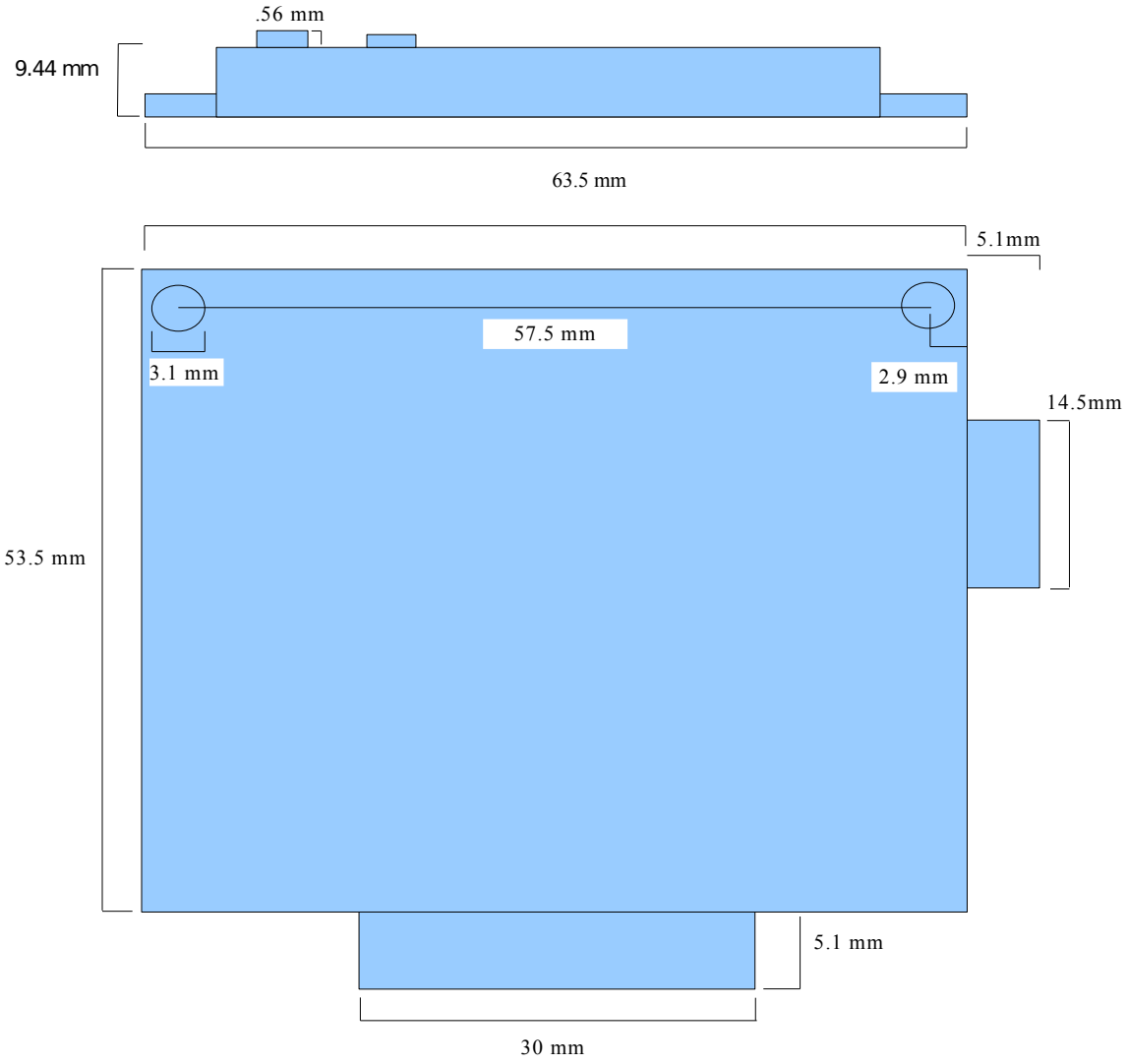
Since MPLAB™ doesn't support programming the PIC16F57 using a PICkit 2, the utility that comes with PICkit 2 must be used to program the Fireloch™ display .

