LCD–LINK OPTION

Terminal Interface for COMM–LINK2 & HCS II

Rev. 1.0

****** NOTICE *******

It is very important to understand the
function of R15 ** BEFORE ** you install it.
Please read the entire manual
before using the board.

******* CAUTION *******

PLEASE READ THE ENTIRE MANUAL ** BEFORE ** ATTEMPTING ASSEMBLY !!
Understand the significance of each component and jumper setting
' BEFORE ' assembly and applying power. CHECK AND RECHECK !
FAILURE TO DO SO WILL PUT YOUR WARRANTY AT RISK !!

CIRCUIT CELLAR INC.

4 Park Street ● Vernon, CT 06066

TECHNICAL MANUAL

LCD-LINK OPTION

Terminal Interface
for COMM-LINK2 & HCSII

Technical Manual

Release 1.0
4/6/92

Circuit Cellar Inc.
4 Park St.
Vernon, CT 06066

********** COPYRIGHT **********

********** Disclaimer **********

While we have attempted to provide accurate and up to date information in this manual, CIRCUIT CELLAR INC. makes no representations or warranties respecting its contents. We reserve the right to make periodic changes to the text and to issue new editions of this manual without notification.

Occasionally in this manual we refer to the products of manufacturers other than ourselves. Such references do not constitute an endorsement of these products, but are included for the purpose of illustration or clarification. We do not intend such technical information and interface data to supersede information provided by the individual manufacturers.

****

Conditions of Sale and Product Warranty

CIRCUIT CELLAR, Inc. (CCI) and the Buyer agree to the following terms and conditions of Sale and Purchase:

1. CCI extends the following warranty: a factory manufactured circuit board or assembly carries with it a one-year warranty covering both parts and labor. Any unit which is found to have a defect in materials or workmanship will, at the option of CCI, be repaired or replaced.

2. A minimum inspection fee must be prepaid for the repair of units that are no longer under warranty. Contact CCI for information on current minimum charges.

3. NO WARRANTY is extended on USER ASSEMBLED systems or kits. However, assembled kits will be inspected and repaired with charges based on the current minimum one hour charge. CCI retains the right to refuse to repair any USER ASSEMBLED item. This right is at the sole discretion of CCI. However, in the event that the repair charge would exceed a reasonable amount, the user may be consulted for a determination. Repairs on user assembled items must be prepaid. Return authorization must be obtained prior to any return.

4. CCI will not be responsible for the repair or replacement of any unit damaged by user modification, negligence, abuse and/or mishandling, or improper installation.

5. CCI is not responsible to the Buyer for any loss or claim of special or consequential damages.

6. All units returned for repair must first receive prior authorization from CCI. A return authorization number may be obtained by phone or letter. Please retain a record of this number, since most subsequent correspondence will refer to this authorization. Under no circumstances should any product be returned to CCI without prior authorization. CCI will assume no responsibility for unauthorized returns. Returns must be shipped prepaid. Insurance is recommended as losses by a shipping carrier are not the responsibility of CCI. Repaired units will be returned postage and insurance paid.

7. CCI reserves the right to alter any feature or specification at any time. This right extends to fees, charges, and any other conditions or warranties contained herein.

Table of Contents

MICROCONTROLLER REVOLUTION

'Expandable', the catch phrase of the 80's, is taking on new meaning in the 90's. The ability to add functions to a small microcontroller through an expansion bus has been a hot selling 'feature' for many companies. Today's cost-conscious product developer may not be willing to pay for that flexibility. Their product is well defined and must be minimal in cost. Expansion capabilities require additional parts, real estate, and power that is wasted in many cases where expansion will not be used.

What many need is an 8031/8052 controller in a plain brown wrapper, that is, optimized for these minimal-configuration applications.

The COMM-LINK2 board meets this criteria. Optimized both for single-use drop-in solutions as well as volume OEM applications, this new microcontroller is designed to be small and cost effective.

Measuring only 3.5 by 5.2 inches, the COMM-LINK2 board contains an 8031 (or 80C52) processor, EPROM and RAM memory, 12-bits of parallel I/O, and an RS/232-RS/485 serial port. A single 5 VDC regulated supply is necessary for operation or this can be obtained from a 9-12 VDC unregulated input by the on-board linear regulator.

