

NeoLoch

NLT-28P-LCD-5S Tester Datasheet

Overview

The LCD IC Tester is designed to test 9114, 5101, 6116, 6264, 6810, 7489, M5M55189BP and W24257 RAM ICs. The tester provides ongoing information about the RAM IC under test and further information on memory devices that fail the test.

The tester also includes a number of new features, including:

- New easier to use layout.
- 2.1mm DC power jack for connecting to external power sources.
- 2 line by 16 character LCD screen.
- Faster PIC MCU operating at 16 MHz.
- Power switch.
- Relay controlled power connections for most 28 pin RAM ICs, allowing for the testing of high current ICs.

This document details the operation of the default configuration of the IC tester as well as details on the device's operation for custom code design.

Information contained in this document is provided for your convenience only and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. NeoLoch makes no representations or warranties of any kind, expressed or implied, related to the information contained in this document. NeoLoch disclaims all liability arising from the information contained in this document and its use. Use of NeoLoch devices in life support and/or safety applications is entirely at the buyer's risk. And, the buyer agrees to indemnify and hold harmless NeoLoch from any and all damages, claims, suits, or expenses resulting from such use.

Trademarks

The NeoLoch name, the NeoLoch logo, and Fireloch are trademarks of NeoLoch, LLC in the U.S.A.

Table of Contents

1.0 Device Connection.....	3
1.1 CN1 – ICSP.....	3
1.2 J1 & JP1 – Power Connector.....	3
1.3 CN2 – +V Solder Power Connector	3
1.4 ZIF 1 – Zero Insertion Force Socket.....	3
1.5 S7 – AB Switch.....	3
2.0 Understanding RAM Tester Passes.....	4
2.1 Passes 1 through 16 – Memory Check.....	4
2.1 Pass 17 – Address Check.....	4
2.1 Display Bad RAM Results.....	4
3.1 Testing 9114 RAM ICs.....	5
3.1.1 2114 & 6514 RAM.....	5
3.2 Testing 2101 / 5101 RAM ICs.....	5
3.3 Testing 4016 / 6116 RAM ICs.....	5
3.4 Testing 2465 / 6264 RAM ICs.....	6
3.5 Testing 6810 RAM ICs.....	6
3.6 Testing 7489 RAM ICs (Requires 16 Pin Adapter Board).....	6
3.7 Testing M5M55189BP RAM ICs.....	6
3.8 Testing 62256 / W24257 RAM ICs.....	7
4.0 Schematic.....	8
5.0 Ports Configuration.....	9
6.0 Parts List.....	10
7.0 Code Examples.....	11
6.1 LCD Control Code.....	11
7.2 Reading The Switch Array.....	14
6.3 Switch Array Example Code.....	15
Appendix A: Revision History.....	16

1.0 Device Connection

1.1 CN1 – ICSP

ICSP connector, this port is designed to attach to a PICKit 2, PICKit 3 or compatible programmer.

1.2 J1 & JP1 – Power Connector

J1 is a 2.1mm barrel style DC power jack, allowing either a 5V or 12V compatible power supply to be easily connected to the tester. **IMPORTANT:** See note regarding Jumper 1 below.

JP1 allows the tester to either run on a regulated 5V supply or a regulated / unregulated power source in the range of 7V to 12V. By default, assembled unit have this jumper set to the higher voltage position. If you are going to use the +5V setting, make sure your power source can supply a regulated +5V at 200ma to 300ma.

IMPORTANT: If you are assembling your own unit, please pay careful attention to JP1's setting, having this set to +5V and supplying the unit with a voltage greater than 5.5V can and most probably will damage or destroy the tester.

1.3 CN2 – +V Solder Power Connector

Included for convenience, this connector operates exactly as J1 and is included for connecting to other external power sources. This connection can also be used to route power to external devices from the DC power jack.

1.4 ZIF 1 – Zero Insertion Force Socket

The ZIF socket provides an easy way to insert and remove ICs under test. Pin 1 of the ZIF socket is the pin closest to the lever of the ZIF socket.

