Chapter 5 Programming

Programming

Before programming, follow the instructions in *Chapter 2, Setup*, and the installation procedures in *Chapter 3, Connecting the LL 500*.

You can program your unit by scanning the bar code menus in *Chapter 6, Parameter Menus*. For alternate means of programming, such as PC Download, Advanced Data Formatting, and Cloning, consult the *Advanced Programmer's Guide*.

Regardless of the means you use, remember that not all of the available parameters apply to your specific host. For example, if you have an OCIA terminal, RS-232C parameters such as baud rate and parity do not apply. Ignore those parameters not designed for your application. If you're not sure which parameters apply, refer to *Parameter Selections* beginning on page 5-3 for your specific terminal type.

Scanning Sequence

If the default values suit your requirements, all you need to do is scan the **SET ALL DEFAULTS** bar code on page 6-5. Parameters other than default values can be set by scanning appropriate bar codes. *Chapter 6, Parameter Menus* contains all the bar codes necessary to program the controller for each parameter selection.

Scanning Sequence Examples

If you have an RS-232C terminal, for example, first refer to *Parameter Selections* beginning on page 5-3 to see what parameters apply to you.

To set a baud rate of 4800, refer to Table 6-1 beginning on page 6-1. Notice that the default baud rate is 9600. *Single or Dual Port RS-232C* parameters begin on page 6-84. Scan the **BAUD RATE 4800** bar code on page 6-86.

The controller issues a warble tone, signifying a successful parameter entry, then transmits data to your terminal at a 4800 bps baud rate. Most parameters can be set by scanning a single bar code.

If you want to add or change prefixes and suffixes or customize the data transmission format, you have to scan several bar codes. See the individual parameters for instructions.

Errors While Scanning

If you make an error during a scanning sequence, just reenter the correct parameter.

Code Types

The LL 500 can decode the following code types:

- UPC-E, UPC-A, EAN-8, EAN-13
- Code 39
- Interleaved 2 of 5
- Codabar
- Code 39 Full ASCII
- Code 128

The controller autodiscriminates between all code types, except for Code 39 and Code 39 Full ASCII. You may also choose to enable all code types, or enable common code types. When common code types (Code 39, UPC/EAN, I 2 of 5 and Code 128) are enabled, all other code types are disabled.

Parameter Selections

Not all parameters apply to all host types. *Table 5-1* alphabetically lists all standard parameters and applicable hosts.

Table 5-1. Parameter Selections

Decode Parameters	**RS-232C**	**4683***	**OCIA**	**OCR**	**Wedge**
Add Codes to be Decoded	X	X			X
ASCII Format	X				X
Baud Rate	X				X
Beep on BEL	X				X
Beep After Good Decode	X	X	X	X	X
Beeper Volume	X	X	X	X	X
Check Parity	X				X
CLSI Editing	X				X
Codabar Any Length	X				X
Codabar Length W/I Range	X				X
Codabar One Length	X				X
Codabar Two Lengths	X				X
Code 39 Buffering	X				X
Code 39 Check Digit	X	X			X
Code 39 Any Length	X	X	X	X	X
Code 39 Length W/I Range	X	X	X	X	X
Code 39 One Length	X	X	X	X	X
Code 39 Two Lengths	X	X	X	X	X
Code 128 Any Length	X	X	X	X	X
Convert UPC-E to A	X		X	X	X
*Some parameters programmed through store's computer					

Table 5-1. Parameter Selections (cont'd)

Decode Parameters	RS-232C	4683*	OCIA	OCR	Wedge
Data Transmission Format	X				X
Decode Redundancy	X				X
EAN Zero Extend	X		X	X	X
Hardware Handshaking	X				
Host Interface Code	X	X	X	X	X
Host Laser Off Delay	X	X	X	X	X
IBM 4683/4 Magstripe Comm.		X			
Ignore Unknown Characters					X
Intercharacter Delay	X				X
I 2 of 5 Any Length	X	X	X	X	X
I 2 of 5 Length Within Range	X	X	X	X	X
I 2 of 5 One Length	X	X	X	X	X
I 2 of 5 Two Lengths	X	X	X	X	X
Magstripe Prefix	X	X			X
Magstripe Suffix	X	X			X
NCR 2152 Fast Transmit					X
NOTIS Editing	X				X
Number of Stop Bits	X				X
OCIA Clock Polarity			X		
OCIA Transmit Time-out			X		
Parity	X				
Pause Duration	X				X
Predecode Block	X	X	X	X	X
Primary Laser Off Time-out	X	X	X	X	X
Primary Laser On Time-out	X	X	X	X	X
*Some parameters programmed through store's computer					

