Using the

4x20 Serial LCD Module

☒ **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

Copyright © Scott Edwards, 1996. All rights reserved.

4x20 Serial LCD Module with Backpack Plus

The 4x20 Serial LCD Module with Backpack Plus receives serial data at 2400 or 9600 bits per second (bps) and displays it on a 4-line by 20-character liquid-crystal display. The Backpack Plus understands common terminal/printer control codes (such as linefeeds, carriage returns, tabs, bell, backspace, formfeeds) for easy formatting. It extends the terminal model with additional features like backlight control, column-clear, and cursor positioning. A compatibility mode allows the Backpack Plus to emulate older LCD Serial Backpack products.

Contents of this Manual

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.

Warranty, Return/Repair/Replacement Policy

Scott Edwards Electronics warrants the 4x20 Serial LCD Module with Backpack Plus against defects in materials and workmanship for a period of 90 days. If you discover a defect, we will, at our option, repair, replace, or refund the purchase price. Return the product with a description of the problem. We will return your product or its replacement via standard shipping. Expedited shipping is available at the customer’s expense.

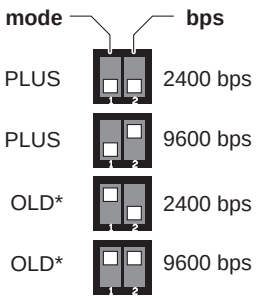
• **NOTE:** Physically abusing the module, removing the LCD Serial Backpack Plus daughterboard from the LCD, or attempting to repair or modify the module or the Backpack Plus, voids this warranty.

Terminology

The 4x20 Serial LCD Module consists of two major assemblies, the LCD module with the viewing screen, mounting holes and LCD-driver circuitry, and the LCD Serial Backpack Plus, a daughterboard mounted to the back of the LCD module. All connections and adjustments to the Serial LCD Module are made on the Backpack Plus circuit board.

Mode and Serial Data Rate

A pair of switches on the Backpack Plus allow you to set its mode and serial data rate. The switch marked "1" sets the mode, and should be left in the down position to take advantage of the Backpack Plus features documented in this manual. The OLD mode is compatible with our older 4x20 serial LCD modules, and should be used only with software written for the original LCD Serial Backpack. For more information on OLD mode, see page 9.



*Set the switches with power **OFF**. The Backpack Plus only reads switch settings at startup.*

**OLD* mode is for compatibility with original LCD Serial Backpacks. New users should select PLUS mode.*

***Tip:** A toothpick or straightened paperclip makes a handy tool for setting these switches.*

The switch marked "2" sets the serial data rate to 2400 (down) or 9600 (up) bits per second. At either speed, the Backpack Plus uses a fixed protocol of

no parity 8 data bits 1 stop bit

Power Requirements

The combined power requirements of the Backpack Plus and 4x20 LCD are:

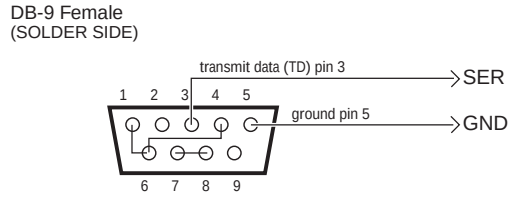
- Non-backlit or backlight off: 5 Vdc ± 0.5V at less than 8mA
- Backlight on: 5 Vdc ± 0.5V at less than 80mA

Connect the power source to the terminals of J1 marked +5 and GND. Note that there are two sets of power and ground connections; they are identical and it does not matter which you use. The layout lets you make a reversible connector that will line up correctly rightside up or upside down.

Connecting the Serial Input

The Backpack Plus accepts serial data from RS-232 (±12-volt) or logic-level (5-volt) sources. The figure below shows connections for PCs and other devices that follow the customary RS-232 connector layout.

Only two connections are needed—transmit data (TD) and signal ground. The diagrams show other connections between pins of the DB9 and DB25 connectors. These are required only if the software you'll be using to send data to the Backpack Plus uses hardware handshaking, and cannot be configured to turn it off ("flow control = none" or "handshaking off").



