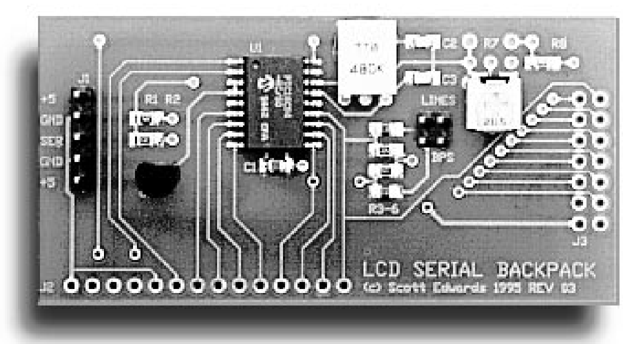
Installing and Using the

LCD Serial Backpack



Scott Edwards Electronics
964 Cactus Wren Lane
Sierra Vista, AZ 85635 USA
ph: 520-459-4802 • fax: 520-459-0623
e-mail: 72037.2612@compuserve.com

Using the LCD Serial Backpack rev. 3

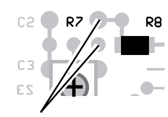
The LCD Serial Backpack is a daughterboard that attaches to common LCD modules. The Backpack receives data serially and displays it on the LCD, giving any computer with serial-output capability (e.g., PC or BASIC Stamp) an easy-to-use display. Through the Backpack's *instruction toggle*, the host computer can access the LCD's special effects, such as scrolling, cursors, and special characters.

Although the examples below concentrate on the BASIC Stamp, the Backpack may be directly connected to any RS-232 data source at 2400 or 9600 bps (no parity, 8 data bits, 1 stop bit). PC hookup details and programming examples start on page 7. PC users should still skim the Stamp-oriented material below to get acquainted with the Backpack's features and functions.

Configuring the Backpack

The Backpack offers several configuration options to accommodate most types of LCDs. As a default, it is configured for a one-line LCD requiring an adjustable, positive contrast voltage. The default serial rate (no jumper installed at BPS) is 2400 bps.

To provide an adjustable, negative contrast voltage (required for some LCDs to operate at low temperatures), connect the negative voltage source to one of the points shown in figure 1 and remove resistor R8.



Connect negative contrast voltage at one of these points on board

Fig. 1

To set up the Backpack for a two-line LCD, install one of the supplied jumper blocks on the posts marked LINES on the circuit board. To set the Backpack for 9600-baud operation, install one of the supplied jumper blocks on the posts marked BPS on the circuit board. Note that the Backpack only checks the LINES and BPS jumpers when it is first powered up. If you change these jumper settings while the Backpack is powered, the changes will take effect *after* it is turned off and back on.

Connecting the Backpack

The BASIC Stamp can talk to the Backpack through any of its eight input/output (I/O) pins. The Stamp's power supply can provide regulated +5 to the Backpack, which draws less than 2 mA. Figure 2 shows the layout of the power and serial connections. A Stamp I/O pin (pin 0 in the examples below) would be connected to the terminal marked SER. The +5 and ground connections are duplicated to allow you to make a reversible 5-pin connector. The connections will line up correctly whether the connector is installed rightside-up or upside-down.

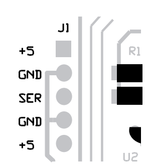


Fig. 2

Programming the Backpack (Stamp)

Displaying a message on the Backpack's LCD requires just one line of Stamp code:

```
Serout 0,N2400,("Hello")    ' Print "Hello" to LCD.
```

To display numbers, use the # character before the constant or variable:

```
Serout 0,N2400,(#b2)        ' Print value of b2 to LCD.
```

To clear the screen, you must send the LCD a clear-screen instruction. First you must put it into the instruction mode by sending the instruction-toggle command, 254 (0FE hex):

```
SYMBOL I = 254             ' Instruction-toggle command.
SYMBOL clear = 1           ' Clear-screen instruction code.
Serout 0,N2400,(I,clear,I) ' Clear the LCD screen.
```

After instructions you must toggle the LCD back into its normal mode. Otherwise, subsequent data sent to it are treated as instructions. Here's a list of useful LCD instructions:

Instruction/Action	**Code**
Clear screen	1
Scroll display left	24
Scroll display right	28
Display home (undo scrolling)	0
Move cursor left	16
Move cursor 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

Consult your LCD spec sheet for other instructions. They're normally shown as sequences of bits; you can convert them to decimal values as we have here, or enter them into your program as binary numbers. For example, the scroll-left instruction would be %00011000.