Table 5-1. Parameter Selections (cont'd)

Decode Parameters	**RS-232C**	**4683***	**OCIA**	**OCR**	**Wedge**
Primary Trigger Mode	X	X	X	X	X
RTS Line State	X				
Scan Prefix	X	X			X
Scan Suffix	X	X			X
Secondary Laser Off Time-out	X	X	X	X	X
Secondary Laser On Time-out	X	X	X	X	X
Secondary Trigger Mode	X	X	X	X	X
Serial Prefix	X				X
Serial Suffix	X				X
Serial Response Time-out	X	X			X
Software Handshaking	X				
Transmission Direction**	X				
Transmit Code ID Character	X				X
Transmit No Decode Message	X				X
Transmit UPC-A Check Digit	X				X
Transmit UPC-E Check Digit	X				X
UPC/EAN Supplemental	X				X
UPC-A Preamble	X				X
UPC-E Preamble	X				X
UPC/EAN Security Level	X	X	X	X	X

*Some parameters programmed through store's computer
**RS-232C Dual Port Only

Beeper Indications

Table 5-2 lists the beeper tones used by the LL 500.

Table 5-2. Beeper Indications

Beeper Sequence	Indication
Standard Use	
1 Beep - short high tone	A bar code symbol is decoded successfully (if decode beeper is enabled).
3 Beeps - long high tone	Power-on or reset. Occurs immediately after the unit is turned on or after a system reset, indicating the system software is working properly. If three beeps occur during normal operation, it is due to a reset; any work in progress is lost. If this occurs often, contact the Symbol Support Center.
4 Beeps - long low tone	A format or transmission error is detected in a scanned symbol. The data is ignored. This occurs if a unit is not properly configured. Check option settings.
5 Beeps - long high tone	Data conversion error.
Parameter Menu Scanning	
1 Beep- short high tone	Correct entry scanned or correct menu sequence performed.
1 Beep- lo/hi tone	Input error, incorrect bar code or CANCEL scanned, wrong entry, incorrect bar code programming sequence; remain in program mode.
1 Beep- hi/lo tone	Keyboard parameter selected. Enter value using bar code keypad.
1 Beep- hi/lo/hi/lo tone	Successful program exit with change in the parameter setting.

Table 5-2. Beeper Indications (cont'd)

Beeper Sequence	Indication
Serial Communications	
1 Beep - short high tone	Serial communication was successfully received, then transmitted to the host.
Code 39 Buffering	
1 Beep-hi/lo tone	New Code 39 data was entered into the buffer.
3 Beeps - long high tone	Code 39 buffer is full.
1 Beep - lo/hi/lo tone	The buffer is erased, or there is an attempt to transmit an empty buffer. When the Code 39 buffer was empty, the scanner read a command to clear or to transmit a Code 39 buffer.
2 Beeps - long high tone	Error in data transmission.
1 Beep - lo/hi tone	A successful transmission of buffered data.

RS-232C Communications Protocols

Host Communication

Host RS-232C communications may be operated in single port or dual port mode. In single port mode, the controller talks to the host only, and the aux port does not perform aux port communications.

In dual port mode, the host and aux ports are used. Data transmission may be directed to the host port only, or to both the host and aux ports. In this mode, the controller is transparent to the host and auxiliary device.

Hardware Handshaking

Hardware Handshaking allows you to check the readiness of the host before transmitting data. If the host is periodically occupied with other tasks, Hardware Handshaking is needed to prevent loss of transmitted data. Refer to the Hardware Handshaking diagram, *Figure 5-1*.

The data interface consists of an RS-232C port. This port is designed to operate either with or without hardware handshaking lines Request to Send (RTS) and Clear to Send (CTS).