1.5 S7 – AB Switch

This switch is used to configure the ZIF socket for testing 6810 or all other RAM. In position A the ZIF is configured for 2114/9114, 5101, 6116, 6264, M5M55189BP and W24257. And in position B the ZIF is configured for 6810 RAM ICs. Please make note of this switches position when testing RAM ICs if you've tested a 6810 in the past.

2.0 Understanding RAM Tester Passes

The tester was designed to be quickly and easily operated. RAM ICs can be inserted into the device while it is powered. When not testing, the ZIF socket is not powered and all pins are either at ground potential or in a high-impedance state.

2.1 Passes 1 through 16 – Memory Check

These passes consist of either writing 0X00 through 0X0F for 4 bit memories, or 0X00 through 0XFF for 8 bit memories.

Pass 1 consists of writing 0X00 to each memory location and then reading the data back and with each successive pass 0X01 or 0X11 is added to the test value. This quickly covers all possible combinations and will bring out any flaws.

At the end of each pass a summary will be displayed on the LCD screen for higher memory devices. Lower memory devices test in just a few seconds, so no information is displayed for them at this point. However, this information can be displayed at the end of a test for those RAM ICs that fail.

2.1 Pass 17 – Address Check

It's possible that a RAM IC's memory is just fine while the addressing block has a failure, permitting, for example, two different addresses to access the same memory location. Pass 17 writes a different offset value to each memory location to detect if there's an address issue.

For 4-bit devices this consists of writing 0X00 to address 0X00, 0X01 to address 0X01, and so on. Once the program reaches address 0X10 however, the offset value kicks in and we write 0X01 to address 0X10. This offset value slowly increases with every rollover of the nibble that's being written to the RAM IC.

This offset will cause a bad address to come out on the read back, and reveal an otherwise hard to detect issue.

For 8 bit devices, the process is the same except the offset doesn't kick in until a rollover has happened from the full 8 data bits.

2.1 Display Bad RAM Results.

If a RAM IC tests bad, the collected data on each pass can be displayed by pressing the up and down arrow keys on the right side of the LCD display. The first line consists of the pass number, the first address where a bad read was detected, and if that pass failure or a success. An example of what line 1 appears like: "Pxx S:xxx FAILED or PASSED"

Line 2 consists of the number of bad reads from that pass and a accumulation of the bad bits found in the pass. Good or not relevant bits are displayed with a “-”, while bad bits are displayed with a “X”. An example of this line: “L:FF B:---X--X”3.0

Operating The Tester
When first powered on, the tester will configure itself. When ready, The LCD screen will display “Select Test” on line 1 and the currently selected test on the second line. The tester will remember the last type of RAM IC tested and will power up on that test for convenience.

3.1 Testing 9114 RAM ICs.

Step 1: Use the left or right arrow keys to select 9114 as the test.

Step 2: Make sure the AB switch is in the A position.

Step 3. Insert the RAM IC into the ZIF socket, lower the lever, and then press the “TEST” key.

Testing a 9114 RAM IC takes several seconds, as each pass is completed a result of that pass will be displayed on the LCD screen.

3.1.1 2114 & 6514 RAM

2114 & 6514 RAM is no longer officially supported in this tester, Older version of the tester will display 2114 & 6514 as an option, but testing these RAM ICs is not recommended as the test has proven unreliable with some variants.

3.2 Testing 2101 / 5101 RAM ICs.

Step 1: Use the left or right arrow keys to select 2101/5101 as the test.

Step 2: Make sure the AB switch is in the A position.

Step 3. Insert the 5101 RAM IC into the ZIF socket, lower the lever, and then press the “TEST” key.

Testing a 5101 RAM IC takes only a few second, due to the speed of this test no data will be displayed during the test.

3.3 Testing 4016 / 6116 RAM ICs.

Step 1: Use the left or right arrow keys to select 4016/6116 as the test.