8031 FAMILY PIN DESCRIPTION

Port 0	pins 39-32	8-bit open drain bidirectional I/O (multiplexed low-order data/address for external memory)
Port 1	pins 1-8	8-bit quasi-bidirectional I/O
Port 2	pins 21-28	8-bit quasi-bidirectional I/O (high-order address for ext. memory)
Port 3	pins 10-17	8-bit quasi-bidirectional I/O (secondary functions as follows: **RXD/data** Serial Channel's receiver **TXD/data** Serial Channel's transmitter ***INT0** Interrupt 0/counter gate 0 input ***INT1** Interrupt 1/counter gate 1 input **T0** Counter 0 input **T1** Counter 1 input ***WR** write for external data memory ***RD** read for external data memory
ALE	pin 30	Address latch enable
***PSEN**	pin 29	read for external program memory
***EA**	pin 31	tied to logic high for executing code masked within the 8x5x series processors with internal ROM or tied to logic low to disable internal ROM and fetches all instructions from external program memory

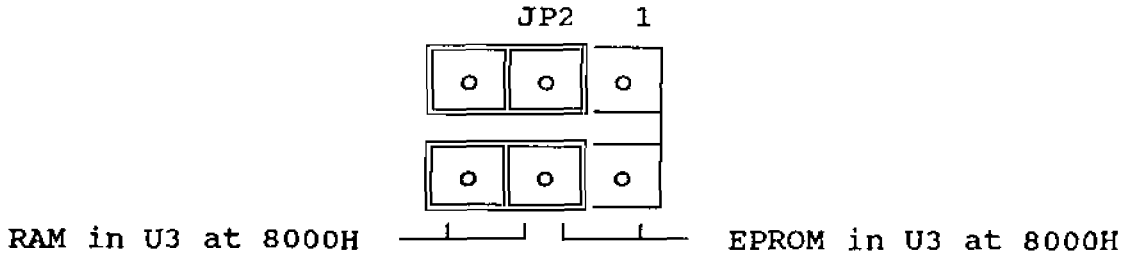
EXTERNAL ADDRESSING SPACE

The COMM-LINK2 microcontroller can directly address 64K of external memory. That is, overlapped DATA/CODE memory, for an "anything goes here" space.

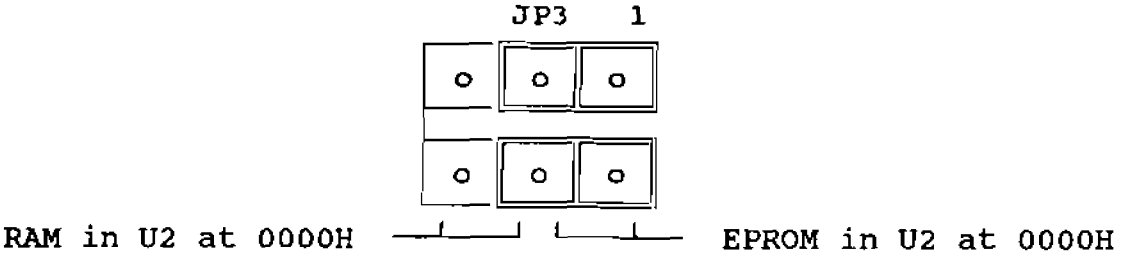
Combinations 32K RAMs and EPROMs are allowed on the COMM-LINK2 board. **When using the 8031, your EPROM code starts execution at 0000H. When using the 80C52 (masked with BASIC in the internal ROM), RAM must be at 0000H.**

DEVICE TYPE SELECTION

Two memory sockets are provided on the COMM-LINK2 board (U2 & U3). Each memory socket has a jumper used in selecting the type of address space assigned to the socket. JP1 selects the device type for U2 and JP2 selects the device type for U3. The selection made determines where the control lines are connected on the memory socket. RAM and EPROM devices require slightly different configurations. The following is an example of memory configuration when using an 8031.



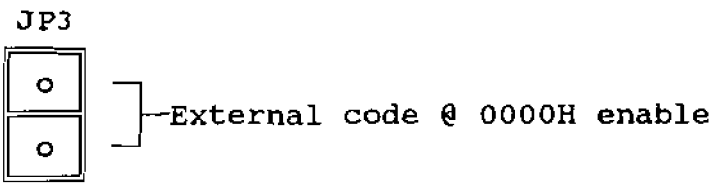
JP2 shows RAM in U3 starting at address 8000H



JP1 shows EPROM in U2 starting at address 0000H

EXTERNAL CODE SELECTION

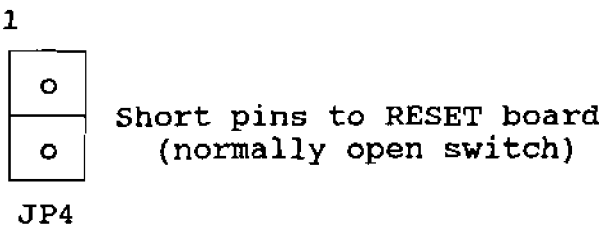
The 8031 microcontroller requires *EA (pin 31 on the microcontroller) to be pulled down to a logic low level. This instructs the processor to start executing machine language code starting at address 0000H. The 8052 is masked with BASIC and will run internal code (BASIC interpreter) if *EA is pulled up to a logic high. Pulling *EA low on an 8052 will disable BASIC and execute code starting at 0000H (similar to an 8032).