If RTS / CTS handshaking is selected, scan data is transmitted with the following sequence (note that the DTR signal is hard wired active):

1. The controller reads the CTS line for activity. If CTS is asserted, the controller waits up to two seconds for the host to negate the CTS line. If after two seconds (default) the CTS line is still asserted, the controller sounds a transmit error (if equipped with optional beeper) and any scanned data is discarded.
2. When the CTS line is negated, the controller asserts the RTS line and waits up to two seconds for the host to assert CTS. When the host asserts CTS, data is transmitted. If, after two seconds (default) the CTS line is still not asserted, the controller sounds a transmit error (if equipped with optional beeper) and any scanned data is discarded.
3. When data transmission is complete, the controller negates RTS 10 msec after sending the last character.

4. The host should respond by negating CTS. The controller checks for a negated CTS upon the next transmission of data.

During the transmission of data, the CTS line should be asserted. If CTS is deasserted for more than 50 msec between characters, the transmission is aborted, the controller sounds a transmission error, and the data is discarded.

If the above communications sequence fails, the controller issues a transmit error. In this case, the data is lost and must be rescanned.

If no Hardware Handshaking is selected, data is transmitted based on the Software Handshaking options which follow.

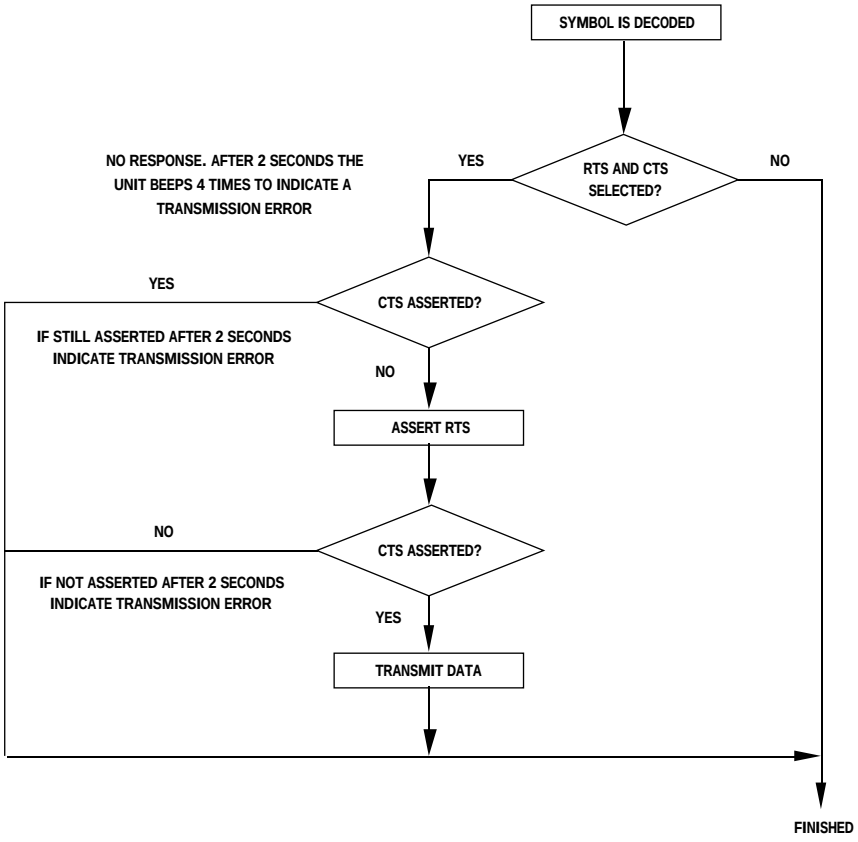

Figure 5-1. Hardware Handshaking

Software Handshaking

Software Handshaking offers control of the data transmission process as an alternative to that offered by hardware handshaking. There are five Software Handshaking options: None, ENQ, ACK/NAK, ACK/NAK with ENQ, and XON/XOFF. See *Figure 5-2*.

None

No handshaking is selected and data is transmitted immediately.

ACK/NAK Only

The ACK/NAK option checks the success or failure of a transmission. The controller expects one of the following host responses after a data transmission:

<ACK> - Acknowledges a valid and successful transmission.

<NAK> - Indicates a problem with the transmission.

When a <NAK> is received, the controller retransmits the same data and awaits an ACK/NAK response. After three unsuccessful attempts to transmit the data, an error indication sounds (four long beeps) and the data is discarded.

The controller waits up to the programmable Host Serial Response Time-out to receive an ACK or NAK. If the controller does not get a response in this time, it issues an error indication and discards the data. There are no retries when a time-out occurs.