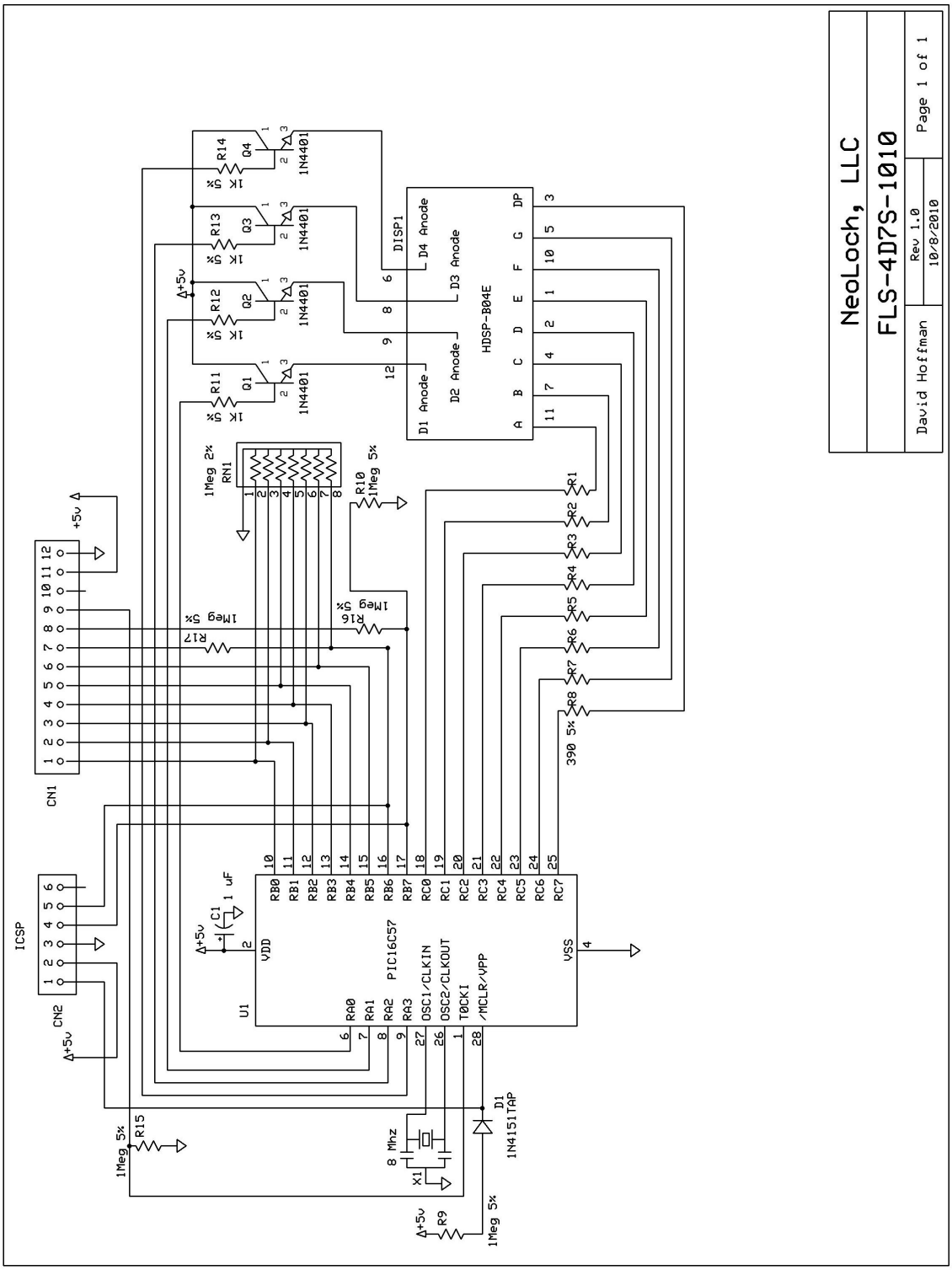
2.2 **Programming with PICkit 3**

MPLAB supports programming the 16F57 with the PICkit 3. So the Fireloch™ display can be programmed directly from MPLAB.

For more information about programming the PIC16F57 please reference Microchip's PIC16F5x datasheet.

3.0 **Dimensions**




NeoLoch, LLC
FLS-4D7S-1010
David Hoffman
Rev 1.0
10/8/2010
Page 1 of 1
HDSP-B04E
D1 Anode
D2 Anode
D3 Anode
D4 Anode
DISP1
1N4401
1K 5%
R11
R12
R13
R14
Q1
Q2
Q3
Q4
1Meg 2%
RN1
1Meg 5%
R10
R16
R17
390 5%
R1
R2
R3
R4
R5
R6
R7
R8
CN1
CN2
ICSP
+5v
C1
1 uF
U1
PIC16C57
VDD
VSS
RB0
RB1
RB2
RB3
RB4
RB5
RB6
RB7
RC0
RC1
RC2
RC3
RC4
RC5
RC6
RC7
RA0
RA1
RA2
RA3
OSC1/CLKIN
OSC2/CLKOUT
T0CKI
/MCLR/VPP
8 Mhz
X1
R15
R9
D1
1N4151TAP