JP3 shows the microcontroller enabled for external code execution

RESETTING THE COMM-LINK2

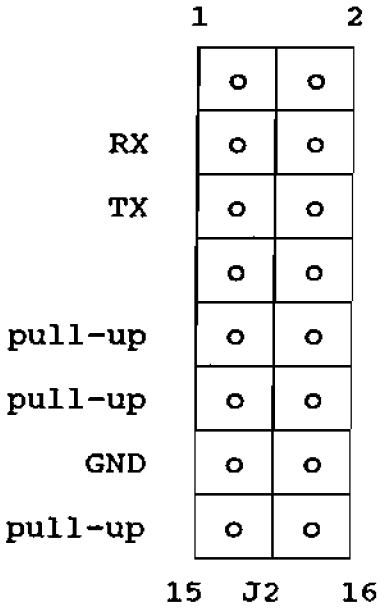
Reset of the COMM-LINK2 board occurs when a normally open push-button switch is attached to JP4 and momentarily pressed. A logic high is applied to the system RESET line and held momentarily high by an R/C circuit.

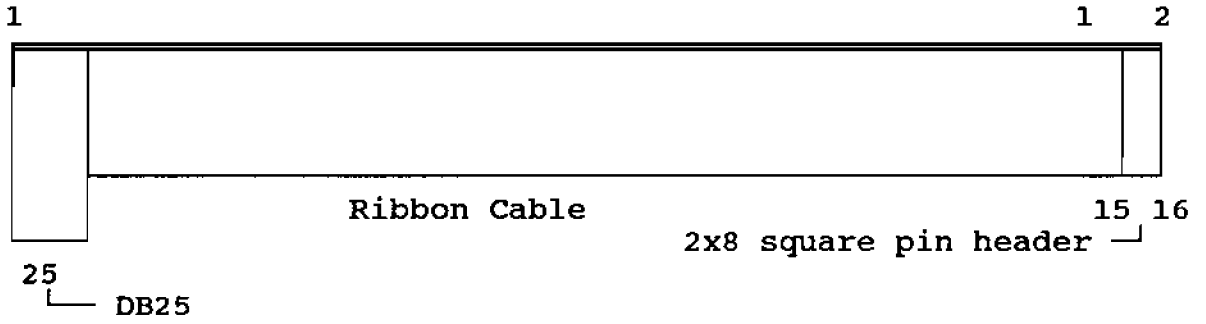


Use JP4 for connecting a normally open push-button switch as an external system RESET

RS-232 COMMUNICATIONS

The COMM-LINK2 contains a full-duplex serial channel. TTL-level serial signals are converted to ±10-volt RS-232-compatible signals by U7, the MAX232 device. A 16-lead flat ribbon cable made with a DB-25 at one end (for connection to DTE terminal device) and a 16-pin plug for connection to JP2) will permit serial communication with a dumb terminal or computer.



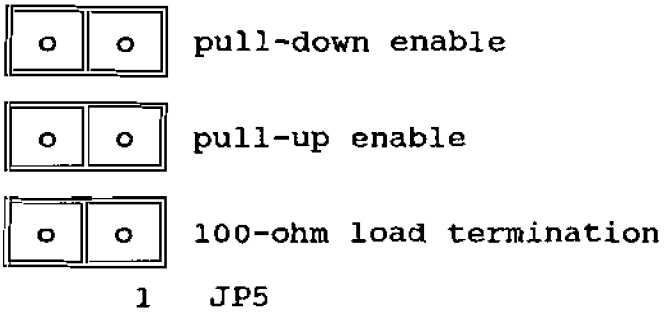


Cable required for RS-232 communications

To eliminate unwanted noise on the RX input to the processor, remove the unused line driver chip. (U7-MAX232 or U8-75176)

RS-485 COMMUNICATIONS

RS-485 communications over a single twisted pair can include multiple (up to 32) devices. Since each device can transmit and receive, certain protocols must be adhered to to prevent message collision. The simplest being "listen to the line and transmit only if free". (The protocol you use will depend on the application and is beyond the scope of this manual.) JP5 enables termination resistors on the twisted pair and should be installed only on the microcontrollers located at the extreme ends of the twisted pair (one at each end). U8, the 75176 RS-485 device, draws as much current as the rest of the COMM-LINK2 board. If you don't require RS485 communications removing U8 will reduce overall power requirements.



