

IR-LINK

MCIR-LINK

InfraRed Interfaces for 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.

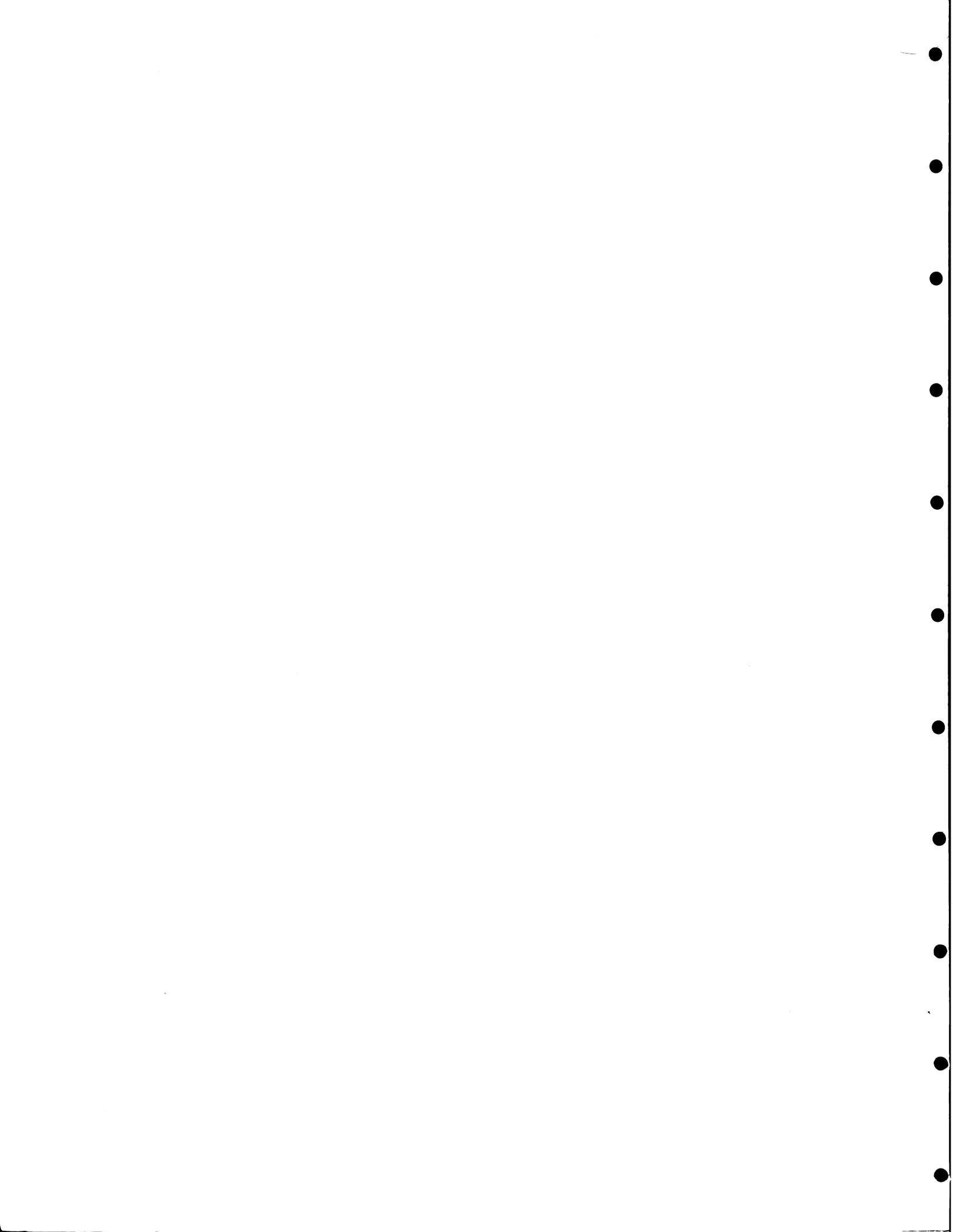
******* CAUTION *******

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



**IR-LINK
MCIR-LINK**

**Infrared Interfaces
for HCS II**

Technical Manual

Release 1.0
10/26/92

Copyright (c) 1992 Circuit Cellar Inc.

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

All rights reserved

COPYRIGHT

IR-Link, MCIR-Link, Comm-Link, and HCSII
are trademarks and copyright (c) 1991 and 1992 of:

Micromint Inc.
4 Park St.
Vernon, CT 06066

DISCLAIMER

While we have attempted to provide accurate and up-to-date information in this manual, Micromint 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.

In this manual, we occasionally refer to other manufacturers' products. 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 individual manufacturers.

Conditions of Sale and Product Warranty

Micromint Inc. and the Buyer agree to the following terms and conditions of Sale and Purchase:

1. Micromint Inc. 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 found to have a defect in materials or workmanship will, at the discretion of Micromint Inc., be repaired or replaced.
2. Products distributed, but not manufactured by Micromint Inc. carry the full original manufacturer's warranty, usually 90 days. Such products include, but are not limited to, power supplies, sensors, I/O modules, LCD displays, battery-backed RAM modules, and disk drives.
3. A minimum inspection fee must be prepaid for the repair of units no longer under warranty. Call Micromint Inc. for information on current minimum charges.
4. Micromint Inc. will not be responsible for the repair or replacement of any unit damaged by user modification, negligence, abuse and mishandling, or improper installation.
5. Micromint Inc. 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 Micromint Inc. A return authorization number may be obtained by phone or letter. Please retain a record of this number, because most subsequent correspondence will refer to this authorization. Under no circumstances should any product be returned to Micromint Inc. without this authorization. Micromint Inc. assumes no responsibility for returns unaccompanied by an authorization number. All returns must be shipped prepaid. Insurance is recommended because losses by a shipping carrier are not the responsibility of Micromint. Repaired units will be returned postage and insurance paid.
7. Micromint Inc. 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.