The connections between pins (i.e., 1,6,4; 7,8) are required only if your software's hardware handshaking cannot be turned off.

If you are using this module with a single-board computer or other controller, see the manufacturer's documentation for information on connecting serial devices. If the serial output is logic-level (0–5V), make sure that it is inverted. In terms of serial communication, this means that the stop-bit condition should be low (0V) and the start-bit condition should be high (3.5V or higher).

Connecting a Piezo Buzzer "Bell"

You may connect a piezo buzzer to the pads marked J4 BELL. The buzzer will beep briefly when the Backpack Plus receives the ASCII BELL character, cntl-G. The piezo buzzer must meet the following requirements:

- Built-in driver (oscillator) circuitry
- Operable on 5Vdc at 25mA or less

Suitable units are available from Jameco (ph: 1-800-831-4242; www.jameco.com; PN 76021), Radio Shack (PN 273-065), Digi-Key (1-800-344-4539; www.digikey.com; PN P9948). If you don't need a buzzer/bell in your application, leave J4 open. *Connecting anything other than a suitable piezo buzzer to J4 will void the warranty.*

When you connect the buzzer, make sure that the + connection on the buzzer goes to the pad of J4 marked +.

Optional Backlight

If your unit is backlit, the backlight is connected to the pads marked J5 AUX on the Backpack Plus circuit board. The backlight may be turned on and off in plus mode by sending control characters: cntl-N = ON; cntl-O = OFF. In OLD mode, turn the backlight on by sending the two-byte sequence <254> <255> and off with <254> <0>. On non-backlit units, J5 is not connected. *Connecting anything other than the factory installed backlight to J5 will void the warranty.*

Display Contrast

R8, marked CONTRAST on the Backpack Plus board, lets you adjust the contrast of the LCD display. You should *not* adjust this control until you have completed an initial power-on checkout of the display, since the range of adjustment includes settings at which the screen will be blank. If this happens, turn the control fully clockwise. Then back off the control to find the optimum setting.

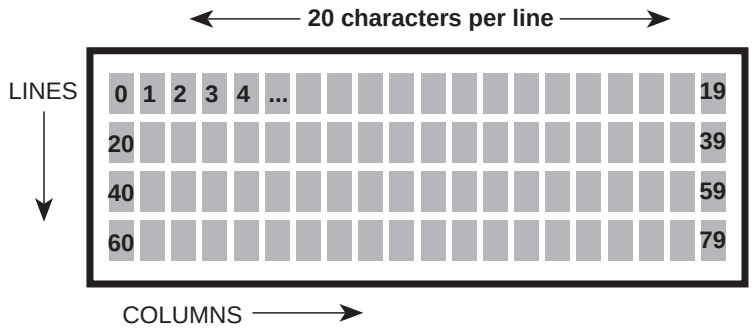
Initial Checkout and Basic Operation

The best way to check out your 4x20 Serial LCD Module is to connect power and serial data as outlined in the previous sections, boot up a terminal communications program (like the Windows Terminal accessory or Procomm for DOS), and type some text. The text you type in your terminal software will display on the LCD.

When you set up your terminal program to talk to the display, remember to

- Set the program and the display for the same baud rate.
- Configure the program for no parity, 8 data bits, 1 stop bit.
- Turn off any hardware handshaking.
- Set the program to talk to the COM port to which the display is connected.
- To see your typing on the PC monitor, set the program for "half duplex."

With the display powered and connected, anything you type in the terminal program will also appear on the LCD screen. If it does not, check the settings and connection. If text is garbled, chances are that the baud rate or parity is set incorrectly.



Layout of the 4x20 LCD.

As you send text to the display, characters print from left to right. When they reach the end of a line, the next character appears at the beginning of the next line. When text reaches the end of the display (character position 79) the next character appears at the top left corner position (character 0) of the display.