ENQ Only

When this option is selected, the controller waits for an ENQ (Enquire) character from the host before it transmits data. This ensures that data transmission occurs only when the host is ready to receive. With ENQ enabled, the controller must receive an ENQ from the host within two seconds after the last activity or the controller sounds four long beeps to indicate a transmission error and discards the data; the unit is now ready to scan again.

ACK/NAK with ENQ

This combines the two previous options.

XON/XOFF

- When XON / XOFF is enabled, the controller assumes an XON for the first transmission, and proceeds to transmit scanned data until an XOFF is received. Transmission continues when an XON is received. There are two situations for XON / XOFF:
  - XOFF is received before the controller has data to send. When the controller has data to send, it then waits for an XON character before transmission. The controller waits up to two seconds to receive the XON. If the XON is not received within this time, the controller issues an error indication and discards the data.
  - XOFF is received during a transmission. Data transmission then stops after sending the current byte. When the controller receives an XON character, it sends the rest of the data message. The controller waits indefinitely for the XON.

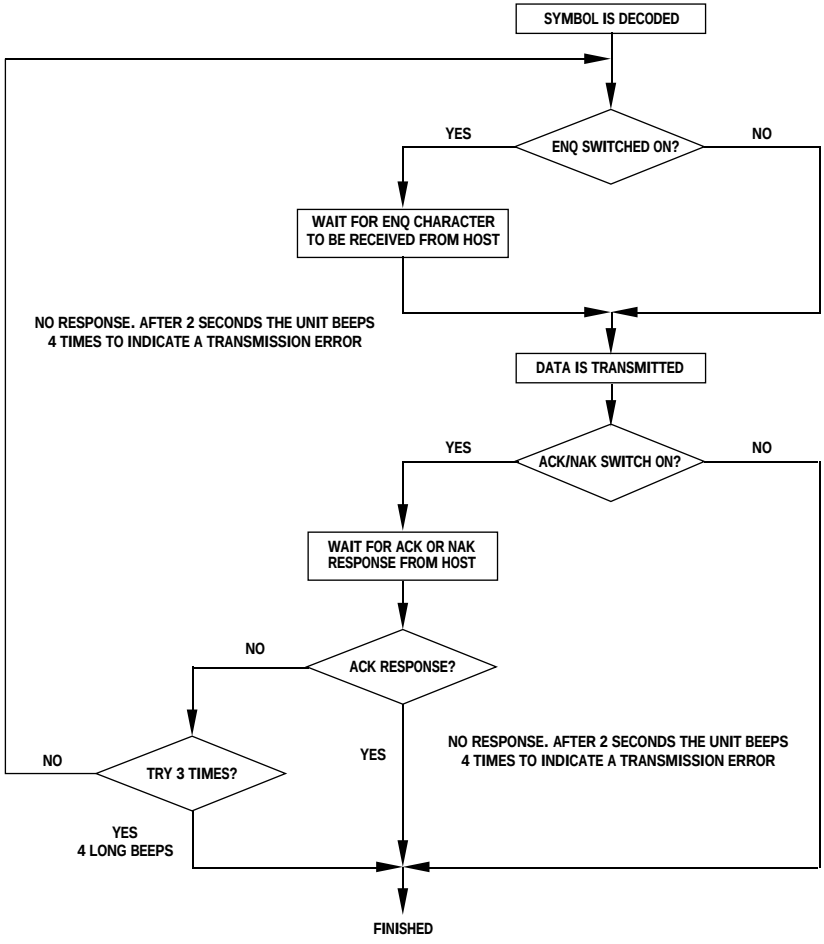

Figure 5-2. Software Handshaking

Auxiliary Port Communications

The aux port acts primarily as a receiver. When data is received from a serial output device and the message is error free, the controller sounds a decode beep and the message is transmitted to the appropriate host.

Hardware Handshaking

If RTS / CTS is asserted, the controller checks to see if CTS is asserted. If it is, the controller asserts RTS and keeps it asserted until the controller's data buffer reaches 100 characters.

The serial output device must respond to the controller's deasserted RTS line by halting transmission of new serial data and waiting for the controller to assert RTS before resuming serial communications.

If any parity errors are detected in the data buffer, the controller does not transmit to the host and sounds 4 long beeps.