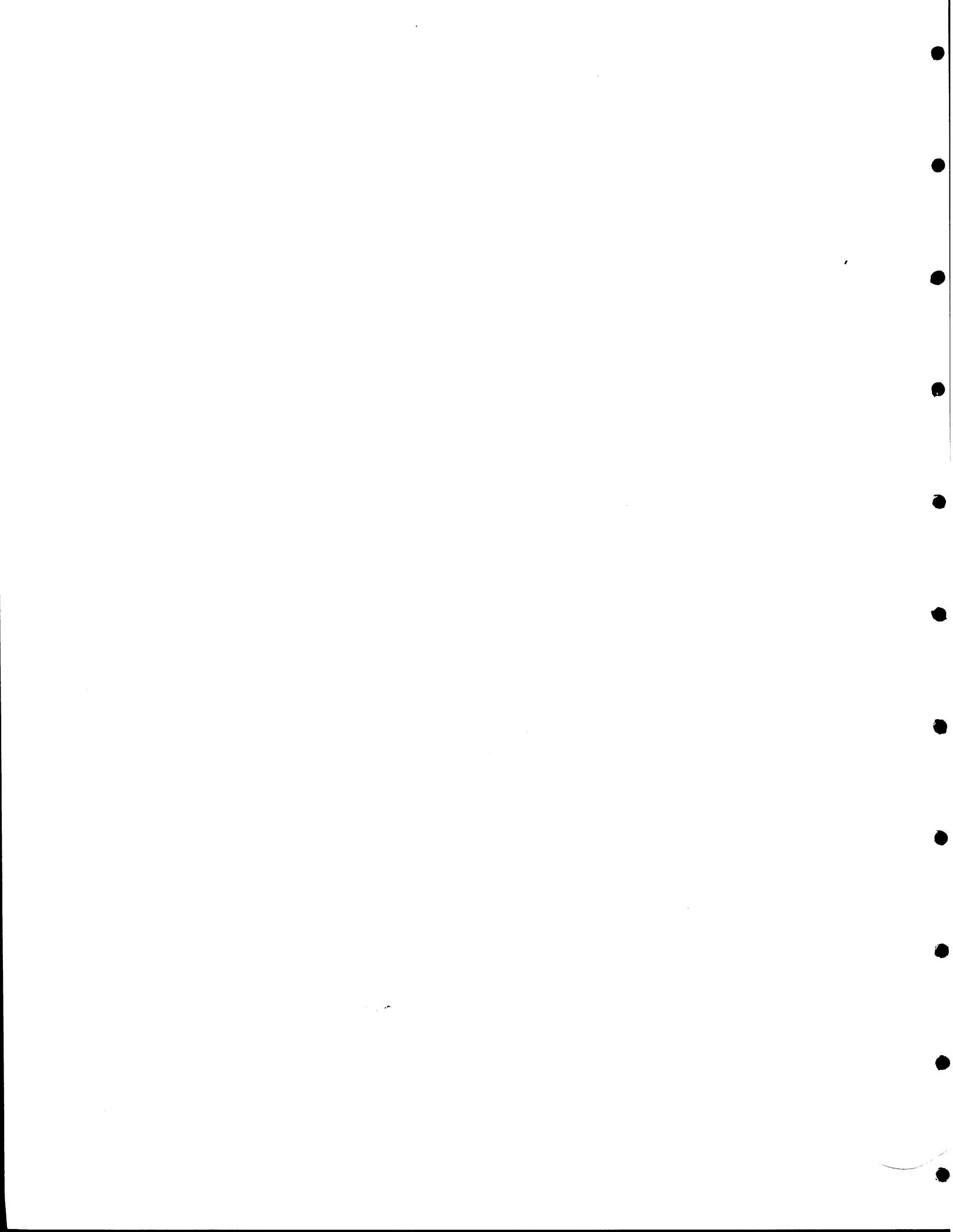
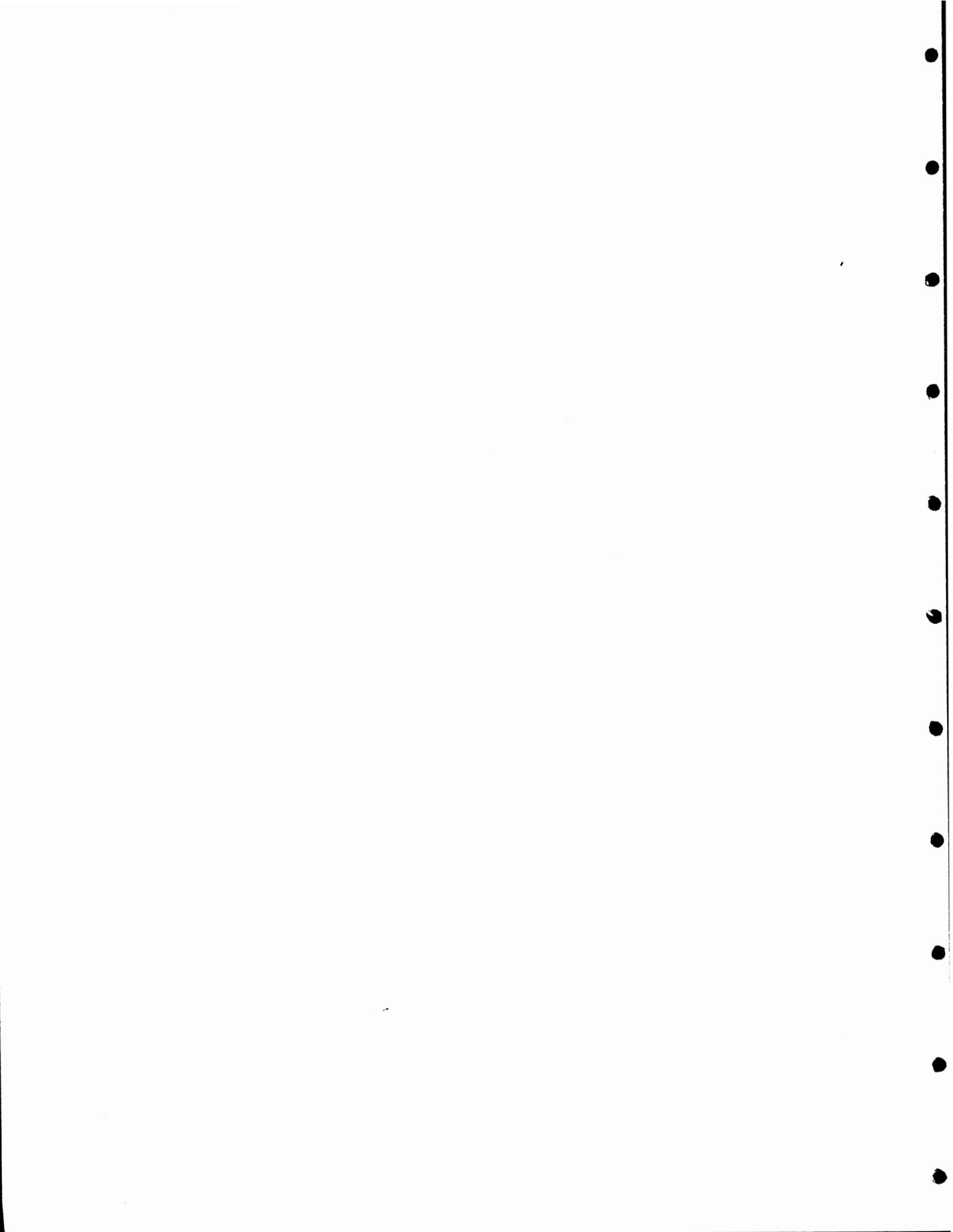