The display will respond to the following special-purpose keys on your keyboard. In these descriptions, *cursor* means the position in which the next character typed will appear. The default mode is invisible cursor.

TAB: The cursor will jump to the next multiple-of-4 column position (see the figure on the previous page).

RETURN: The cursor will jump to the beginning of the next line.

BACKSPACE: The cursor will back up one column, erasing that location.

Control Codes and Special Features

The table below lists the control codes and their functions. To send control codes from most terminal programs, hold down the control key and press another key. For example, to send cntl-E (turn on underline cursor), hold down control and press E.

ASCII VALUE	CODE	ASCII NAME	4X40 SERIAL LCD FUNCTION
0	cntl-@	NUL	ignored prior to buffer
1	cntl-A	SOH	send cursor home (position 0)
2	cntl-B	STX	begin big numeric display
3	cntl-C	ETX	ignored
4	cntl-D	EOT	blank cursor
5	cntl-E	ENQ	underline cursor
6	cntl-F	ACK	blinking-block cursor
7	cntl-G	BEL	pulse buzzer output
8	cntl-H	BS	backspace
9	cntl-I	HTAB	tab to next multiple-of-4 column
10	cntl-J	LF	smart linefeed; move down one row (line)
11	cntl-K	VTAB	vertical tab; move up one row
12	cntl-L	FF	form feed; clear entire screen
13	cntl-M	CR	return; move to first column of next line
14	cntl-N	SO	turn backlight on (if backlit display)
15	cntl-O	SI	turn backlight off (if backlit display)
16	cntl-P	DLE	accept cursor-position data, text or binary
17	cntl-Q	DC1	clear column
18—31	–	–	all ignored

Most of the control codes are self-explanatory, so feel free to experiment. If you need more information about a particular code, see the detailed descriptions that follow.

NULL

ASCII 0 *Control-@*

The Backpack Plus ignores nulls without storing them in its data buffer. Sending a null is the equivalent of a brief time delay. Delay lengths depend on serial speed:

Speed (bps)	Null delay (ms)
2400	4.16
9600	1.04

CURSOR HOME

ASCII 1 *Control-A*

Control-A moves the cursor to position 0 (upper left corner) of the display.

BEGIN BIG NUMERIC DISPLAY

ASCII 2 *Control-B*

Control-B puts the display into big-number mode. In this mode, the following characters are displayed as graphics spanning all four lines of the display:

`0 1 2 3 4 5 6 7 8 9 - . : <sp>`

(numbers 0–9, minus, decimal point, colon, and space)

The display stays in big-number mode until it receives a non-numeric character (other than null).

The numerals 0–9 are four columns wide; the minus, point, colon and space are two columns wide. Bear this in mind if you want to overwrite old numbers with new ones. Differences in character width can prevent new numbers from completely erasing the numbers they replace. To clear a four-line-high portion of the screen, use the clear-column feature, control-Q. Also be aware that the two-column characters (- . : <space>) overwrite a space four columns wide, then move back to 'trim' the extra width. This erases text two columns to the right of these characters. This is not a problem in normal, left-to-right printing.

Don't position big numbers so close to the right edge of the screen that they split and wrap back to the left edge. The lefthand portion of the characters will be scrambled.

Displaying big numbers takes several milliseconds. The display can receive and buffer up to 32 bytes of data, so this is not normally a problem. However, if big numbers are sent continuously, the buffer will fill up and data will be lost. Make sure your program allows 10ms for each big numeric character displayed. If your programming language doesn't support brief time delays, just send an appropriate number of null characters.

Try your program without delays to determine whether they are needed. In many cases, other processing takes enough time that delays are not necessary.

IGNORED

ASCII 3 *Control-C*

Control-C is ignored but takes space in the buffer. Use control-@ if you need a time delay. Many systems use control-C as a break character to halt execution; this will not affect the display.

BLANK CURSOR

ASCII 4 *Control-D*

Control-D makes the cursor invisible. If the cursor is already invisible, control-D has no effect.

