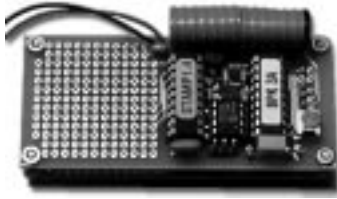Assembling and Using the

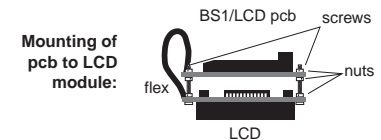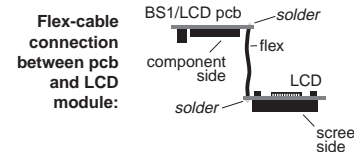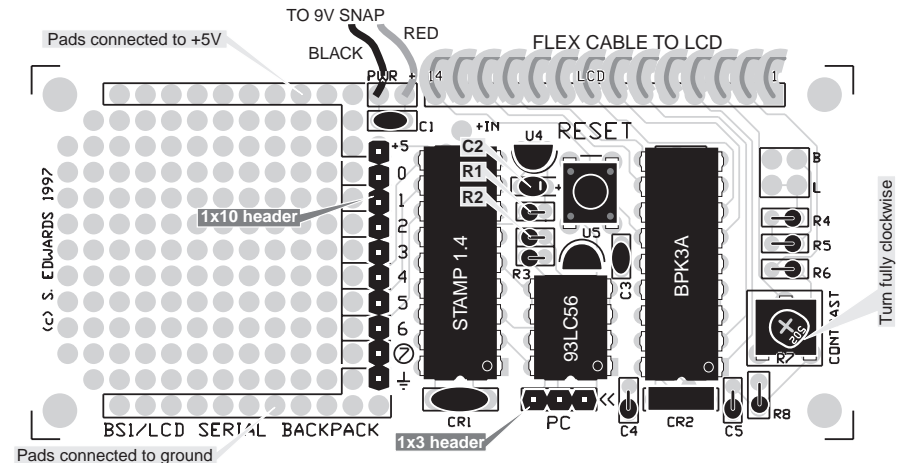# BS1/LCD Serial Backpack

☐ **with LED backlight**

☐ **non-backlit**

Scott Edwards Electronics

PO Box 160
Sierra Vista, AZ 85636-0160 USA
ph: 520-459-4802 • fax: 520-459-0623
e-mail: 72037.2612@compuserve.com

Assembling the BS1/LCD Serial Backpack Kit

Inventory and identify the parts included in this package. If any are missing or damaged, contact Scott Edwards Electronics for a replacement. Use a small (25W) soldering iron to solder the components to the BS1/LCD circuit board as shown in the drawing below. Do not install socketed ICs until all other assembly work is done.

QTY	PART #	DESCRIPTION	NOTES
7	R1—6, 8	4.7k, 1/8W resistor	Color code: yellow/violet/red. Bend into hairpin shape for installation.
1	R7	5k trim pot	Set fully clockwise
2	C1, C3	0.1µF capacitor	Marked "104" on body.
1	C2	10µF tantalum cap	Marked "10." Install longer lead (marked w/stripe) in hole marked +.
2	C4, C5	220pF capacitor	Marked "221." Bend into hairpin shape for installation.
1	U1	PBASIC PIC16C56	Solder 18-pin socket to pcb. Plug chip in after assembly is complete.
1	U2	93LC56 EEPROM	Solder 8-pin socket to pcb. Plug chip in after assembly is complete.
1	U3	BPK3A PIC16C52	Solder 18-pin socket to pcb. Plug chip in after assembly is complete.
1	U4	LP2950 5V regulator	Make sure that flat side matches outline printed on pcb.
1	U5	MN1381/TC44VC4303	Make sure that flat side matches outline printed on pcb.
1	S1	6mm pushbutton	
1	CR1	4MHz ceramic resonator	Marked "4.00M." No polarity—install either way.
1	CR2	480kHz ceramic	Marked "480K." No polarity—install either way.
1	J1	1x10 male header	Optional—may leave off if circuits will be soldered directly to I/O pads.
1	J2	1x3 male header	
1	J3	9V battery snap	Solder red wire to pad marked +, black to neighboring pad.
1	FLEX	(2) 1x7 or (1) 1x14 flex	Use to connect pcb to LCD as shown in drawing.
4	MS1	1" 2-56 machine screw	Use to secure pcb to LCD as shown in drawing.
12	MNUT1	2-56 machine nut	Use to secure pcb to LCD as shown in drawing.
1	PCB	BS1/LCD circuit board	Mount all components as shown below.
1	LCD	2x16 supertwist LCD	LED-backlit or non-backlit depending on type ordered.