Table of Contents

Section	Description	Page
	Notices	i
	Warranty Information	ii
1	MICROCONTROLLER REVOLUTION	1
2	8031-FAMILY PIN DESCRIPTION	2
3	COMM-LINK EXTERNAL ADDRESSING SPACE	2
3.1	MEMORY DEVICE TYPE SELECTION (JP2 & JP3)	3
3.2	EXTERNAL CODE SPACE (JP2)	4
3.3	RESETTING THE RTC31 & RTC52 (JP1 & JP20)	4
4	RS-232 COMMUNICATIONS (JP3)	5
5	RS-485 COMMUNICATIONS (JP18 & JP17)	6
6	RS-485/POWER CONNECTIONS (T1, T2)	7
7	PROTOTYPING AREA	8
8	I/O SIGNALS (J1)	8
9	(MC) IR-LINK SOFTWARE AND COMM-LINK HARDWARE	9
10	(MC) IR-LINK SOFTWARE	12
10.1	GETTING STARTED - (MC) IR-LINK MONITOR	13
10.2	(MC) IR-LINK COMMAND SET	14
10.3	(MC) IR-LINK NOTES	15
10.4	MCIR-LINK NOTES	16
10.41	LEARNING IR SIGNALS	17
10.42	LOADING AND SAVING HEX FILES	18
10.5	CALIBRATION (TRANSMITTER/REMOTE OSCILLATOR)	19
10.6	BADGE POLLING	20
11	DEFAULT JUMPER CONFIGURATION	21
12	INTERACTIVE STAND-ALONE MODE	22
13	INSTRUCTIONS FOR (MC) IR-LINK KIT ASSEMBLY	25
14	(MC) IR-LINK SCHEMATICS	28
15	(MC) IR-LINK SILKSCREEN	31
16	(MC) IR-LINK PARTS LIST	32



MICROCONTROLLER REVOLUTION

"Expandable", the catch phrase of the eighties, is taking on new meaning in the nineties. 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-LINK 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-LINK 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, which can be obtained from a 9 to 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 external 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

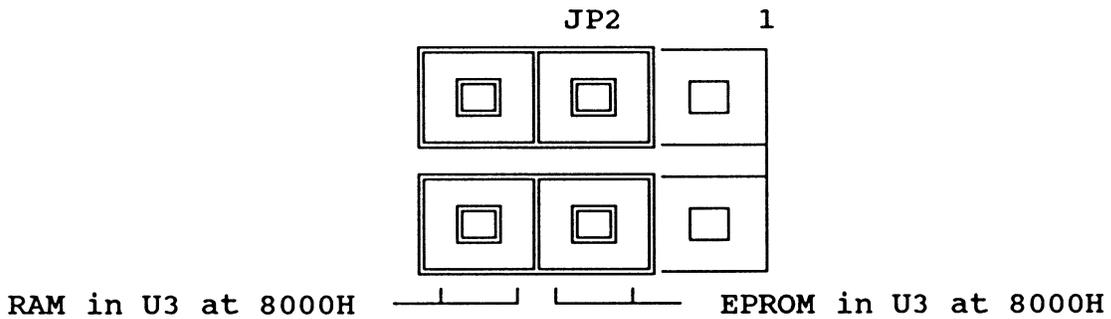
EXTERNAL ADDRESSING SPACE

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

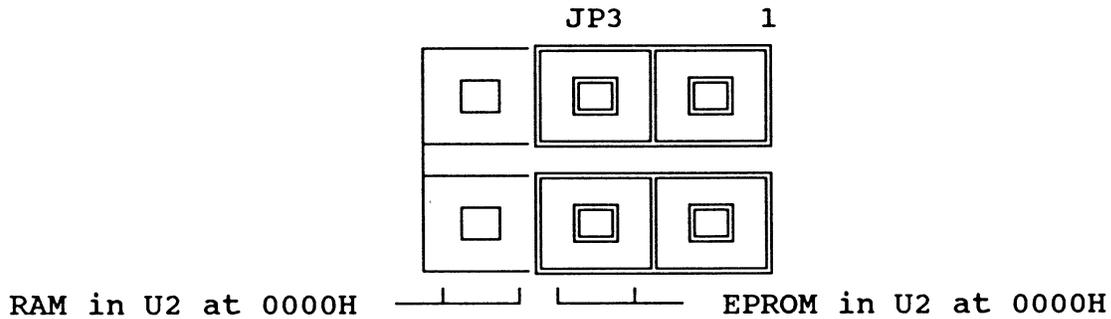
Combinations of 8K and 32K RAMs and EPROMs are allowed on the COMM-LINK board. When using the 8031, your EPROM code starts execution at 0000H. When using the 80C52 (masked with BASIC in the internal ROM), RAM must be at 0000H.

DEVICE TYPE SELECTION

Two memory sockets are provided on the COMM-LINK 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 the memory configuration used with an 8031.



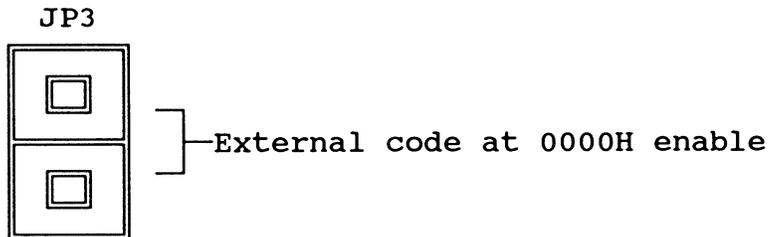
JP2 shows RAM in U3 starting at address 8000H



JP1 shows EPROM in U2 starting at address 0000H

EXTERNAL CODE SELECTION

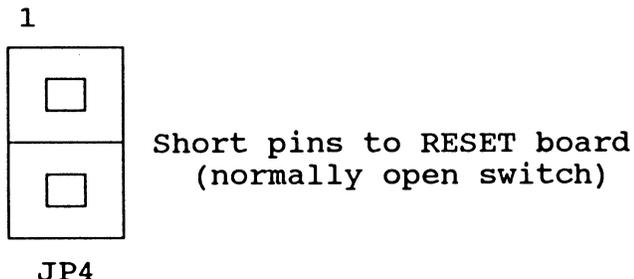
The 8031 microcontroller requires *EA (pin 31 on the micro) to be pulled down to a logic low level. This step instructs the processor to begin executing machine language code starting at address 0000H. The 8052 is masked with BASIC and will run internal code (BASIC interpreter) if *EA is pulled up to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