The last two instructions allow you to 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 listings 1 (Stamp) and 3 (PC) show how.

By entering values into character-generator (CG) RAM, you can define character patterns to create simple graphics. Figure 3 shows how to calculate the data for these patterns. Listings 2 (Stamp) and 4 (PC) demonstrate 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.

Figure 3. Defining custom-character bitmaps.

Hints on using LCDs and the Backpack

• When you purchase an LCD, get the spec sheet. It provides valuable information on things like the contrast voltage, character set, instructions, operating temperature range, etc.

• Some LCD modules have gaps in their memory organization because the manufacturer left out a memory expansion chip known as an H44100 or Oki 5259. With such modules, the first eight characters will display normally, but the rest of the characters will seem to be lost in space until character 65, which will again display normally. There are two ways to correct this problem. The easiest is to avoid buying LCDs that lack the expansion chips. The alternative is to count characters sent to the LCD. After the eighth character, send an instruction toggle, display RAM address 64 (128 + 64 = 192), and another instruction toggle. This forces the LCD to print the next eight characters to the screen, instead of off-screen memory locations. Listings 1 (Stamp) and 3 (PC) demonstrate how to print at a specific memory location.

• Two-line LCDs split their memory between the first and second lines. The second line generally begins at address 64. Again, use the print-at-location capability demonstrated in listing 1 to display text on the second line. If the second line doesn't appear, check the LINES header. It must be jumpered to configure the Backpack for two-line displays.

• Running a program (ALT-R) interrupts what the Stamp is doing at the time, including sending serial data. The Backpack may receive random data during reprogramming. As a result, the display may freeze or display garbage. You can correct these problems by momentarily disconnecting power from the Backpack, then reconnecting it.

• The Backpack requires about 0.5 seconds to initialize the LCD and prepare to receive data. Your program should delay approximately a second after power-on before sending data to the Backpack.

Notes for 9600-baud operation. The LCD executes most of the instructions relayed to it by the Backpack within a few microseconds. However, clearing the screen and moving the cursor home can take a couple of *milli*seconds. During this time, the LCD cannot receive new data or instructions. At 2400 baud, data takes more than 4 milliseconds to arrive, so this is no problem. At 9600 baud the LCD might miss the first byte following a clear or home instruction. If necessary, insert a delay to prevent this.

Changes in the rev 3 hardware and firmware. The rev 3 Backpacks are marked REV03 in the lower right corner of the circuit board. The rev 3 firmware does not support the 254 passthrough feature of rev 1, which allowed the second of two instruction toggle bytes (ASCII 254s) to be sent to the LCD. In the new firmware, the value 254 *only* serves as an instruction toggle and is *never* written to the LCD. The passthrough feature caused problems with sequences of instructions that could send an unwanted 254 to the LCD.

Changes to the rev 3 hardware and firmware lets you configure the Backpack for 9600-baud operation by installing a shorting block at BPS on the circuit board.

The rev 3 hardware includes a circuit to ensure that the Backpack is reset when the supply drops below approximately 4 volts. Previous Backpacks could be kept partly awake by voltage supplied through the Stamp programming cable—even with the battery disconnected. The reset feature eliminates this problem.

The rev 3 hardware includes a resistor in series with the contrast pot to restrict the range of contrast voltage to 0 to 1.7 volts. This allows finer adjustment of the contrast setting.