Flex-cable connection between pcb and LCD module:

Mounting of pcb to LCD module:

Using the BS1/LCD Serial Backpack (BS1/LCD)

The BS1/LCD combines a BASIC Stamp I® (BS1) computer with an LCD Serial Backpack™ interface and a high-quality 2x16 liquid-crystal display in a single compact package. It uses a single BS1 input/output (I/O) pin to drive the display using a simple serial protocol. The BS1/LCD is ideal for measurement, control, and display projects based on the BS1.

Contents

Disclaimer of Liability

Scott Edwards Electronics is not responsible for any special, incidental, or consequential damages resulting from any breach of warranty, or under any legal theory, including lost profits, downtime, goodwill, damage to or replacement of equipment or property, and any costs or recovering, reprogramming, or reproducing of data associated with the use of the hardware or software described herein.

Trademarks, Copyrights

BASIC Stamp is a registered trademark of Parallax Inc. LCD Serial Backpack is a trademark of Scott Edwards Electronics. The contents of this manual in its entirety are copyright Scott Edwards Electronics 1997 and may not be reproduced by any means without express permission.

Initial Checkout

The EEPROM (U2) included with your kit is factory programmed with the program shown in listing 1. When you connect power to your assembled kit, the program displays a message on the LCD, and toggles pins 0 through 6 on and off. Adjust contrast control R7 clockwise until you can see the message on the LCD.

Programming the LCD

Displaying a message on the LCD requires sending the appropriate text serially at 2400 bps through pin 7 of the BS1 controller. For example:

```
Serout 7,N2400,("Hello")    ' Print "Hello" to LCD.
```

In PBASIC, displaying numbers requires you to use the # character before the constant or variable, as shown below:

```
Serout 7,N2400,(#99)        ' Print constant "99" to LCD.
Serout 7,N2400,(#b2)        ' Print value of b2 to LCD.
```

The LCD also accepts about a dozen instructions that clear the screen, move the cursor, scroll the display, etc. To distinguish instructions from ordinary data, your program must send the instruction-prefix code 254 (0FE hex). The LCD will treat the next byte it receives as an instruction. After it receives a one-byte instruction, the LCD automatically reverts to data mode. For example, to clear the screen, send the instruction-prefix followed by the clear-screen code from the table below. Here's an example:

```
Serout 7,N2400,(254,1)      ' Clear the LCD screen.
```

• NOTE: The instruction prefix and instructions must be sent as single-byte values, not as strings of text! This means leaving off the # symbol before these values. Here is a list of LCD instructions:

Instruction/Action	Code
Clear screen	1
Scroll display one character left	24
Scroll display one character right	28
Home (return cursor to upper left)	2
Move cursor one character left	16
Move cursor one character right	20
Turn on underline cursor	14
Turn on blinking-block cursor	13
Turn off cursor	12
Blank the display (retaining data)	8
Restore the display (w/out cursor)	12
Set display (DD) RAM address	128 + addr
Set character (CG) RAM address	64 + addr

The last two instructions in the list above let you write to specific locations in the LCD's memory. By setting the display data (DD) RAM address, you can control the location on the display at which subsequent characters will appear. This lets you update part of the display without having to reprint the entire screen. Program listing 2 shows how. See the drawing of the LCD on page 9 for a memory map of the 2x16 LCD.

There are also instructions for writing to the LCD's character-generator RAM (CG RAM). This lets you define character patterns to create simple graphics. The figure below shows how to calculate the data for these patterns. Listing 3 demonstrates how to define and display custom characters.

*Address in Character Generator RAM	Bit Map	Data (binary)	Data (value)
0	☐☐☐☐☐	00000	0
1	☐☐■☐☐	00100	4
2	☐■■■☐	01110	14
3	■■■■■	11111	31
4	☐■■■☐	01110	14
5	☐☐■☐☐	00100	4
6	☐☐☐☐☐	00000	0
7	☐☐☐☐☐	00000	0

*Note: To define one of the eight characters in CG RAM, put the LCD into instruction mode, then send the following value to the LCD: 64 + (8 x character_no.) where character_no. is in the range of 0–7. Then send the bit map data calculated as shown above. For example, to define character 3, send 64 + (8 x 3) = 88. If you are defining all of the CG RAM characters, start at 0 by sending 64.

Defining custom-character bitmaps.