UNDERLINE CURSOR

ASCII 5 *Control-E*

Control-E turns on a steady (non-blinking) underline cursor. To make the cursor invisible, send control-D.

BLINKING-BLOCK CURSOR

ASCII 6 *Control-F*

Control-F turns on a blinking-block cursor. To make the cursor invisible, send control-D.

BELL

ASCII 7 *Control-G*

Control-G pulses J4, the piezo-buzzer output, for approximately 100 ms.

BACKSPACE

ASCII 8 *Control-H* *(Bksp Key)*

Backspace, which may also be sent as control-H, causes the cursor to back up one column and print a space, leaving the cursor in that column.

HORIZONTAL TAB

ASCII 9 *Control-I* *(Tab Key)*

Tab, which may also be sent as control-I, causes the cursor to jump to the next multiple-of-four column position without otherwise affecting the display. For example, if the cursor is at position 0, sending TAB moves it to position 4. Tabs wrap to the next line, or from the last line back to the first line.

SMART LINEFEED

ASCII 10 *Control-J*

Control-J causes the cursor to drop down to the same column of the next display line. If the cursor is on the last line, it will wrap to the first line. The linefeed function is *smart* because it ignores redundant linefeeds sent immediately after a carriage return.

VERTICAL TAB

ASCII 11 *Control-K*

Control-K causes the cursor to move up to the same column of the preceding display line. If the cursor is on the first line, it will wrap to the last line.

CLEAR SCREEN

ASCII 12 *Control-L*

Control-L clears the entire display screen and moves the cursor to position 0 (upper left corner) of the display.

RETURN

ASCII 13 *Control-M* *(Return Key)*

Return, which may also be sent as control-M, sends the cursor to the first column of the next line of the display. If return is immediately followed by a linefeed, the linefeed will be ignored.

BACKLIGHT ON (IF EQUIPPED)

ASCII 14 *Control-N*

Control-N turns on the LED backlight, if installed. If not, control-N is ignored.

BACKLIGHT OFF (IF EQUIPPED)

ASCII 15 *Control-O*

Control-O turns off the LED backlight, if installed. If not, control-O is ignored.

POSITION CURSOR

ASCII 16 *Control-P*

Control-P puts the display into cursor-positioning mode. In this mode, there are two ways to move the cursor to a particular position on the screen:

Text method: Send the display position as text. For example, from a terminal program, press control-P, then type "63" followed by a space (to exit the mode). As soon as the space is typed, the cursor will jump to position 63 (fourth character of the fourth line). Note that the space (or other non-numeric character other than null) that terminates position mode is ignored.

One-byte binary method: Send the display position as a single byte value equal to the position plus 64. For example, from a terminal program, press control-P, then type "A". The cursor will jump to position 1 (top line, second column from the left) because the ASCII code for A is 65. The Backpack Plus subtracts 64 from the binary value to arrive at the screen position.

With either method the Backpack Plus will accept values larger than 79 (the highest valid position on a 4x20 screen). The cursor will wrap around to *position–80*. For example, if the position value is 85, the cursor will go to position 5. With the text method, it is possible (though not sensible) to send any decimal number as a position value. The number will be truncated to 8 bits *and* if larger than 79, wrapped around to *position–80* (with 80 subtracted as many times as necessary to make the result

less than 80). For example, the value 999 (3E7 hex) would be truncated to 8 bits (231 or 0E7 hex), then wrapped to 231–80–80 = 71.

CLEAR COLUMN

ASCII 17 *Control-Q*

Control-Q clears the current column, and advances the cursor to the next column, on the same line. This feature can be used to quickly clear a four-line-high section of the display in preparation for printing big numeric characters.

IGNORED

ASCII 18–31 *(misc. control characters)*

ASCII codes 18 through 31 are ignored, but do take space in the buffer. Use control-@ if you need a time delay.