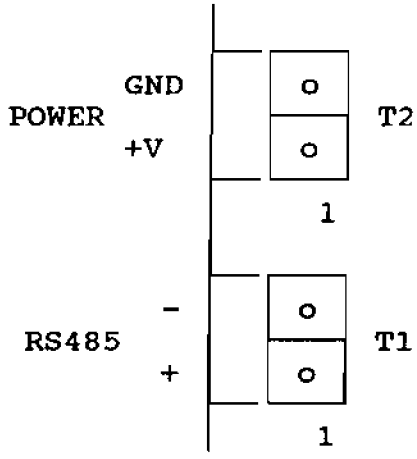
JP5 shows termination of the RS-485 lines enabled

The 100 ohm termination which is across the RS485 twisted pair may be needed on the node at the end of the twisted pair daisy chain. The pull-up and pull-down resistors may be required to assure that the twisted pair is coaxed to the idle state when no RS485 driver is active. It is suggested that any termination be done on the last node for less confusion.

RS485/POWER CONNECTIONS

Four screw-terminal connections are used for connecting power and the RS485 network to the COMM-LINK2 board. This allows the COMM-LINK2 board to operate remotely with only a four wire cable (i.e. telephone cable). Of course you could use the COMM-LINK2 board as a stand-alone controller with RS232 as the communications medium.

T1 is a two position screw terminal block used for RS-485 twisted-pair communications. T2 is used as the power supply input, requiring either 9-12 volts of unregulated DC from a remote supply on a second pair of wires or a local 5 VDC regulated supply (with the zero ohm jumper at R15 installed).



Screw terminal blocks for Power and RS485 Communication

POWER for the COMM-LINK2

Use T2 for attaching the power supply connections to the COMM-LINK2 board. The input can be any unregulated DC voltage from 9-12 volts. A series diode protects the COMM-LINK2 board from accidental voltage reversals. The unregulated DC powers a linear voltage regulator on the COMM-LINK2 board which provides the nominal 5.0 volts.

Alternately, a regulated 5.0 volts can be used as input. When the input is a regulated 5.0 volts, the user must install a zero ohm jumper at R15. This bypasses the linear regulator and series diode. Caution: Voltage reversal protection is eliminated when a jumper is placed at location R15.

To eliminate unwanted noise on the RX input to the processor, remove the unused line driver chip. (U7 or U8)

COMM-LINK, THE BASIC NETWORK NODE FOR THE HCS II

The HCS II is an expandable, network-based, intelligent-node, industrial-oriented supervisory control system which, in its minimal configuration, performs quite suitably as a home control system. HCS II incorporates direct digital inputs and outputs, direct analog inputs and outputs, real time or boolean decision event triggering, X-10 transmission or reception, infrared remote control transmission and reception, remote displays as well as a master console, and it has the capability to perform as a complete badge monitoring and personnel tracking system.

The HCS II system architecture consists of a central supervisory controller connected to up to 31 other functional modules (called links) via an RS-485 serial network. The system controller and the links can operate independently and do not need the system controller or other links to function. This allows easy testing or incorporation as intelligent subsystems in other control equipment. The subsystem links all share a common 8031 controller board (generically called a COMM-LINK) with the I/O customized for each application.

The supervisory controller is quite literally the brains of the system. While every Link has its own processor, they each rely upon the supervisory controller for timing, control commands, and overall system coordination.

The supervisory controller is a 9.218 MHz 64180-based single board computer. It has two serial ports, up to 96K bytes of memory (32K bytes are battery-backed), an eight channel 8-bit or 10-bit A/D, a real time clock-calendar, and 24-bits of parallel I/O (an additional 48-bits of parallel I/O can be added if required). These parallel I/O bits can be further conditioned by externally connecting them to optoisolators, relays, or drivers. Depending upon your application you may find that the supervisory controller is all you need.

One of the serial ports performs as a dedicated 9600 bps serial RS-485 connection to the various network links. The wire from this port can be up to 4000 feet long (as opposed to 50 feet for RS-232). The second port connects (RS-232) to a PC/AT which functions as the master console. Programs running on the master console allow the user to enter and store control sequences which will be subsequently compiled and downloaded to the supervisory controller's non-volatile memory. The master console also acts as an on-line real time display of all control activity in the system. When these downloading or display functions are not required, the master console can be turned off to reduce system power consumption.