Frequently Asked Questions

Here are some questions you may have while using the BS1/LCD kit:

• *What are the pads marked L and B for? They don't seem to serve any purpose.*

The LCD Serial Backpack circuit is capable of driving different styles of LCDs (1, 2, or 4-line types) and receiving data at 2400 or 9600 bps. In this application, it only needs one configuration: 2-line LCD/2400 bps, since that matches the characteristics of the LCD and the BS1 used in this kit. Still, it made sense to bring out the configuration connections for (L)ines and (B)ps in the event someone needs to get at them for a special application.

• *Why is the display blank? I'm sure that my program is sending data to it.*

Try turning the contrast pot, R7, clockwise all the way. Light-gray pixels should appear on the screen, even if there's no text to display. Is the temperature close to freezing? LCDs lose contrast at low temperatures—try moving the unit to a warmer location.

• *Why is the LCD displaying black pixels in all positions?*

The contrast may be set too high—turn R7 counterclockwise. Is the temperature very hot (above 100°F) or is the LCD in direct sunlight? High temperatures drive up LCD contrast. Move the LCD into the shade or to a cooler location.

• *Why does sending an instruction make the LCD lock up or otherwise misbehave?*

You may be sending the instruction as a text string instead of a single byte. Eliminate the formatting prefix # from values that you mean to send as instructions.

• *Where can I find more program examples for the BS1/LCD?*

Check out our Internet archive at ftp.nutsvolts.com. You'll find application examples for the LCD Serial Backpack in the directory pub/nutsvolts/scott. Read the file index.txt for a listing. When you adapt example programs, remember that in your kit the LCD is connected to pin 7 of the BS1—not all program listings use the same pin. Just change the pin used for serial output to the LCD to pin 7. For instance:

```
Serout 0,N2400,("Hello")    ' BEFORE: output on pin 0.

Serout 7,N2400,("Hello")    ' AFTER: output on pin 7.
```

• *What's the best way to use the prototyping area?*

This depends on the application and your skills/personal preferences. The grid of plated holes may be used to support wire-wrapping posts, or as anchoring points for point-to-point soldered wiring. The hinged design of the BS1/LCD board allows access to both sides of the circuit board while it remains connected to the LCD module. This makes testing and modifications relatively easy. However, if you expect to make lots of changes in your custom circuitry, consider building it on a plug-in breadboard and making temporary connections to the BS1/LCD. When the circuit is working the way you want it to, transfer it to the more permanent on-board proto area.

• *I have a specific application in mind. Will you design it for me?*

Scott Edwards Electronics does not provide custom design services, but we'll be happy to refer you to others who do.

• *I'd like to incorporate components of this kit into a product. Are they sold separately?*