JP3 shows the microcontroller enabled for external code execution

RESETTING THE COMM-LINK

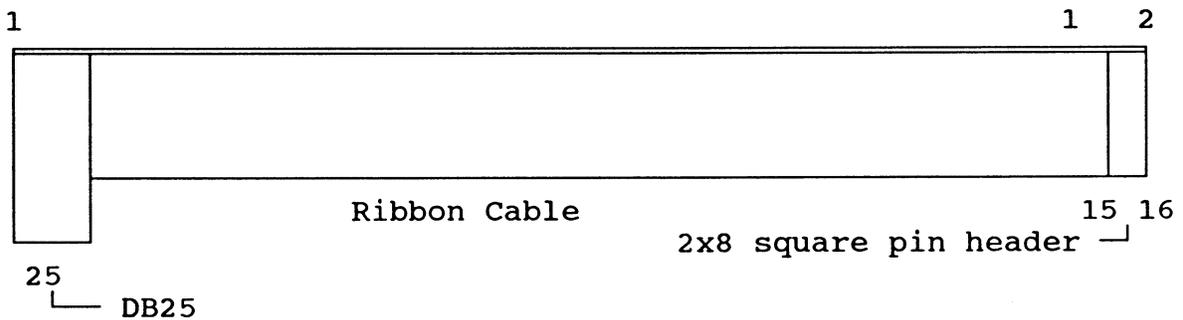
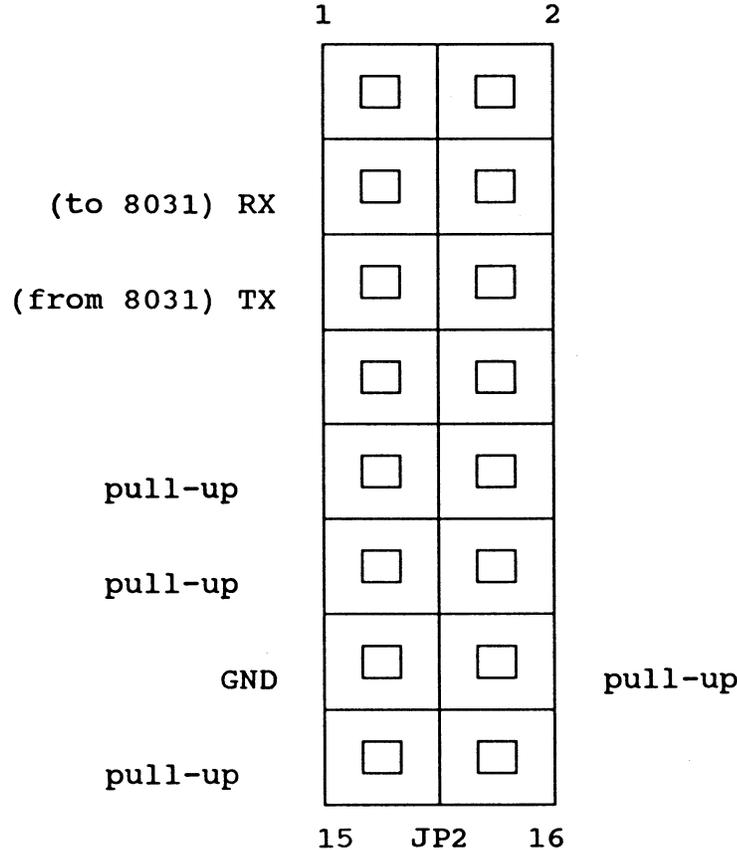
Reset of the COMM-LINK board occurs when a normally open push-button switch is attached to JP4 and momentarily pressed. A logic high is applied to the system RESET line and held momentarily high by an R/C circuit.



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

RS-232 COMMUNICATIONS

The COMM-LINK 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.

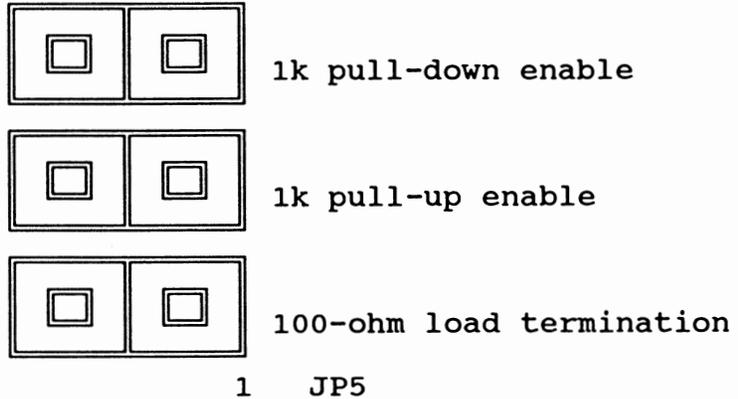


Cable required for RS-232 communications

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. Because each device can transmit and receive, you must adhere to certain protocols to prevent message collision. The simplest being "listen to the line and transmit only if free." (The protocol you use depends 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-LINK board. If you don't require RS485 communications, removing U8 will reduce overall power requirements.



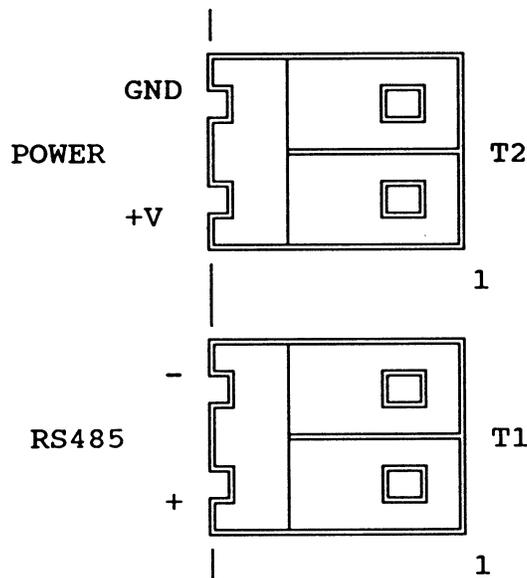
JP5 shows termination of the RS-485 lines enabled

The 100-ohm termination 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 ensure that the twisted pair is coaxed to the idle state when no RS485 driver is active. Any termination should be done on the last node for less confusion.

RS-485/POWER CONNECTIONS

Four screw-terminal connections are used for connecting power and the RS-485 network to the COMM-LINK board. They allow the COMM-LINK board to operate remotely with only a four-wire cable (i.e., telephone cable). Of course, you could use the COMM-LINK board as a stand-alone controller with RS-232 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 to 12 volts of unregulated DC from a remote supply on a second pair of wires or a local 5-VDC regulated supply (with the 0-ohm jumper at R15 installed).



Screw-terminal blocks for Power and RS-485 Communication

POWER for the COMM-LINK

Use T2 for attaching the power supply connections to the COMM-LINK board. The input can be any unregulated DC voltage from 9 to 12 volts. A series diode protects the COMM-LINK board from accidental voltage reversals. The unregulated DC powers a linear voltage regulator on the COMM-LINK 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 0-ohm jumper at R15 to bypass the linear regulator and series diode. Caution: Voltage reversal protection is eliminated when a jumper is placed at location R15.

PROTOTYPING AREA

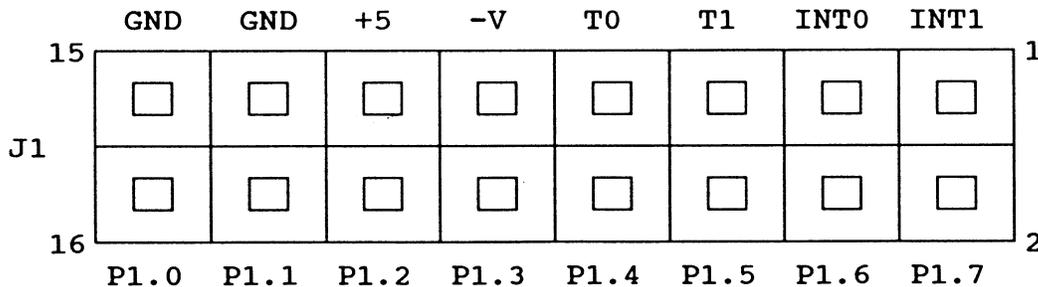
Adding-memory mapped I/O is impossible because the COMM-LINK board does not have expansion addressing circuitry. The 12 bits of TTL I/O are the only means of expansion. Eight bits (of the 12) are bidirectional from PORT1 of the microcomputer. Four TTL open collector drivers are provided to drive LEDs directly (i.e., LED5). These can be removed if you wish to use the lines as inputs.