Besides the Supervisory Controller, the HCS II system consists of a multitude (none to 31) of specialized-function network nodes which facilitate remote data acquisition, closed-loop control, and display. With the exception of the ADIO-LINK, all comm-links consist of the same generic 8031 computer with only the external parallel I/O circuitry and operating software being different. The generic COMM-LINK board is configured with minimal components. It accommodates up to 64K bytes of either RAM and/or EPROM memory. The single serial port is configured to operate both RS-232 and RS-485. When connected to the Supervisory Controller in normal operation, the connection is via a twisted pair to the RS-485. Additional line balancing and termination resistors are included. According to the driver chip manufacturers, this line can be 4000 feet long. In normal operation only RS-485 is used and the MAX232 (U7) can be removed to reduce power.

The available I/O on the comm-link circuit is limited to the PORT1 bits of the processor, interrupt lines, and T0 and T1. When used as a powerline interface, these lines connect to the TW-523 X-10 module; when used as an infrared gateway, these lines connect to infrared LED driver logic and an IR receiver; when used as an LCD display, these lines supply character data to the LCD. In addition, to facilitate additional link designs or modifications to the existing circuits, the comm-link PC boards contain a prototyping area.

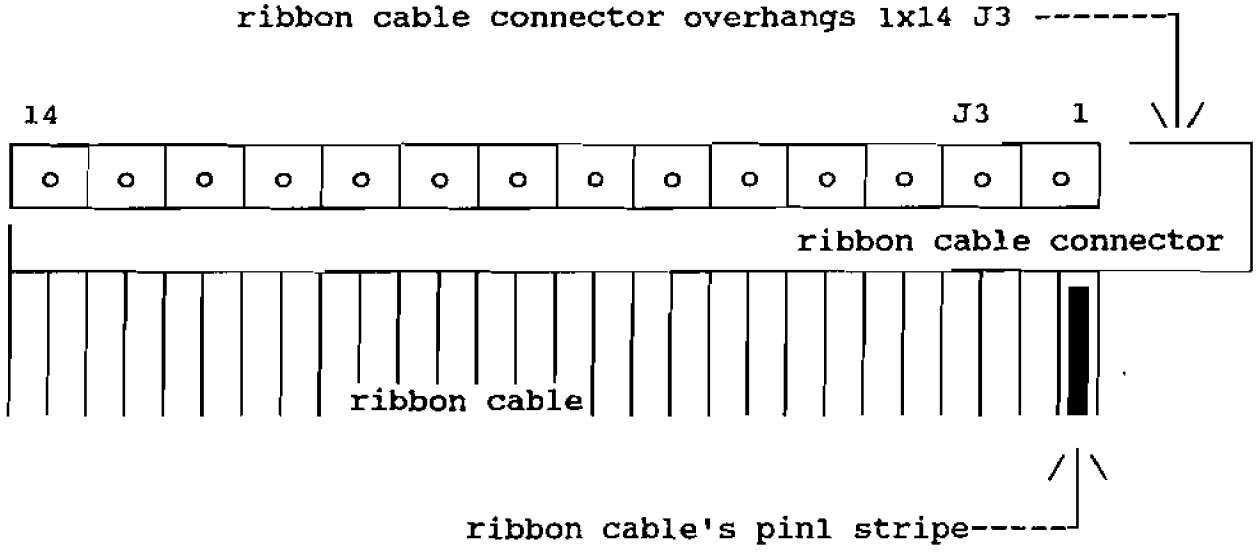
The RS-232 is reserved for local non-system direct testing or use (of course, you could also operate it stand-alone with RS-485). Each link has a unique command set and, for multiple units of the same type, a unique address. By connecting the link to the serial port of a terminal or any PC running a terminal emulation program, you can directly command the comm-link to do any activity which it would normally do for the Supervisory Controller.

With the PL-LINK alone, for example, you could directly control all X-10 on/off, dim/bright functions from a PC. In addition, because the PL-LINK is "smart" it offers the user the added capability to "listen" to the power line and record whether any other X-10 codes have been transmitted (either manually or automatically, from this or any other transmitter), and it will automatically "refresh" desired X-10 modules at a prescribed refresh rate. Intelligent links simplify testing and problem diagnosis. More often than not, the location where sensors are placed have no convenient AC power outlet and 5 volts is no longer 5 volts if run over any length of a twisted pair. To remedy this, the individual network links will operate on a 9-12 VDC input.