We offer the LCD Serial Backpack controller chip at excellent quantity discounts; see our catalog or call our order desk for details. The PBASIC interpreter chip is a product of Parallax Inc., so you'll get the best pricing on that item from them (916-624-8333). The rest of the parts are standard, off-the-shelf components available from parts distributors everywhere.
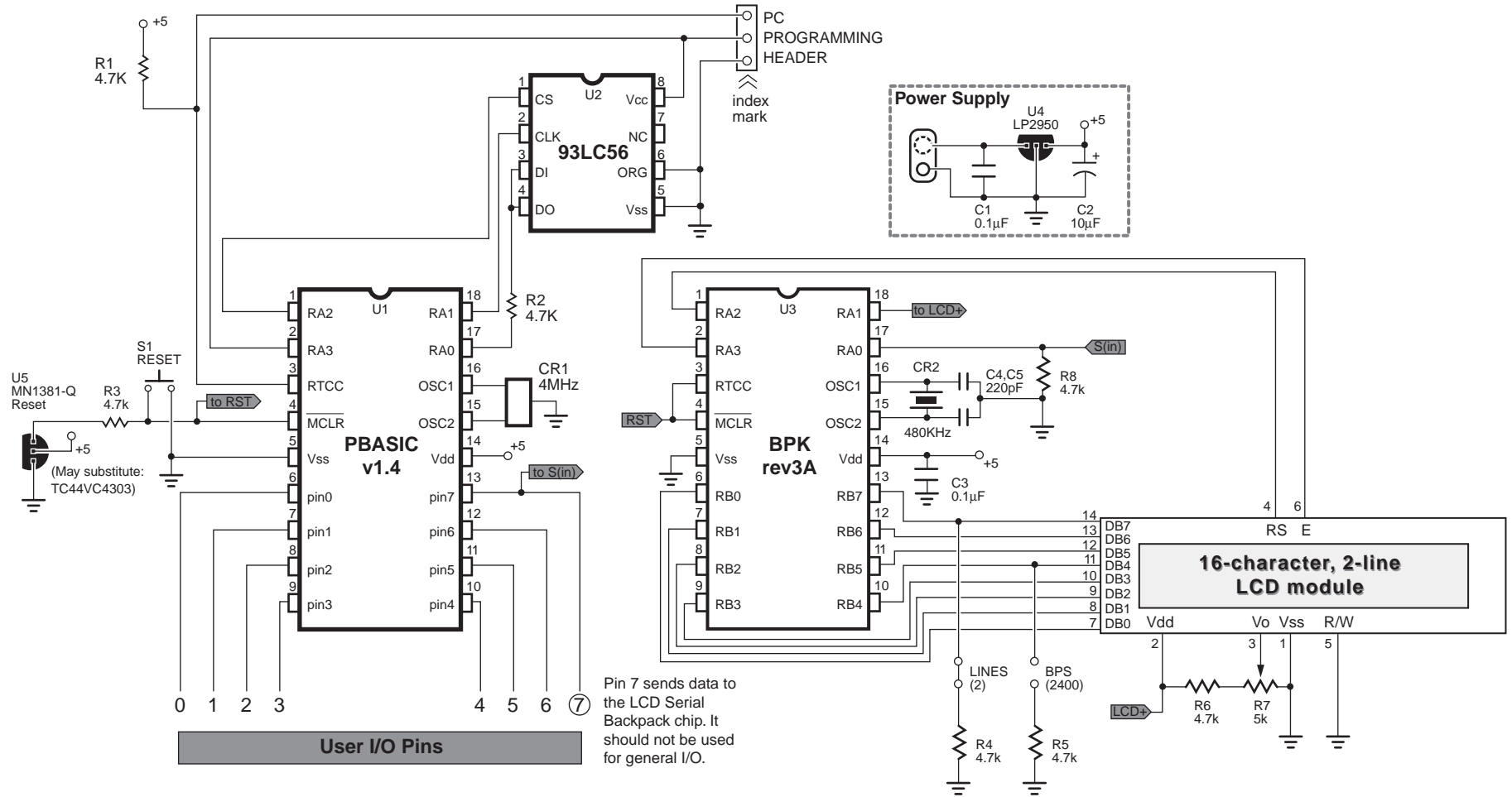
Schematic diagram of the BS1/LCD

Listing 1. Demo Program

```
' Program: LCD_OK.BAS (Preloaded demo program for BS1/LCD kit)
' This is the program that's loaded in the EEPROM that comes
' with the BS1/LCD Serial Backpack kit. When you finish the kit
' and apply power this program will run. It displays a message
' on the LCD, sets the seven user I/O pins to output, and
' goes into a loop toggling the pins and displaying their
' status on the LCD.

SYMBOL  lcd = 0         ' Serial output to LCD.
SYMBOL  I = 254         ' Instruction prefix for LCD.
SYMBOL  lcdCls = 1      ' Clear-screen instruction for LCD.
SYMBOL  line2 = 192     ' Position: start of line 2.
SYMBOL  pinPos = 200    ' Position: start of pin status line.
SYMBOL  thePin = b13    ' Byte variable to hold pin #.
SYMBOL  bitMask = b12   ' Bit pattern of thePin.
SYMBOL  result = b11    ' Result of ANDing bitMask with pins.

' ===Program Setup
pause 1000                        ' Wait for LCD to wake up.
serout 7,n2400,(I,lcdCls)         ' Clear the screen.
serout 7,n2400,("  BS1/LCD DEMO") ' Print message on line 1.
serout 7,n2400,(I,line2," pins:") ' Go to line 2; print "pins:".
let dirs = $0FF                   ' Set all pins to output.

' ===Main Program
' The program is a loop that toggles each of the output pins
' 0-6 in turn. It determines the state of the current output pin
' by ANDing it with a 'bit mask' with a 1 in the same position
' as the pin. If the result is 0, then the pin is 0; if the result
' is not 0, the pin is 1. The program prints the results as a
' line of characters ("0" or "1" for each pin) across the 2nd line
' of the display.
again:
  bitMask = 1                     ' Start with %00000001 in bitMask.
  serout 7,n2400,(I,pinPos)       ' Go to start of pin status line.
  for thePin = 0 to 6             ' For each of the output pins:
    toggle thePin                 ' -Toggle the current pin.
    result = bitMask & pins       ' -Get state of that pin in result.
    if result = 0 then skip       ' -Make result match state of
    result = 1                    ' -  current pin (0 or 1).
skip:
    serout 7,n2400,(#result)      ' -Show result as a number on LCD.
    bitMask = bitMask * 2         ' -Shift bitMask to match next bit.
    pause 100                     ' -Pause to slow the display.
  next                            ' Do the next pin.
goto again                        ' Repeat until power off.
```