PRINTABLE CHARACTERS

ASCII 32–255 *(alphanumeric characters)*

The chart on the next page shows the LCD bit patterns for the printable characters, 32–255. Characters from 32–127 correspond for the most part to the standard keys on a terminal/PC keyboard. Characters from 128 through 134 are symbols used to draw the big-number characters; 135 through 160 are blanks; and 161 through 255 are miscellaneous special symbols and Japanese Kanji characters.

"OLD" Compatibility Mode

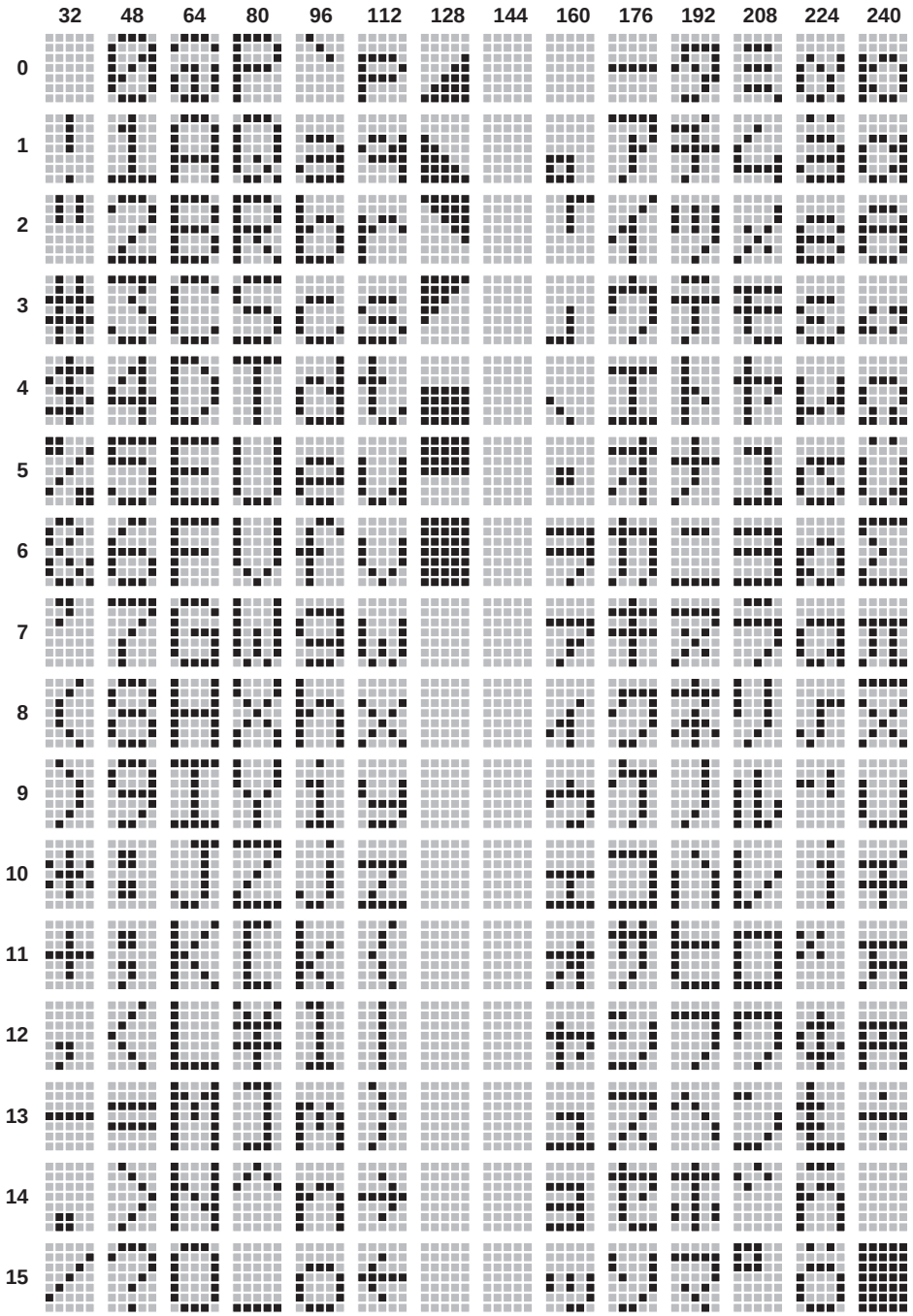
Many 4x20 serial LCD modules were sold with the original LCD Serial Backpack™ interface. To avoid forcing users to modify programs written for that product, we included a compatibility mode in the LCD Serial Backpack Plus for 4x20 displays. In OLD mode, the Backpack Plus responds to the same instructions as the original Backpack. However, there are a few inevitable differences between the two:

- The Plus/4x20 LCD typically draws 4.5mA versus 2mA for the older unit.
- The Plus does not require a delay after clear-screen instructions at 9600 bps.
- The Plus can switch the LED backlight on and off via serial commands; send bytes <254> <255> for ON and <254> <0> for OFF.

If you are a new user of this product, you should probably ignore the OLD compatibility mode. The newer PLUS mode is more convenient for most applications. However, if you need direct access to the LCD controller for special effects, you may want to use OLD mode anyway. Refer to the documentation for the original LCD Serial Backpack, available via Internet from ftp.nutsvolts.com/pub/nutsvolts/scott. The file name is BPK_MNL.PDF. Or you may request a free hardcopy by mail. (Sorry, this document is not available via fax.)

Character Code Chart for 4x20 LCD Modules

To find the ASCII code for a character or symbol, add the row and column numbers from the chart below. For example, capital S is in the column marked 80, row 3, so the ASCII code for S is 83.

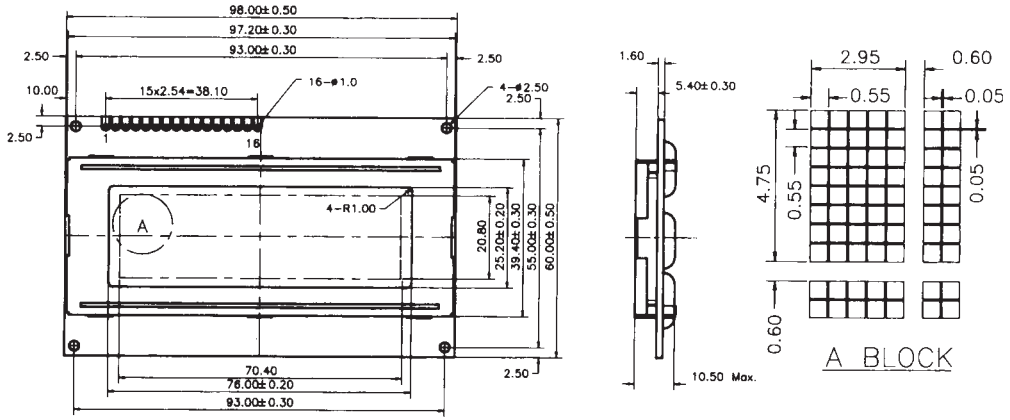


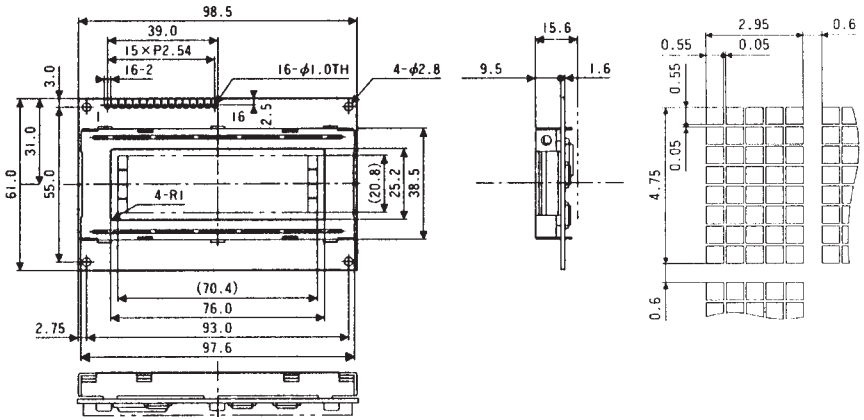

```
' Program: 420DEMO.BAS (Demonstrate 4x20 Serial LCD with Stamp I)
' This program demonstrates many of the features of the 4x20 Serial
' LCD Module from Scott Edwards Electronics. Set the module to
' 2400 bps (S1: both switches down). Connect the serial input
' of the module to Stamp pin 0, GND to GND and +5V to +5V.
' Run this program.

' ===========Define names for LCD instructions==========
SYMBOL noCurs = 4                   ' Make cursor invisible.
SYMBOL ulCurs = 5                   ' Show underline cursor.
SYMBOL clrLCD = 12                  ' Clear entire LCD screen.
SYMBOL posCmd = 16                  ' Position cursor.
SYMBOL clrCol = 17                  ' Clear column.
SYMBOL bell  = 7                    ' Ring bell.
SYMBOL wedge = 128                  ' Wedge-shaped symbol.
SYMBOL bksp  = 8                    ' Backspace.
SYMBOL bigNums = 2                  ' Begin big numbers.

' =================Begin demonstration=================
pause 1000                          ' Wait a sec.
serout 0,n2400,(clrLCD)             ' Clear the screen.
for b2 = 0 to 79
  serout 0,n2400,(wedge)            ' Fill screen with wedges.
next
pause 1000                          ' Wait 1 second.
serout 0,n2400,(posCmd,68)          ' Move to position 4 (64+4= 68).
for b2 = 1 to 12                    ' Clear a 12-character swath with
  pause 350                         ' ...the clear-column feature.
  serout 0,n2400,(clrCol,bell)      ' Ring bell (if available) each loop.