While the comm-link circuitry runs on 5 volts, the design incorporates a linear regulator so that the 5V can be derived from a wide-ranging DC input. The intention is to promote connecting the individual links via a 4-wire cable, rather than just a 2-wire twisted pair, when local power is unavailable. Two wires supply RS-485 while the other pair supplies +12V and ground (actually 9-12V is fine. Higher voltages require a larger heatsink on the regulator). Of course, using the regulator is optional, and the board can be operated (jumper selectable) on +5V.

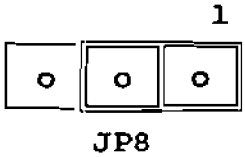
LCD CONNECTIONS

The LCD-LINK converses with a 4 line by 20 character LCD through J3. Your HCSLCD/4 comes complete with a 34-pin ribbon cable and should be connected to the LCD-LINK as shown here.



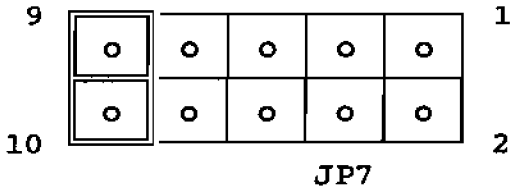
The LCD must be a Hitachi LM044L, a Vikay VK2420-ST, an Epson EA-D20040AR-S, or other compatible 4x20 character display. Use potentiometer P1 to adjust the contrast of the LCD. Start with the pot rotated fully clockwise, before powering up the board. After the power is applied, slowly turn P1 counterclockwise, until the display becomes visible.

A shorting jumper on JP8 pins 1 & 2 will enable the heartbeat LED and assure you that the software is running.



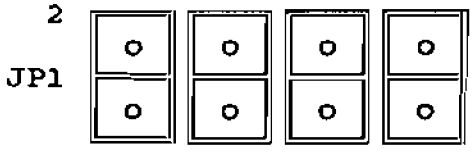
JP8 shown enabling the heartbeat LED

Interactive mode is enabled through JP7 pins 9 & 10. When a shorting jumper is placed on these pins (and the board is reset) the LCD-LINK enters the interactive mode. Remove this jumper for the LCD-LINK to be used on the HCSII network.

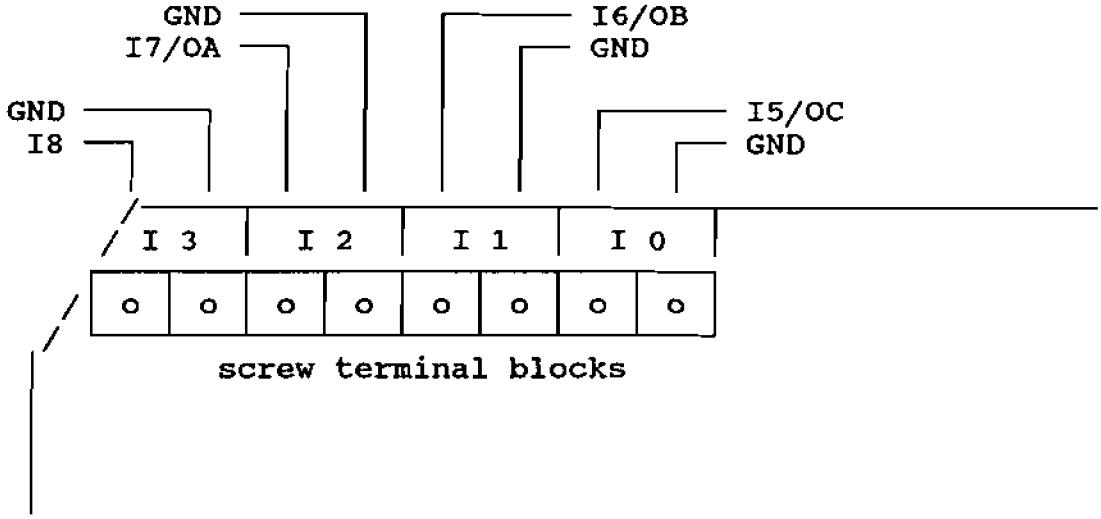


JP7 shown enabling the interactive mode

To use the screw terminals for inputs 5-8, place shorting jumpers on J1 pins 1 & 2, 3 & 4, 5 & 6, and 7 & 8. This will connect LED/input8 (P1.7) to screw terminal I3, input7/outputA (P1.6) to screw terminal I2, input6/outputB (P1.5) to screw terminal I1, and input5/outputC to screw terminal I0.