Step 2: Make sure the AB switch is in the A position.

Step 3. Insert the RAM IC into the ZIF socket, lower the lever, and then press the “TEST” key.

Testing a 6116 RAM IC takes several seconds, as each pass is completed a result of that pass will be displayed on the LCD screen.

3.4 Testing 2465 / 6264 RAM ICs.

Step 1: Use the left or right arrow keys to select 2465/6264 as the test.

Step 2: Make sure the AB switch is in the A position.

Step 3. Insert the RAM IC into the ZIF socket, lower the lever, and then press the “TEST” key.

Testing a 6264 RAM IC takes about 15 seconds, as each pass is completed a result of that pass will be displayed on the LCD screen.

3.5 Testing 6810 RAM ICs.

Step 1: Use the left or right arrow keys to select 6810 as the test.

Step 2: Make sure the AB switch is in the B position.

Step 3. Insert the RAM IC into the ZIF socket, lower the lever, and then press the “TEST” key.

Testing a 6810 RAM IC takes about only a few seconds. due to the speed of this test no data will be displayed during the test.

3.6 Testing 7489 RAM ICs (Requires 16 Pin Adapter Board).

Step 1: Insert the 7489 adapter into the ZIF socket and lock it into place.

Step 1: Use the left or right arrow keys to select 7489 as the test.

Step 2: Make sure the AB switch is in the A position.

Step 3. Insert the RAM IC into the ZIF socket, lower the lever, and then press the “TEST” key.

Testing a 7489 RAM IC takes about only a few seconds. due to the speed of this test no data will be displayed during the test.

3.7 Testing M5M55189BP RAM ICs.

Step 1: Use the left or right arrow keys to select M5M55189BP as the test.

Step 2: Make sure the AB switch is in the A position.

Step 3. Insert the RAM IC into the ZIF socket, lower the lever, and then press the “TEST” key.

Testing a 6264 RAM IC takes about 20 seconds, as each pass is completed a result of that pass will be displayed on the LCD screen.

3.8 Testing 62256 / W24257 RAM ICs.

Step 1: Use the left or right arrow keys to select 62256/W24257 as the test.

Step 2: Make sure the AB switch is in the A position.

Step 3. Insert the RAM IC into the ZIF socket, lower the lever, and then press the “TEST” key.

Testing a W24257 RAM IC takes about 2 minutes and 45 seconds, as each pass is completed a result of that pass will be displayed on the LCD screen.

5.0 Ports Configuration

MCU Pin Number	Description	Assigned to
1	MCLR	ICSP
2	RA0	ZIF Pin 1
3	RA1	ZIF Pin 2
4	RA2	ZIF Pin 3
5	RA3	ZIF Pin 4
6	RA4	ZIF Pin 5
7	RA5	ZIF Pin 6
8	RE0	ZIF Pin 7
9	RE1	ZIF Pin 8
10	RE2	ZIF Pin 9
11	VDD	GND
12	VSS	+5V
13	RA7	ZIF Pin 10
14	RA6	ZIF Pin 13
15	RC0	ZIF Pin 12
16	RC1	ZIF Pin 11
17	RC2	LCD CD
18	RC3	LCD R/W
19	RD0	VDD Relay (ZIF pin 28)
20	RD1	VSS Relay (ZIF pin 14)
21	RD2	ZIF Pin 19
22	RD3	Serial Register Clock
23	RC4	Shift Register Data
24	RC5	ZIF Pin 21
25	RC6	ZIF Pin 20
26	RC7	ZIF Pin 19
27	RD4	ZIF Pin 18
28	RD5	ZIF Pin 17
29	RD6	ZIF Pin 16
30	RD7	ZIF Pin 15
31	VSS	GND
32	VDD	+5V
33	RB0	ZIF Pin 22
34	RB1	ZIF Pin 23
35	RB2	ZIF Pin 24
36	RB3	ZIF Pin 25
37	RB4	ZIF Pin 26
38	RB5	PCB Switch
39	RB6	ICSP / ZIF Pin 27
40	RB7	ICSP