Listing 1a. Basic Stamp (BS1) Program Demonstrating Print-At Capability

```
' Program: COUNT.BS1 (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 toggle command.
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 0,N2400,(I,ClrLCD,I)          ' Clear the LCD.
serout 0,N2400,("Number: ")          ' Print the fixed label.
Loop:
  for j = 0 to 20000
  serout 0,N2400,(I,prn_at,I,#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 1b. Basic Stamp (BS2) Program Demonstrating Print-At Capability

```
' Program: COUNT.BS2 (Print-at-location with new Stamp 2)
' This is the same as the previous program, but modified for the
' new BS2 model. Since this program sends data to the Backpack
' at 9600 baud, be sure to install a jumper at BPS on the board.
' :::Note::: The new Serout uses square brackets [] instead of
' parentheses () to enclose output data and has a new system for
' specifying baud rate and mode. See the manual for details.
I        con   254                   ' Instruction toggle command.
ClrLCD   con   1                     ' Clear-LCD instruction.
prn_at   con   136                   ' Display RAM, address 8 (128+8).
j        var   word                  ' 16-bit counter variable.
N96N     con   $4054                 ' 9600 baud, inverted, no parity.
low 0                                ' Make the serial output low
pause 1000                           ' Let the LCD wake up.
serout 0,N96N,[I,ClrLCD,I]           ' Clear the LCD.
serout 0,N96N,["Number: "]           ' Print the fixed label.
Loop:
  for j = 0 to 20000
  serout 0,N96N,[I,prn_at,I,DEC j,"    "]   ' Print j at address 8.
  pause 30                                  ' Slow the count a little.
  next                                      ' Keep going to 20,000.
goto Loop                                   ' Do it again.
```

Listing 2a. Basic Stamp (BS1) Program Demonstrating Custom Characters

```
' Program: SPECHAR.BS1 (Defining special characters with the Backpack)
' This program defines the diamond-shaped character shown in figure 3,
' and writes it across a 16-character LCD using the LCD Serial Backpack.
Symbol    I = 254                        ' Instruction toggle command.
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 0,N2400,(I,ClrLCD,cgRAM,I)        ' Clear LCD, enter CG RAM addr 0.
serout 0,N2400,(0,4,14,31,14,4,0,0)      ' Define char 0 as diamond shape.
serout 0,N2400,(I,ddRAM,I)               ' Switch to display RAM address 0.
for j = 1 to 16                          ' Repeat 16 times.
   serout 0,N2400,(0)                    ' Display character 0.
next                                     ' Done.
end
```

Listing 2b. Basic Stamp (BS2) Program Demonstrating Custom Characters