JP1 shown enabling the scew terminal blocks for I/O



USER CONNECTIONS TO SCREW TERMINAL I0-I3

LCD-LINK SOFTWARE

Setup your board as suggested for an 8031 based system, with the LCD-LINK EPROM in U2 and a RAM in U3. Upon power-up or reset, LED3 will blink about once a second. This gives you a bit of reassurance right away that the system is operating correctly.

If the LED is not blinking, here are some things to check:

1. Shorting jumper positions
2. RAM and EPROM positions
3. Power Supply connections
4. 5 volts on U1 pin 40 to ground (pin 20)
5. Reset U1 pin 9 low
6. ALE U1 pin 30 oscillating (about 2MHz)

Bit usage for the LCD-LINK Monitor

P1.7	LED/I8	heartbeat at 1 Hz and input bit8
P1.6	I7/OA	input bit7 or output A
P1.5	I6/OB	input bit6 or output B
P1.4	I5/OC	input bit5 or output C or BELL pulse
P1.3	LCD4	LCD data bus
P1.2	LCD3	LCD data bus
P1.1	LCD2	LCD data bus
P1.0	LCD1	LCD data bus
P3.3	INT1	LCD enable
P3.2	INT0	LCD R/-W select
P3.5	T1	LCD Data/-Register select
P3.4	T0	controls RS-485 driver (LED ON = disabled)

LCD connections:

1	Ground	
2	+5 VDC	
3	Display contrast (if necessary else, grounded)	
4	Data/-Register select	(P3.5)
5	R/-W select	(P3.2)
6	Enable	(P3.3)
11	LCD data bus	(P1.0)
12	LCD data bus	(P1.1)
13	LCD data bus	(P1.2)
14	LCD data bus	(P1.3)

GETTING STARTED - LCD-LINK MONITOR

For RS232 Use (N1-network mode default):

1. Set Communications to 9600 b/s 8N1.
2. Remove 75176 transceiver chip, install MAX-232.
3. Connect PC to RS-232 header as usual.
4. Verify commands to satisfy yourself that it works...
   Prefix all commands with "! TERM0 " (note spaces!)
   TERM0 is the default address, to change, use the A command
   There is no command echo in this mode

If you have trouble, insert shorting jumper JP7 pins 9 & 10 and reset the CPU to enter "interactive" mode (N0-interactive mode):
A Descriptive banner appears.
Do not use the prefix with the commands in the interactive mode.

For Network Use (N1-network mode default):

1. Set Communications is 9600 b/s 8N1.
2. Install 75176 tranceiver, remove MAX-232.
3. Connect PC through RS-485 converter.
   Your software must enable/disable RTS correctly!
   Make sure P1.4 (JP7 pins 9 & 10 have no jumper) is not grounded when you turn power ON.

CHECKSUMS

Network mode transmissions can be validated by using a checksumed transmission. The "!" prefix indicates a non-checksumed command, while a "#" prefix includes a checksum. The command and response syntax is as follows:

#xx addr command

where:	
#	prefix w/checksums
xx	checksum byte in hex
blank	ASCII 32 (20 HEX)
addr	nodes' network addr
blank	ASCII 32 (20 HEX)
command	command text
<CR>	ASCII 13 (0D HEX)

$xx addr response

where:	
$	prefix w/checksums
xx	checksum byte in hex
blank	ASCII 32 (20 HEX)
addr	nodes's network addr
blank	ASCII 32 (20 HEX)
response	response text
<CR>	ASCII 13 (0D HEX)

Compute the checksums as follows:

1. Assemble the command w/"00" (30 HEX 30 HEX) for the checksum
2. Add up all the characters in the command including checksum
3. Find the 2's complement of the total's lower byte
4. Convert this into ASCII characters and replace the checksum

Verify the checksum as follows:

1. Remove the command's checksum and replace w/"00"
2. Convert the checksum characters to a binary byte
3. Add up all the characters including the replaced checksum
4. Add the original checksum's binary byte
5. If the lower byte = zero then it verifies correctly

LCD-LINK COMMAND SET:

A=string Set network address to string (16 characters max)
D Dump program status (debugging use)
E Show and clear error flags (debugging use)
Ln Set logging mode (bit mapped)
L report current mode
L0 disable (default)
L1 show ANSI decoding sequence
L2 show LCD command processing

Nn Set network/interactive mode
N report current mode
N0 set interactive mode
N1 network mode (no error messages) (default)
N2 network mode with command echo and error msgs
N4 same as N1 with command echo and error msgs

Ox=n Set output bit x to state n (OA=1, OB=0)
Bits are normally high after reset
Q Query buttons pressed since last Q
Buttons are bit-mapped in HEX byte
Current hardware returns buttons 80, 40, 20, and 10 only
RESET perform power-on reset, must be completely spelled out
S=string Send string to LCD via ANSI decoder
(string continues to end of line)

NOTES

The Ox=n command controls output pins P1.4-6. The bits are normally high after a hardware or software reset. Issuing OA=0 sets P1.5 LOW, OA=1 sets it high. Because 8031 CPU bits have very weak pullups, you should probably use drivers with active low inputs.