These are easily used as a parallel bus, with the other 4 bits used as control lines for a connection to most any parallel device. Three-wire serial chips, such as ADC083x or MAX170 A/Ds and MAX500 D/As are also easily implemented.

Three and a half square inches of 0.1 by 0.1 inch prototyping grid will accommodate up to a 40-pin 0.6-inch device with room left over for discretes.

I/O SIGNALS

In addition to 12 bits of I/O brought out to J1, +5 volts, Ground, and -9 volts (about 7 mA) are also available. These signals are located next to the prototyping area for easy access. NOTE: P1.4-P1.7 have open collector drivers. To use these as input lines, remove U9 and jumper pin1 to pin2, pin3 to pin4, pin5 to pin6, and pin8 to pin9.



(MC) IR-LINK SOFTWARE AND COMM-LINK HARDWARE
AS A NETWORK NODE FOR THE HCS II

The HCS II is an expandable network-based, intelligent-node industrially oriented supervisory control system that, 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 (SC) connected to up to 31 other functional modules (called Links) via an RS-485 serial network. The SC and the Links can operate independently and do not need the system controller or other links to function. This feature 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 SC is quite literally the brains of the system. While every Link has its own processor, they each rely upon the SC for timing, control commands, and overall system coordination.

The SC is a 9.218-MHz 64180-based single-board computer. It has two serial ports, up to 96 kB of memory (32 kB are battery backed), an eight channel 8-bit or 10-bit ADC, 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 SC 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 the 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 SC'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 SC, 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 64 kB of RAM, EPROM memory, or both. The single serial port is configured to operate both RS-232 and RS-485. When connected to the SC in normal operation, the RS-485 is attached using a twisted pair. 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 SC.

For example, the (MC)IR-LINK works via the same RS-485 link as the other network nodes to send and receive data. In this instance, the data consists of wireless infrared transmissions, which are sent and received by the (MC)IR-LINK board.

The (MC)IR-LINK board contains both a 38 kHz infrared transmitter and infrared receiver. The data it transmits and receives is called Manchester coding. A single transmitted character consists of 9 bits and represents 1 of 512 combinations (multiple characters can be used to represent significantly more combinations if necessary).

The idea behind the (MC)IR-LINK is to use a trainable hand-held IR remote controller like the one you might already be using with your TV and have it contain 10 or 20 (or all 512) of these codes. Simply aim the remote at the PL-LINK and that code would be received by the HCS II's SC. Within the SC's event repertoire would be events such as IF IRCODE=20 then set pump ON, LIGHT6 OFF, FAN1 ON, or, IF IRCODE=1 then All lights off, and so forth. Given 512 interpretable commands, your hand-held control could do quite a bit.

Because (MC)IR-LINK has the capability to both send or receive Manchester codes, it can serve as the basis for a simple "people tracking" system based on either active or passive badges. In this context, active badges send their ID periodically without prompting and passive badges respond to a poll from the tracking system.

Active badges require no special setup because the (MC)IR-LINK will record all IDs seen by its sensor. Each ID is cleared as it is reported, so each Query command returns new IDs since the last Query. Passive badges must receive their own ID to trigger its transmission. The (MC)IR-LINK can send IDs (transmitting high-powered IR throughout a room requires additional circuitry to the (MC)IR-LINK board) at regular intervals and will record the response; there is no difference between a polled ID and any other ID. The (MC)IR-LINK does not record its own transmissions, so there is no conflict between the outgoing and incoming IDs. HCS II can also accommodate up to eight (MC)IR-LINK units on its RS-485 line, so these Links can be placed in separate locations to broaden the tracking area.

Intelligent Links simplify testing and problem diagnosis. More often than not, the locations 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 to 12 VDC input.

While the COMM-LINK circuitry runs on 5 volts, the design incorporates a linear regulator so the voltage can be derived from a wide-ranging DC input. The intention is to promote connecting the individual Links via a four-wire cable, rather than just a two-wire twisted pair, when local power is unavailable. Two wires supply the RS-485 while the other pair supplies +12 volts and ground. (Actually 9 to 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 +5 volts.

(MC) IR-LINK SOFTWARE

Set up your board as suggested for an 8031-based system, with the IR (or MCIR) -LINK EPROM in U2 and a RAM in U3. The IR-LINK firmware requires an 8K RAM in U3 while the MCIR-LINK requires 32K of battery backed RAM (Smart socket between the U3 IC-socket and the 32K RAM chip). Upon power up or reset, LED5 will blink about once a second, reassuring you 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 or RAM size
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 2 MHz)

Bit usage for the (MC)IR-LINK Monitor

P1.7	LED	heartbeat at 1 Hz and IR indicator
P1.6	OUTPUT	A
P1.5	OUTPUT	B
P1.4	OUTPUT	C
P1.3	INPUT	high bit of network address, gnd=1
P1.2	INPUT	middle bit, ditto
P1.1	INPUT	low bit of network address
P1.0	INPUT	gnd at reset -> RS-232 interactive mode
P3.3	INT1	HIGH to turn IR transmit LED ON
P3.2	INT0	INPUT from Sharp IS1U60 IR detector
P3.5	T1	transmitter oscillator input
P3.4	T0	controls RS-485 driver (LED ON = disabled)

Sharp IS1U60 Connections:

Hold the module with leads down and IR window facing you. Lead 1 is on the left.

1	Output signal, ground for IR present
2	Ground
3	+5V

GETTING STARTED - (MC)IR-LINK MONITOR

For RS-232 Use (N1-network mode default):

1. Set Communications at 9600-bps 8N1.
2. Remove 75176 transceiver chip, install MAX232.
3. Connect PC to RS-232 header as usual.
4. Verify commands to satisfy yourself that it works...
 Prefix all commands with "! IRx " (note spaces!). x must match
 the net address set on P1.1-P1.3 (default=0, no jumpers)

If you have trouble, install JP7 (which grounds P1.0) 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 at 9600-bps 8N1.
2. Install 75176 transceiver, remove MAX232.
3. Ground P1.1-3 to set network address (ground = "1" bit).
4. Connect PC through RS-485 converter.
 Your software must enable/disable RTS correctly!
 Make sure P1.0 is not grounded when you turn power ON.