next

' Turn on the underline cursor, move to pos. 26, and show message.
serout 0,n2400,(ulCurs,posCmd,90,"4x20 LCD")
pause 1000                          ' Wait 1 second.

for b2 = 1 to 8
  pause 100                         ' Backspace to erase message.
  serout 0,n2400,(bksp,bell)        ' Ring bell (if available) each loop.
next
pause 1000                          ' Wait 1 second.
serout 0,n2400,(noCurs)             ' Turn cursor off.

' Print the numbers 0 to 99 in big numbers in the middle of
' the screen. Pause a half second between numbers.
for b2 = 0 to 99
  serout 0,n2400,(posCmd,70,bigNums,#b2)
  pause 500
next b2
END                                 ' Finished.
```

```
' Program: 420DEMO.BS2 (Demonstrate 4x20 Serial LCD with Stamp II)
' This program demonstrates many of the features of the 4x20 Serial
' LCD Module from Scott Edwards Electronics. Set the module to
' 9600 bps (S1: switch 1 down; 2 up). Connect the serial input
' of the module to BS2 pin P0, GND to GND and +5V to +5V.
' Run this program.

' =========Define names for LCD instructions, bps=======
noCurs    con     4                 ' Make cursor invisible.
ulCurs    con     5                 ' Show underline cursor.
clrLCD    con     12                ' Clear entire LCD screen.
posCmd    con     16                ' Position cursor.
clrCol    con     17                ' Clear column.
wedge     con     128               ' Wedge-shaped symbol.
bigNums   con     2                 ' Begin big numbers.
N9600     con     $4054             ' Baudmode for inverted, 9600-bps output.
' The constants "bell" and "bksp" are predefined in PBASIC2.

' =================Begin demonstration=================
pause 1000
serout 0,N9600,[clrLCD]             ' Clear the screen.
for b2 = 0 to 79
  serout 0,N9600,[wedge]            ' Fill screen with wedges.
next
pause 1000                          ' Wait 1 second.
serout 0,N9600,[posCmd,68]          ' Move to position 4 (64+4= 68).
for b2 = 1 to 12                    ' Clear a 12-character swath with
  pause 350                         ' ...the clear-column feature.
  serout 0,N9600,[clrCol,bell]      ' Ring bell (if available) each loop.
next

' Turn on the underline cursor, move to pos. 26, and show message.
serout 0,N9600,[ulCurs,posCmd,90,"4x20 LCD"]
pause 1000                          ' Wait 1 second.
for b2 = 1 to 8
  pause 100                         ' Backspace to erase message.
  serout 0,N9600,[bksp,bell]        ' Ring bell (if available) each loop.
next
pause 1000                          ' Wait 1 second.
serout 0,N9600,[noCurs]             ' Turn cursor off.