Because pins P1.4-6 can be used as both button inputs and bit outputs, the firmware samples and stores the output state, sets the output high, reads the input value, and restores the output state. The sampling occurs 200 times per second and results in a narrow pulses on the output because low outputs glitch high and high outputs glitch low. Output devices that respond to narrow pulses may false-trigger on the glitches. If this is a problem, an RC low-pass filter before the output driver should eliminate the glitches.

Pin P1.4 has four functions. If the pin is grounded during RESET or power-on the firmware comes up in interactive mode and does not use the standard network headers. The pin is both a switch input (for the Q command) and an output pin (for the O command), and the firmware can distinguish the two as described above. Finally, P1.4 goes low for 500 ms when the firmware decodes the \b character sequence in an S command, as described below.

The S command interprets several C-style escape sequences to simplify entering "unprintable" characters that control LCD formatting. All sequences start with a backslash (ASCII 5C)...

```
\b      Bell (pulses P1.4 low for 0.5 sec)
\cn     Control character n (\cZ = Ctrl-Z)
\e      Escape character, ASCII 27
\f      Form feed character, ASCII 12
\n      New line (linefeed and carriage return)
\r Carriage return (to current line, leftmost column)
\t      Tab to next stop: column 4, 8, 12, 16, 20
\xnn    Send hex char nn directly to LCD panel Must have two digits
\\      Single backslash
```

A backslash followed by any other character simply displays that character, so the backslash vanishes.

The \x sequence is intended to display any character in the LCD controller's character set including those above 0x7F. The firmware displays a blank and then overwrites it with the \x character, so do not put a \x character in the last column of the last row. Although \x provides access to the LCD's programmable characters (\x00 through \x07), the firmware does not initialize those characters and they display as random "junk."

The \c sequence is a different way to generate control characters that might otherwise use \x. For example, \x01 and \cA are equivalent. Only the low-order five bits of the character are used, so \cA and \ca are equivalent.

The \b sequence simulates the BELL character by pulsing P1.4 low for about 500 ms. You may use this to control a piezo buzzer to attract attention to the display, but remember the very limited 8031 output drive capability. This is the only way to activate the output, as Ctrl-G (\x07 or \cG) is routed directly to the LCD panel. If P1.4 is also used as a switch input, the buzzer will sound whenever the switch is closed, but the firmware can tell the difference between a low output and a low input.

ANSI Control Sequences

All control sequences begin with two common characters:
ESC - ANSI 27, the Escape character (use \e)
[- ANSI 91, the left square bracket
There are no blanks within the command strings.

Numeric parameters are decimal and default to 1 if omitted.

Row and column numbers start with 1.

The upper/lower case of the trailing letter is important!

Command	Example	Function
ESC[#A	ESC[2A	Cursor up # rows (up 2)
ESC[#B	ESC[B	Cursor down # rows (down 1)
ESC[#C	ESC[10C	Cursor right # columns (right 10)
ESC[#D	ESC[5D	Cursor left # columns (left 5)
ESC[#;#H	ESC[H	Set cursor to row;column (to 1,1)
ESC[#;#f	ESC[1;2f	Set cursor to row;column (to 1,2)
ESC[#;#j	ESC[3j	Set cursor to row;column (to 3,1)
ESC[s	ESC[s	Save current cursor location (1 level)
ESC[u	ESC[u	Restore saved cursor location
ESC[2J	ESC[2J	Clear display and home cursor
ESC[K	ESC[K	Clear from cursor to end of row
ESC[6n	ESC[6n	Query current cursor location
ESC[#;#R	ESC[3;4R	Terminal's response to location query (cursor at row 3, col 4 = ESC[3;4R)
ESC[#h	ESC[7h	Set display mode (ESC[7h to wrap at end of rows)
ESC[#l	ESC[7l	Set display mode (that's an "ell") (ESC[7l to force cr/lf at end of row)
ESC[#;#m	ESC[3m	Set display attributes (ignored)
ESC[#;#p	ESC[3;4p	Reassign key code (ignored)