CHECKSUMS

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

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

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 the complement to 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 = 0, then it verifies correctly

(MC) IR-LINK COMMAND SET

All time intervals are in units of 5.16 ms

- A=string** Set network address (16 characters max)
- B** Set badge response delay (default 40 ticks)
- CRn** Calibrate remote MC145030 bit clock (12.4 kHz, 1290 μ s)
n=0-3 sets report detail
- CT** Calibrate transmitter oscillator
connect 38-kHz output to T1 input
- D** Dump program status (debugging use)
- E** Show and clear error flags (debugging use)
- * **HDn** Dump MCIR RAM in Intel HEX format, base address n
- * **HEn** Dump MCIR EPROM data in Intel HEX format, base address n
- * **HR** Receive Intel HEX data to RAM storage, address ignored
- In** Set badge polling interval (default 190 ticks)
- Ln** Set logging mode (bit mapped)
- L report current mode
 - L0** **disable (default)**
 - L1 show received IR messages
 - L2 show transmitted IR messages
 - L4 show generated polls
 - * L8 trace MCIR data capture
 - * L16 trace MCIR data transmission
 - * L32 trace MCIR data decompression
 - * L64 trace MCIR data compression
- Nn** Set network/interactive mode
- N report current mode
 - N0 sets interactive mode
 - N1** **sets network mode (no error messages) (default)**
 - N2 sets network mode with command echo (w/err msgs)
 - N3 sets network mode, returns '*! IRO OK' when a
correctly formatted command is received (no err msgs)
- Ox=n** Set output bit x to state n (OA=1, OB=0)
bits are normally high after reset
- Pn** Set number of badges to poll (default 0, no polling)
- Q** Query and reset received IDs
- Q dump all 512 IDs in hex bytes (about 132 chars)
ID 0 is bit 0 of first byte
 - Qn report ID n status in format 003=1
 - Qn-m report IDs n through m in hex bytes
ID n is bit 0 of first byte
- RESET** perform power-on reset must be completely spelled out!

- Sn** Send IR ID n (decimal, 0-511)
- * **SMn** Send MCIR signal n (1-999 from RAM, 1001-1999 from EPROM)
- * **Tn** Train MCIR signal n into RAM (1-999 only)
- * **TD** Set duty cycle correction (0-255)
- * **TX** Clear MCIR data from RAM

(*) denotes added commands when using MCIR-LINK firmware

(MC)IR-LINK Notes

(MC)IR-LINK sends and receives the IR codes defined by the MC145030 Remote Control Encoder/Decoder. The nominal bit frame is 1.29 ms, corresponding to a 12.4-kHz clock. (MC)IR-LINK can receive signals from encoders with clock frequency offsets of a few percent, but it will always transmit at the nominal frequency.

The MC145030 transmits two copies of the ID (its "address" input pins), but requires only one correct ID for reception. (MC)IR-LINK requires both a correct bit format and matching copies of the ID before it will record the signal.

Signals from other remote control units with different modulation schemes will generally be ignored, but it is possible that portions of the signals will "match" a correct ID. (MC)IR-LINK discards invalid signals and begins waiting for a new signal as soon as it detects an error; there is no fixed delay.

The A command sets the network address. This command is most useful for battery-backed RAM; if the stored address is not valid on power up, the firmware will use the default address of IRx (x matching the input on P1.1-P1.3). The last character of the string will be replaced by the p1 bit setting, so you should use a dummy digit at the end. For example, CONTROLx will become CONTROL0 through CONTROL7.

The Ox=n command controls output bits P1.4-6. The bits are normally high after a hardware or software reset. Issuing OA=0 (that's "Oscar Able equals Zero") sets P1.6 LOW, OA=1 sets IR high. Because 8031 CPU bits have a weak pull-ups, you should probably use drivers with active low inputs.

MCIR-LINK Notes

The MCIR-LINK firmware adds infrared remote control recording and playback to the functions of the original IR-Link firmware. You may use any feature at any time.

Appliance remote control functions

The MCIR-LINK firmware allows you to record the signals from ordinary infrared appliance controllers and play them back under computer control. The hardware includes 24 kB of RAM for signal storage and 2 kB of EPROM; either or both may be used at any time. Unless you provide battery backup (smart socket), all signals in RAM will vanish when the power fails.

However, battery-backed RAM does not guarantee that the IR signals will be completely safe. If the firmware is "glitched" by a transient it may write garbage throughout RAM. You should use a well-filtered battery backup power supply to prevent glitches and protect the MCIR-LINK hardware from ESD and RFI problems.

The MCIR-LINK receives signals through a SHARP IS1U60 IR receiver. This device was designed to receive signals with a 38 kHz carrier modulation, but will work well with other frequencies as long as the remote unit is nearby. Because the IS1U60 strips the carrier information and sends only the demodulated data bits to the CPU, the firmware has no way to measure the actual remote control carrier frequency.

Infrared transmissions from the MCIR-LINK board are modulated by a hardware oscillator that is normally set to 38 kHz. You may adjust the actual frequency to match your remote units, but you must pick a single frequency that will work with all the appliances and other receivers you intend to control.

The firmware compresses the IR signals before storing them in RAM. In round numbers, a keystroke occupies about 100 bytes of storage, so there is room for about 200 keys in the 24 kB RAM and 15 to 20 keys in the 2 kB EPROM space.

Learning IR signals

The MCIR-LINK module must be in manual mode (P3.0 grounded during reset) to learn IR signals from a remote control. Use the Tn command, where "n" is a number from 1 to 999 that will identify the signal. The firmware deletes a signal with that number from RAM before the new signal is learned, so be careful. The heartbeat LED is ON while the firmware is waiting for the signal. While the LED is in the OFF state of its heartbeat (blinking mode), it will flicker when the IR receiver sees IR from an IR remote, an external light source, or electrical noise from the board. If you are NOT sending IR and the LED still flickers, move it physically away from the noise source before trying to record IR signals.

Aim the remote control at the IS1U60 and tap the key once to send the signal. The firmware records the signal, compresses it, stores it in RAM, and tells you how much RAM is left. If there is not enough room or something goes wrong, the firmware issues messages describing the problem.

Use the SMn command to test the signal, where "n" is the same number you used with the T command. The firmware decompresses the stored data and sends the signal; the appliance should respond just as though you had pressed the remote control key.

The IR LED on the MCIR-LINK board is driven at about 90 mA, which is considerably less than most special-purpose remote controls. You may need to position the MCIR-LINK board closer to the appliance and ensure that the LED is aimed directly at it to ensure good reception. If you are controlling several appliances you may need a custom LED fixture that broadcasts IR signals throughout the room.

Use the TX command to ERASE all signals stored in RAM. There is no confirmation for this command, so BE CAREFUL!

Loading and saving HEX files

Once you have stored the signals in RAM, you should save them on diskette "Just In Case" the power goes out. Use the HDn command to dump the contents of RAM in Intel HEX format and capture the resulting output into a disk file. The "n" value sets the starting address of the HEX file and defaults to 7800 so you can burn the resulting file directly into the EPROM (the MCIR-LINK EPROM has room from 7800-7FFF).

Use the HR command to transfer a HEX file directly to the MCIR-LINK RAM. The firmware ignores the HEX line addresses and puts the data in the right place, replacing any previous RAM. Combining two files is impossible, although you can load one file, learn additional keys, and save the result as a new HEX file.

The HE command dumps the IR signal stored in EPROM. This step is included for completeness, because you will already have the HEX file you used to program the EPROM.