### Listing 2. Program Demonstrating Print-At Capability

```
' Program: COUNT.BAS (Printing at a screen location with the Backpack)
' This program prints a fixed label, "Number:" on the LCD, followed
' by a count value that cycles from 0 to 20,000 by sending commands to
' the LCD Serial Backpack.
Symbol    I = 254               ' Instruction prefix.
Symbol    ClrLCD = 1            ' Clear-LCD instruction.
Symbol    prn_at = 136         ' Display RAM, address 8 (128+8).
Symbol    j = w2               ' 16-bit counter variable.
low 0                           ' Make the serial output low.
pause 1000                      ' Let the LCD wake up.
serout 7,N2400,(I,ClrLCD)       ' Clear the LCD.
serout 7,N2400,("Number: ")     ' Print the fixed label.
Loop:
  for j = 0 to 20000
  serout 7,N2400,(I,prn_at,#j,"   ")    ' Print j at address 8.
  pause 30                              ' Slow the count a little.
  next j                        ' Keep going to 20,000.
goto Loop                       ' Do it again.
```

Listing 3. Program Demonstrating Custom Characters

```
' Program: SPECHAR.BAS (Defining special characters with the Backpack)
' This program defines a diamond-shaped character shown on page 4,
' and writes it across a 16-character LCD using the LCD Serial Backpack.
Symbol    I = 254               ' Instruction prefix.
Symbol    ClrLCD = 1            ' Clear-LCD instruction.
Symbol    cgRAM = 64            ' Special-character RAM, address 0.
Symbol    ddRAM = 128           ' Display RAM, address 0.
Symbol    j = b2               ' 8-bit counter variable.
low 0                           ' Make serial output low.
pause 1000                      ' Let LCD wake up.
serout 7,N2400,(I,ClrLCD,I,cgRAM)   ' Clear LCD, enter CG RAM addr 0.
serout 7,N2400,(0,4,14,31,14,4,0,0) ' Define char 0 as diamond shape.
serout 7,N2400,(I,ddRAM)        ' Switch to display RAM address 0.
for j = 1 to 16                 ' Repeat 16 times.
  serout 7,N2400,(0)            ' Display character 0.
next                            ' Done.
end
```

LCD Dimensions, Display RAM Map, and Backlight

The physical layout and important dimensions of the 2x16 display sold with the BS1/LCD appear below. Please note that some of the mechanical specs may vary slightly from this drawing, as LCD manufacturers make changes/improvements in their products. If a particular dimension is critical, double-check it against an actual measurement.
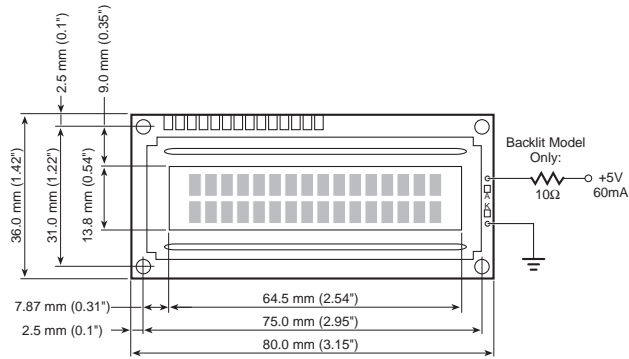
Accessing Locations on the Screen: The two lines of the display are mapped to the 80 bytes of internal display RAM as shown in the memory map below. Note that characters printed beyond the 16th character of either line do not appear on the display, but are stored in the LCD's display RAM. To view these characters, scroll the display left.

1st Line	address:	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	set-address command:	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143
2nd Line	address:	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79
	set-address command:	192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207

To print at a particular location on the screen, send the instruction prefix followed by the set-address command from the table above. For instance, to print at the beginning of the second line, you would send 254 (instruction prefix) followed by 192 (set-address command). See page 3 for more information on sending instructions to the LCD.

When power is first applied to the BS1/LCD, the screen is cleared, and the printing location is set to address 0—the first character of the first line.

Using the Backlight: The backlight is an LED array wired for approximately a 4.4-volt forward drop. Its + and – connections are isolated from those of the BS1/LCD and LCD module electronics. To light the backlight, connect a 5V regulated power supply (such as the +5V power from the BS1/LCD board) through a 10Ω series resistor. This combination will draw approximately 60 mA. If you don't require maximum brightness and would like to reduce current draw, try higher resistor values. (Please note that the standard model is not backlit; backlighting is an extra-cost option.)