Software Handshaking

When XON / XOFF is selected, the serial output device connected to the aux port assumes an XON for the first data transmission, and may transmit at any time after receiving an XON.

The device transmits until receiving an XOFF from the controller. The controller transmits an XOFF when its data buffer reaches 100 characters, or it detects no serial activity for a time duration set to the serial response time-out.

Data received to this point is transmitted to the host. When the controller detects a parity error, it sounds a transmit error beep, does not transmit an XOFF, and continues with serial communications as described above.

Dual Port RS-232C Only

Table 5-3 shows the options for a typical dual port RS-232C configuration, assuming that the host computer is connected to the 40-pin Host port on the LL 500, and the local terminal is connected to the Aux port. The first two columns show how the controller's transmission direction is related to the communication mode of the host system. Columns 3 and 4 show how software handshaking further determines transmission direction.

Table 5-3. Dual Port Transmission

Communication Mode Between Host & Local Terminal	Transmission Direction	Software Handshaking	Transmission Direction Selection
Full Duplex	Out 40-pin Connector Only	With Host Computer	Talk & Receive With Host Computer Only
Half Duplex	Out 40-pin Connector and Aux Modular Connector	With Host Computer	Talk to Host Computer & Local Terminal. Receive from Host Computer Only
Block Mode	Out 40-pin Connector Only	With Host Computer	Talk & Receive With Local Terminal Only

Figure 5-3 on page 5-16 shows how the host and local terminals are connected to the LL 500 for full and half duplex modes. Figure 5-4 on page 5-16 shows how they are connected for block mode.

Software and hardware handshaking are only accomplished through the Host port in dual port RS-232C mode.

Transmission can only be set in dual port RS-232C mode.

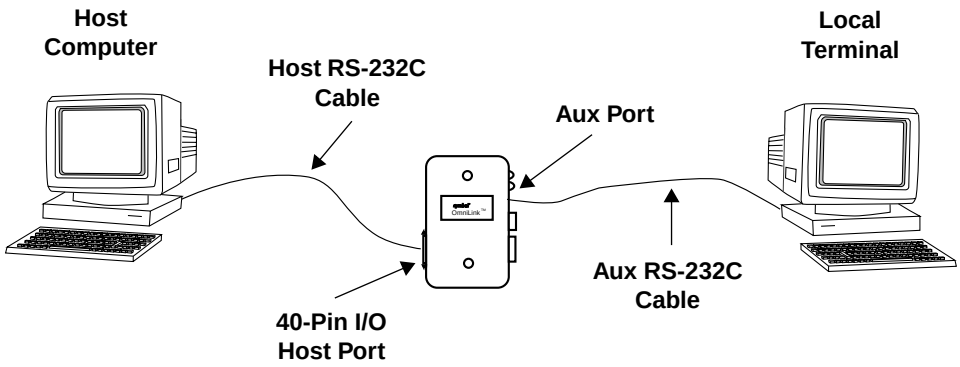

Figure 5-3. Full & Half Duplex Mode Between Host and Local

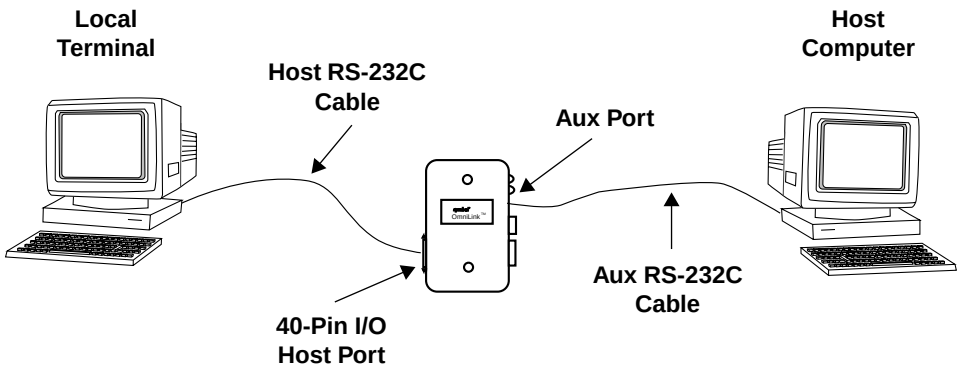

Figure 5-4. Block Mode