File Capture

Your communications program must be able to capture and send files. For example, when using "PROCOMM" to capture the HEX output file use the <Alt-F1> (log file ON/OFF). You may need to edit out any garbage characters logged into the file prior to the first ":". To send the file back to the MCIR-LINK board, simply upload the file as ASCII.

Storing and using signals in EPROM

Once you have a HEX file, you can program it into the firmware EPROM using any EPROM programmer. The data must start at 7800 and not extend past address 7FFF. The standard MCIR-LINK EPROM is blank in that area, so you can add the new data directly to the EPROM (however dumb that may be; you should make a copy of the EPROM first and use only the copy!).

Use the SMn command as described above, but add 1000 to the ID you would use for the RAM signal. For example, if you trained a signal as T1345, use SM2345 to send it from EPROM.

Calibration

You should use the calibration routines in interactive mode (ground P1.0 and reset the CPU) because they disable all interrupts. Use any serial communications routine on your PC to send commands and observe the results.

The heartbeat LED (P1.7) will blink rapidly while the firmware is, in the calibration routines.

Transmitter

To adjust the transmitter, install jumper JP6, connecting the 38 kHz oscillator output to the T1 CPU input, then enter the CT command. The firmware will measure the frequency once per second and display the numeric value and a flying-dot tuning graph:

```

35990 Hz [<<< under 36000]
36249 Hz [  *                               ]
36466 Hz [      *                           ]
37309 Hz [                *                 ]
37865 Hz [                    *             ]
37999 Hz [                        *         ]
38466 Hz [                            *     ]
39309 Hz [                                * ]
40001 Hz [over 40000 >>>]

```

Adjust the trimpot for 38 kHz, then press any key to get out of the calibration routine. Remove the jumper.

Remote unit MC145030 oscillator

Enter the CR command and send an IR signal from the remote unit. The firmware will measure the Manchester bit frame times, which are set by the remote unit's oscillator. Tune the remote for a frame time of 1290 μ s, which implies a 12400 Hz oscillator frequency (16/period). The display will resemble the following:

```

Bit frame time is 1310  $\mu$ s
Bit frame time is 1301  $\mu$ s
Bit frame time is 1290  $\mu$ s
Bit frame time is 1268  $\mu$ s
Bit frame time is 1291  $\mu$ s

```

Press any key to get out of the calibration routine.

You can display more details about the incoming signals by entering CR1, CR2, or CR3.

Badge Polling

(MC)IR-LINK can serve as the basis for a simple "people-tracking" system based on either active or passive badges. In this context, active badges send their ID periodically without prompting and passive badges respond to a poll from the tracking system.

Active badges require no special setup, because (MC)IR-LINK will record all IDs seen by its sensor. Each ID is cleared as it is reported, so each Query command returns new IDs since the last Query.

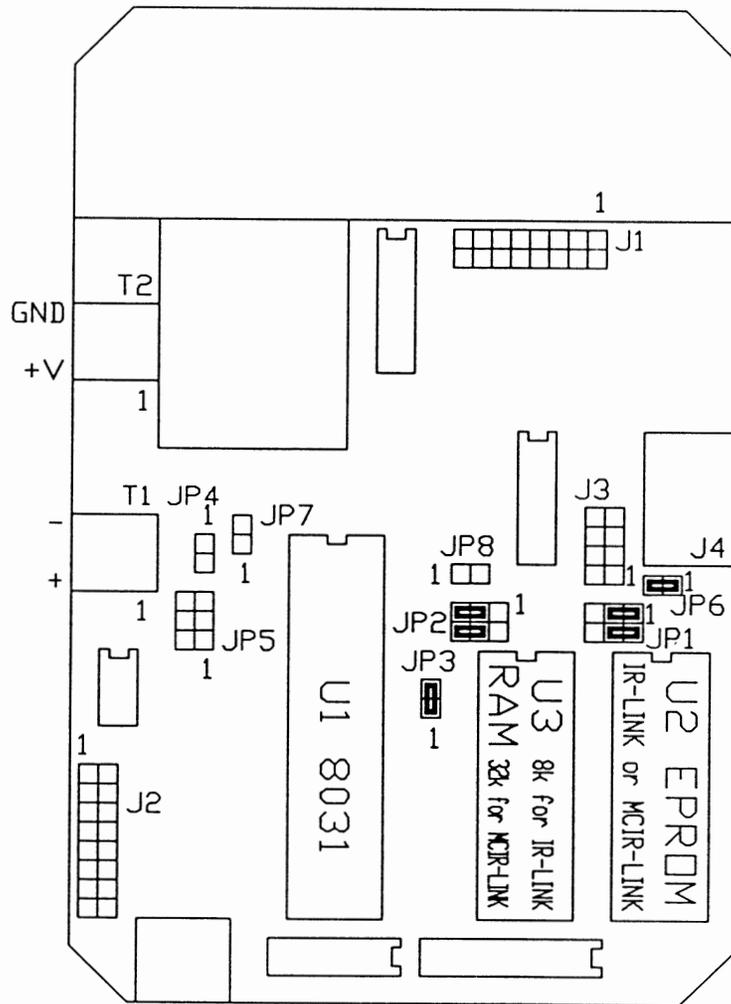
Passive badges must receive their ID to trigger it. (MC)IR-LINK can send IDs at regular intervals and will record the response; there is no difference between a polled ID and any other ID. (MC)IR-LINK does not record its own transmissions, so there is no conflict between the outgoing and incoming IDs.

(MC)IR-LINK has three badge-polling controls. The default values should handle most situations, but each can be changed as needed.

The P command sets the number of badges to poll; the default is P0 to poll zero badges, which disables polling. Badge polling starts with ID 0 and continues to n-1, where n is the number of badges.

The I command sets the interval between successive polls. The default is about two seconds. Because all badges must be active to analyze each poll and each matching poll triggers an IR message, polling too frequently will reduce the lifetime of the badge batteries (or, conversely, require bigger batteries for a given life).

The B command sets the delay between the end of (MC)IR-LINK's polling transmission and the start of the next transmission. This delay ensures that (MC)IR-LINK does not transmit a poll and an S command so close together as to clobber the badge response. The default is about 200 ms, which should be adequate.



Default jumper configuration for the COMM-LINK board using the (MC)IR-LINK FIRMWARE EPROM in U2 at 0000H and RAM in U3 at 8000H running with an 8031 microcontroller.

INTERACTIVE STAND-ALONE MODE

NOTE: THESE COMMANDS ARE INCLUDED IN THE (MC)IR-LINK FIRMWARE FOR THE HCS II. THE RS-232 INTERACTIVE FIRMWARE DOES NOT INCLUDE THE NETWORK COMMANDS. Refer to the source code listing for an explanation of the undocumented status values used for code debugging only.

Install JP7 and follow the directions for RS232 use under the section "GETTING STARTED - (MC)IR-LINK MONITOR"

Upon RESET you should see a sign-on banner similar to:

```
IR-Link Networked Infrared Gateway Controller, Version x.x
Copyright 1992 Circuit Cellar, Inc.
IR0 >
```

Enter the letter B to display the maximum "Badge response time".

```
IR0 >B
Badge response delay is 40 ticks
IR0 >
```

The number of ticks * 5.16 mS (206.4 mS) is the maximum time the system waits for a response from a badge before polling the next badge.