' Print the numbers 0 to 99 in big numbers in the middle of
' the screen. Pause a half second between numbers.
for b2 = 0 to 99
  serout 0,N9600,[posCmd,70,bigNums,dec b2]
  pause 500
next
END                                 ' Finished.
```



```
' Program: Q420DEMO.BAS (QBASIC/QuickBASIC Demo of 4x20 Serial LCD)
' This program demonstrates the 4x20 Serial LCD Module from Scott
' Edwards Electronics with QBASIC or QuickBASIC. Connect the serial
' input of the LCD module to the com port transmit-data pin (pin 3
' of DB9; pin 2 of DB25). Connect GND of module to com port signal
' GND (pin 5 of DB9; pin 7 of DB25). Configure LCD module for 9600 bps
' (S1: switch 1 down; 2 up). Connect +5V power to LCD module : +5V to
' +5V and GND to GND. Run demo.
DECLARE SUB pause (time AS INTEGER)
CONST clrLCD = 12       ' Clear the LCD screen.
CONST noCurs = 4        ' Make cursor invisible.
CONST ulCurs = 5        ' Turn on underline cursor.
CONST bell = 7          ' Beep piezo buzzer (if attached).
CONST bksp = 8          ' Backspace.
CONST bigNums = 2       ' Begin big-number mode.
CONST clrCol = 17       ' Clear column.
CONST wedge = 128       ' Wedge-shaped symbol.
CONST posCmd = 16       ' Position the cursor.
OPEN "com1:9600,N,8,1,CD0,CS0,DS0,OP0" FOR OUTPUT AS #1
PRINT #1, CHR$(clrLCD);         ' Clear LCD.
FOR i = 0 TO 79
  PRINT #1, CHR$(wedge);        ' Fill the screen with wedges.
NEXT
PRINT #1, CHR$(posCmd); CHR$(68);       ' Move cursor to pos. 4 (64+4).
FOR i = 1 TO 12                         ' Clear a 12-column section of
  PRINT #1, CHR$(clrCol); CHR$(bell);   ' ..screen. Ring bell w/each loop.
  pause (350)
NEXT
PRINT #1, CHR$(posCmd); CHR$(90); CHR$(ulCurs);
PRINT #1, "4x20 LCD";            ' Print message on LCD.
SLEEP 1                          ' Wait a second.
FOR i = 1 TO 8
  PRINT #1, CHR$(bksp); CHR$(bell);     ' Backspace to erase message.
  pause (150)                           ' Ring bell w/each loop.
NEXT
SLEEP 1                                 ' Wait a second.
PRINT #1, CHR$(noCurs);                 ' Make cursor invisible.

FOR i = 0 TO 99                         ' Show #s 0-99 as big numbers.
  theNum$ = LTRIM$(RTRIM$(STR$(i)))     ' Trim spaces from number string.
  PRINT #1, CHR$(posCmd); CHR$(70); CHR$(bigNums); theNum$;
  pause (500)                           ' Wait 1/2 second.
NEXT
END
SUB pause (time AS INTEGER)
FOR i = 1 TO time
  PRINT #1, CHR$(0);
NEXT
END SUB
```





Data Vision

Optrex

All dimensions in mm
Check mark indicates model in this package
Dimensions provided by manufacturers—subject to change without notice