6.0 Parts List

- 1 – PC board.
- 1 – 40 pin DIP socket.
- 1 – 14 pin DIP socket.
- 1 – 28 pin ZIF socket.
- 1 – 7805CT 1A +5V regulator.
- 1 – LCD display.
- 2 – Reed relays.
- 1 – 3 pin straight header.
- 1 – 6 pin straight header.
- 1 – 16 pin straight header.
- 1 – Jumper.
- 1 – DC power jack.
- 1 – 0.33 uF capacitor.
- 3 – 0.1 uF capacitors.
- 1 – 1.0 uF capacitor.
- 1 – 1K ohm resistor.
- 3 – 10K ohm resistor.
- 1 – 3,3K ohm resistor.
- 1 – 15K ohm resistor.
- 1 – 20K ohm resistor.
- 1 – 1M ohm resistor.
- 1 – 4.7K potentiometer.
- 5 – PCB mount switches.
- 1 – PIC16F1519 microcontroller (programmed)
- 1 – 74HC174N shift register.
- 2 – PC board mount switches.
- 1 – 1/4" screw and nut.
- 4 – Rubber bumpers (for use as feet for the PCB, these are optional).
- 1 – Shunt.

7.0 Code Examples

The following pages consist of code examples written in C for controlling the LCD and reading the switches for those of you interested in developing your own applications.

6.1 LCD Control Code

Once the LCD is configured and started using `lcd_port_setup()` and `lcd_config()`, writing data to the LCD screen can be accomplished using `printf()`.

Basic LCD control routines are included that will:

`lcd_home()` = return cursor to home position, line 1, column 1 and leave current data on screen.

`lcd_clear()` = clear screen and return cursor to home position.

`lcd_line2()` = moves the cursor to the first column on line 2, does not erase any displayed data.

```
#include <xc.h>

#define LCD_CD      LATC2
#define LCD_RW      LATC3
#define LCD_SDATA   LATC4
#define LCD_CLK     LATD2
#define LCD_SCLK    LATD3

//-----
// LCD delay routines.
//-----
void lcd_delay_5ms()
{
    __delay_ms(2);
}
void lcd_delay_1ms()
{
    __delay_ms(10);
}
//-----
// LCD port configuration.
//-----
void lcd_port_setup()
{
    TRISD3 = 0;
    TRISC4 = 0;
    TRISD2 = 0;
    TRISC2 = 0;
    TRISC3 = 0;
}
```

```
//-----  
// LCD Clock data in.  
//-----  
void lcd_clock()  
{  
    LCD_CLK = 1;  
    NOP();  
    NOP();  
    NOP();  
    NOP();  
    LCD_CLK = 0;  
}  
//-----  
// putch routine used with printf function.  
//-----  
void putch(char cout)  
{  
    for(int i=0;i<8;i++)  
    {  
        if((cout & (1 << i)))  
            LCD_SDATA = 1;  
        else  
            LCD_SDATA = 0;  
        LCD_SCLK = 1;  
        NOP();  
        LCD_SCLK = 0;  
    }  
    LCD_CD = 1;  
    LCD_RW = 0;  
    lcd_clock();  
}  
//-----  
// Send LCD command to LCD.  
//-----  
void lcd_command(char cout)  
{  
    int i;  
    for(i=0;i<8;i++)  
    {  
        if((cout & (1 << i)))  
            LCD_SDATA = 1;  
        else  
            LCD_SDATA = 0;  
        LCD_SCLK = 1;  
        NOP();  
        LCD_SCLK = 0;  
    }  
    LCD_CD = 0;  
    LCD_RW = 0;  
    lcd_clock();  
    lcd_delay_1ms();  
}
```

```
//-----  
// LCD control functions.  
//-----  
void lcd_clear()  
{  
    lcd_command(0x01);  
    lcd_delay_1ms();  
}  
void lcd_line2()  
{  
    lcd_command(0xC0);  
    lcd_delay_1ms();  
}  
void lcd_home()  
{  
    lcd_command(0x02);  
    lcd_delay_1ms();  
}  
//-----  
// LCD display wake and config.  
//-----  
void lcd_config()  
{  
    lcd_port_setup();  
    lcd_delay_5ms();  
    lcd_delay_5ms();  
    lcd_delay_5ms();  
    lcd_command(0x30); // Wake up.  
    lcd_delay_5ms();  
    lcd_command(0x30); // Wake up.  
    lcd_delay_1ms();  
    lcd_command(0x30); // Wake up.  
    lcd_delay_1ms();  
    lcd_command(0x38); // 8 bit, 2 lines.  
    lcd_command(0x10); // Set cursor at home position.  
    lcd_command(0x0c); // Display on, Blink cursor.  
    lcd_command(0x06); // Entry mode set.  
    lcd_command(0x01); // Clear display and set address to 00H.  
}
```