```
' Program: SPECHAR.BS2 (Backpack/special characters for BS2)
' This is the same as the previous program, but modified for the
' new BS2 model. Since this program sends data to the Backpack
' at 9600 baud, be sure to install a jumper at BPS on the board.
' :::Note::: At 9600 baud, data can arrive faster than the LCD
' can accept it after a clear or home-cursor instruction. The
' program below handles this with a 1-millisecond delay between
' the clear instruction and the cgRAM instruction that follows.
I        con   254                       ' Instruction toggle command.
ClrLCD   con   1                         ' Clear-LCD instruction.
cgRAM    con   64                        ' Special-character RAM, address 0.
ddRAM    con   128                       ' Display RAM, address 0.
j        var   byte                      ' 8-bit counter variable.
N96N     con   $4054                     ' 9600 baud, inverted, no parity.
low 0                                    ' Make the serial output low
pause 1000                               ' Let the LCD wake up.
serout 0,N96N,[I,ClrLCD]                 ' Clear the LCD.
pause 1                                  ' Wait to finish.
serout 0,N96N,[cgRAM,I]                  ' Enter CG RAM address 0.
serout 0,N96N,[0,4,14,31,14,4,0,0]       ' Define char 0 as diamond shape.
serout 0,N96N,[I,ddRAM,I]                ' Switch to display RAM 0.
for j = 1 to 16                          ' Repeat 16 times.
  serout 0,N96N,[0]                      ' Display character 0.
next                                     ' Done.
Stop                                     ' End program.
```

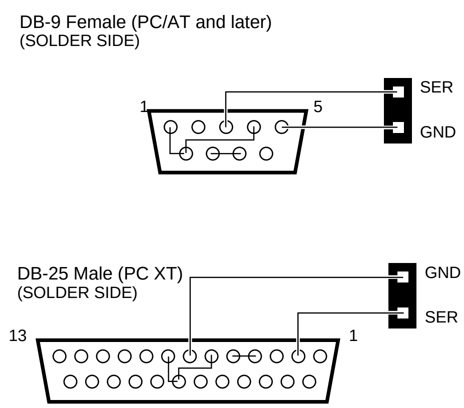
PC-Specific Instructions

Why connect a Backpack-equipped LCD to a PC? Low cost and ease of software development have led to PCs being used as controllers and data-acquisition units. In these roles a big CRT monitor is overkill, but some kind of display is still necessary. Most programming languages make it simple to send data to a serial (COM) port. And the Backpack can receive data directly without any additional hardware, making it a logical choice for embedded-PC displays.

Connecting the Backpack to a PC

Figure 5 below shows how to connect the serial output of PC COM ports to the Backpack. The diagrams include additional connections that loop back the port's handshaking lines. The Backpack doesn't use these connections, and some software requires handshaking in order to work properly. (The example QBASIC programs in listings 3 and 4 disable handshaking when they open the COM port, so the loopback connections aren't required.)

Since typical LCD modules draw less than 2 mA and the Backpack less than 1 mA, it's feasible to steal enough power from a COM port's unused handshaking lines to run the display. This eliminates the need for batteries or external power supplies. Figure 6 shows a parasite power supply. This supply will only operate with full-fledged COM ports; some lap-tops and organizer gadgets cut corners in port design to conserve battery power, and won't work with the circuit. The supply is energized only when the handshaking lines are active. In terms of QBASIC, this means that the supply is turned on by OPENing the COM port and off by CLOSEing it.



Note: The additional connections shown above are loopbacks for the handshaking lines. These deactivate hardware handshaking required by some terminal software.

Figure 5. Backpack serial connections for PCs and compatibles

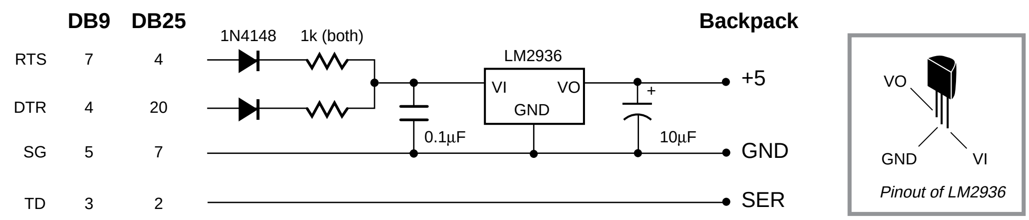


Figure 6. Parasite power supply.

The parasite supply does not loop back the handshaking lines, since its use assumes that you have control over handshaking in software (such as a QBASIC program).

Programming the Backpack (PC)

Displaying a message on the Backpack's LCD requires opening the COM port and PRINTing to it, as in this QBASIC example:

```
' Open the port with all handshaking off (set to zeros).
OPEN "COM1:2400,N,8,1,CD0,CS0,DS0,OP0" FOR OUTPUT AS #1
PRINT #1, "Hello";             ' Print "Hello" to the LCD.
```

Once the COM port is open, any data PRINTed to its file number (#1 in the example above) will go to the Backpack. Note that PRINT statements must end in a semicolon (;). This prevents QBASIC from tacking a carriage return and linefeed onto the end of the data. To the LCD these codes have no control significance—it displays them as graphics characters.

To clear the screen, you must send the LCD the proper instruction. First you put it into the instruction mode by sending the instruction-toggle, 254 (0FE hex); send the clear-screen code (1); then toggle back to the data mode. Assuming the COM port is already OPEN, the code looks like this:

```
PRINT #1, CHR$(254);CHR$(1);CHR$(254); ' Clear LCD screen.
```

The table on page 3 lists the most commonly used LCD instructions, and explains how to print at specific screen locations and define custom characters. Page 4 provides hints on selecting and using LCDs. Listings 3 and 4 (pages 9 and 10) are QBASIC programs that illustrate print-at and custom-character features.

Compatibility with other computers and controllers

Any device capable of sending serial data (2400 or 9600 baud, N81) can be used with the Backpack. The only restriction is that the device be able to suppress generation of linefeeds and carriage returns (LF/CR).

The only incompatible devices we've found are single-board computers based on the 8052-AH BASIC chip. BASIC-52's PRINT instruction will suppress LF/CRs until an internal counter reaches the end of an imaginary line of text. Then it unconditionally inserts an LF/CR. If you purchased a Backpack to use with BASIC-52, contact us about an alternative model with LF/CR filtering (contact info on the cover of this booklet). We did not incorporate this feature into all Backpacks, because it requires a faster system clock and therefore more current draw (3 mA compared to less than 1 mA).

Listing 3. QBASIC Program Demonstrating Print-At Capability

```
' Program: QCOUNT.BAS (QBASIC Example of printing at a screen location)
' This program prints a fixed label, "Number:" on the LCD, followed by
' a count value that cycles from 0 to 20,000.

' Clear PC monitor screen and wait for a keystroke.
CLS : INPUT "Press <Return> to begin. ", dummy$

' Set up the serial port: baud, parity, data bits, stop bits,
' disable all handshaking (CD, CS, DS, OP). If you have the Backpack
' set for 9600 baud (jumper installed at BPS), be sure to change 2400
' to 9600 in the OPEN statement.
OPEN "COM1:2400,N,8,1,CD0,CS0,DS0,OP0" FOR OUTPUT AS #1
PRINT : PRINT "Once the demo has started, press any key to end."

'Clear the LCD screen and pause (SLEEP) for 1 second.
PRINT #1, CHR$(254); CHR$(1); CHR$(254);
SLEEP 1

' Print the label to the LCD screen.
PRINT #1, "Number:";

' To locate the cursor (visible or not) for the next printing command,
' send (128+cursor position) as an instruction. In the Counter loop below,
' we want to print at position 8, so we send 128+8 = 136 between instruction
' toggles (254s). Don't worry about the PLAY command in the FOR/NEXT loop;
' its only purpose is to create a time delay of consistent duration on all
' PCs, slow or fast.
Counter:
  FOR I = 0 TO 20000
    PRINT #1, CHR$(254); CHR$(136); CHR$(254);  ' Position cursor.
    PRINT #1, I; "     ";                       ' Print number, plus spaces.
    IF INKEY$ <> "" THEN END                    ' Look for key press.
    PLAY "P16"                                  ' Play silent note to pause.
  NEXT I
GOTO Counter
```

Listing 4. QBASIC Program Demonstrating Custom Characters

```
' Program: QCHAR.BAS (QBASIC Example of defining special characters)
' This program defines the diamond-shaped character shown in figure 3,
' then writes it across the width of a 16-character LCD.

' Clear PC monitor screen and wait for a keystroke.
CLS : INPUT "Press <Return> to begin. ", dummy$

' Set up the serial port: baud, parity, data bits, stop bits,
' disable all handshaking (CD, CS, DS, OP). If you have the Backpack
' set for 9600 baud (jumper installed at BPS), be sure to change 2400
' to 9600 in the OPEN statement.
OPEN "COM1:2400,N,8,1,CD0,CS0,DS0,OP0" FOR OUTPUT AS #1
PRINT : PRINT "Once the demo has started, press any key to end."

'Clear the LCD screen [CHR$(1)] and enter character-generator (CG) RAM
' address 0 [CHR$(64)].
PRINT #1, CHR$(254); CHR$(1); CHR$(64); CHR$(254);

' Send data to CG RAM defining diamond-shaped character of figure 3.
PRINT #1, CHR$(0); CHR$(4); CHR$(14); CHR$(31);
PRINT #1, CHR$(14); CHR$(4); CHR$(0); CHR$(0);

' Now switch to display (DD) RAM address 0.
PRINT #1, CHR$(254); CHR$(128); CHR$(254);

' Print newly defined character 0 across the screen.
FOR I = 1 TO 16
    PRINT #1, CHR$(0);
NEXT I
END
```