7.2 Reading The Switch Array.

The switch array consists of 5 switches and 6 resistors. When each switch is pressed it creates a different voltage on the analog input. By reading this voltage and converting it to a digital value, it becomes possible to know which switch has been pressed.

Since resistance values of resistors can vary not only from the manufacturing process, but also environmental conditions, such as temperature, it's important to normalize the voltage reading as much as possible so that switch presses can be reliably detected. This is accomplished by dropping most of the digital conversion and only using the 5 most significant bits from the conversion.

The switch ADC code left justifies the conversion data and then only using ADRESH, thereby dropping the 2 least significant bits. However, we still need to drop some more bits to get a stable and reliable number to work with. This is accomplished by filtering out the least significant bits of ADRESH with an AND statement. Now we are just left with the four most significant bits of the ADC value. This filtered ADC value is then compared to what the reading should be for each switch voltage.

Switch	ADC Value After Filtering
None	00h
Up	C0h
Right	70h or 80h
Down	60h
Left	50h
Enter	E0h

To do an ADC scan of the switch array, PORTB, bit 0 must be configured as an analog input. Then KEY_PRESS_SCAN is called, the rest of the required code is self contained in the following routines.

6.3 Switch Array Example Code.

```
// include these lines in main configuration routine.
PORTB = 0;
TRISB = 0b00100000;
ANSELB = 0b00100000;

//-----
// Read ADC for key presses.
//-----
char adc_get_data()
{
    ADCON1 = 0;
    ADCON0 = 0b10110101;
    NOP(); // Give ADC voltage time to settle.
    NOP();
    GO_nDONE = 1;
    while (GO_nDONE == 1)
        NOP();
    return(ADRESH);
}

// part of main routine.

while (1 == 1)
{
    adc_get_data();
    key_press = (ADRESH);
    __delay_ms(5); // Debounce key presses.
    if(key_press < 0x10)
        key_pressed = 0;
    if(key_press >0xE0 && key_press <0xE9)
        key_enter();
    if(key_press >0xB8 && key_press <0xC2)
        key_up();
    if(key_press >0x78 && key_press <0x82)
        key_right();
    if(key_press >0x5E && key_press <0x68)
        key_down();
    if(key_press >0x4D && key_press <0x57)
        key_left();
}
```

Appendix A: Revision History

Revision A (12/2013)

- Initial release of this document

Revision B (8/2014)

- Updated test names to include those actually shown on the tester.
- Corrected parts list to include the 1M ohm resistor placed in R9.
- Added text to clarify the use of the included bumpers in the kit.

Revision C (10/2014)

- Updated section 1.4 to reflect the correct insertion method for RAM under test. This section contained the instructions for insertion RAM into the ZIF socket from the older 5101 tester, which were incorrect for the LCD tester.
- Corrected typographical errors.

Revision D (3/2015)

- Removed 2114 RAM from the support list.
- Removed 6514 RAM from the support list.