Enter the letter D to "Dump the program status".

```
IR0 >D
Interactive mode. Max stack at 53 of 7F
IR interface status:
IRState 00, IRecStat 00, Badgedly 21, BadgeRsp 40
IRCtr 00, INumSend 00, IRBits0 00, TOTicks 6D
Trans: Head 00, Tail 00, Ctr 00, Max 00,
[IOutHead]->0000, [IOutTail]->0000
Rec: Head 00, Tail 00, Ctr 00, Max 00,
[IInHead]-> 0000 , [IInTail]-> 0000
Badge polling status:
PolNumIDs 0, PolNextID 0, PollDly 0, PollInterval 390
IR0 >
```

Enter the letter E to display the "Error flags".

```
IR0 >E
Error flags 00
IR0 >
```

Enter the characters I10 to "Change the badge polling interval".

```
IR0 >I10
*** Badge polling interval forced to 20
IR0 >
```


Enter the characters S0 to send out an IR ID 000 (any ID from 0-511 can be sent).

```
IRO >S0
IRO >
```

Any IR badge with ID 0 that sees the IR ID transmission will answer back. The (MC)IR-LINK will identify the badge as "within range". A trainable IR remote can be used as a remote signalling device once it has been trained by the (MC)IR-LINK's send ID command.

Learning and mimicking hand-held remote controllers is an added feature only available with MCIR-LINK. The SMn command is used to send out previously learned signals. Substitute the "n" for the command you wish to send. This digit is the same you used when the signal was trained, so get out that list you made during training.

```
IRO >SM1
IRO >
```

If you had never recorded any IR signals you would get:

```
*** IR data not valid
IRO >
```

If you did record signals but not this one you would get:

```
*** IR record not found
IRO >
```

Learn hand-held IR remote controller transmissions by using the command Tn. Substitute 1-999 for the "n". Make sure you make a note of the command number and what function it performs.

```
IRO >T1
--- Send signal from IR remote, press ENTER to quit...
```

At this point, the system is waiting for you to tap the handheld remote's button.

```
Raw data captured
There are xxxxx bytes of RAM available
IRO >
```

The MCIR-LINK board has now captured the transmission. The board can actually pick up noise (IR from other sources and clocking through the circuit board), which will add garbage to your signal, so keep the IR receiver as far away from external lights and the on-board circuitry as possible. Each learned transmission should take less than 100 bytes. This aspect is indicated by the decreasing xxxxx bytes of RAM available message. Commands can be learned and relearned in any order.

Although the learned commands are semi permanent with battery-backed RAM, you should "HDn" HEX dump the file to diskette just in case you need to "HR" HEX receive it again if the system fails.

Instructions for Kit Assembly**Tools Required for Assembly:**

Low Wattage or Temperature-controlled Soldering Iron
Rosin Core Solder
Lead Cutters

Additional Tools to Ease the Assembly:

Needle-nose Pliers
Lead Bender
Solvent for Removing Rosin (flux)
Screwdriver

Tools for trouble shooting (and their use):

Oscilloscope	Viewing AC/DC signals
Logic Probe	Indicating Logic Levels (activity)
Continuity Checker	Determining Shorted or Open Traces
Volt/Ohm Meter	Checking Power Supply, Logic Levels, or Continuity

Familiarize yourself with all of the parts included in the kit. Pay particular attention to the proper orientation of the parts. Markings might include a bump, hole, number, arrow, or notch indicating pin1 (or the pin1 end) of ICs and sockets. A stripe may indicate a plus or minus potential lead of a capacitor or the cathode of a diode. An LED might indicate the cathode by a notch or flat side on the girth of the component.

Inspect the PC Board prior to installing any parts. If you hold it up to a lamp, you can usually see the signal traces fairly clearly. Eye each trace for defects, a copper short between adjacent traces or pads, or a break in the copper trace. Verify any traces that look shorted by first looking at the schematic to verify they should not be connected, then checking the traces for continuity. You should check traces that seem to be broken for continuity as well. Circuit Cellar Inc. inspects each and every board for manufacturing defects; we feel confident that the components packaged for you are free from defects. However, inspecting your PCB is much easier before any parts are inserted that would obscure a defect from view.

The best approach to use in building a kit is to install the smallest (or shortest) parts first. Start with any small signal diode. Use the FOIBSAT method on each part or group of parts: Find, Orient, Insert, Bend, Solder, And Trim. Find the part, verifying it using the parts list and silkscreen layout. Orient the part correctly using its markings on the silkscreen layout. Insert the part into its designated location, forming the component leads if necessary. Bend over the component leads to prevent the part from falling out, preferably in the direction of the connecting circuit board trace. Solder each lead, filling in the area between the lead and its plated-through hole. Finally, trim the leads to prevent shorting between traces, components, or holes.

Continue with the 1/4-watt resistors, IC sockets, and smaller monolithic capacitors. Next, insert all the square pin headers and jumpers. These can't be easily bent and will simply fall out if not handled one at a time. Start by soldering only one or two pins, flipping over the board to check that the header isn't tilted, then fix by reheating the soldered pin and/or complete by soldering the rest of the header's pins if necessary. Follow up with the larger components, resistors, capacitors, and axial semiconductors.

Finally, the taller, odd-shaped components are added, transistors, LEDs, crystals, potentiometers, connectors, heatsinks, and so forth.

MOUNTING THE IR DEVICES ON THE PCB

Your kit may include one-LD123 (gray IR transmitter with three leads, the center one cut short) or two-TSUS5200 devices (gray IR transmitter with two leads). The LD123, designated LED1, contains two LEDs within one package. The TSUS5200s, designated as LED3 & LED4, are used as alternate IR sources to LED1. LED1 or LED3 & 4 can be mounted vertically and transmit IR up from the board's surface or bent at an angle if you wish to aim the IR off the side of the board.

If the IS1U60 IR receiver Q5 (green rectangle with a bubble) is mounted vertically, it receives IR through the bubble from the side of the board. If you wish to receive IR beamed toward the board's surface, leave the leads long and bend them over the top of JP6 so the bubble is pointing away from the board's surface. Mounting the receiver on the board can cause noise to be picked up from high-speed signals produced on the board, which can be seen as flickering in the heartbeat LED. This noise can cause reception problems. To reduce noise, bend the receiver away from the board or mount it externally.

**MOUNTING THE IR DEVICES EXTERNAL TO THE PCB
(within 3 feet)**

Your kit may include one-LD123 (gray IR transmitter with three leads, the center one cut short) or two-TSUS5200 devices (gray IR transmitter with two leads). The LD123, designated LED1, contains two LEDs within one package. The TSUS5200s, designated as LED3 & LED4, are used as alternate IR sources. If you wish to mount the IR transmitter externally but three feet, such as through the wall of an enclosure, just extend them with short pieces of hook-up wire. One end of the hook-up wire connected to the component's leads and the other goes to the into the holes for the original component. Check for correct polarity of the component's connections.

Extend the IS1U60 IR receiver Q5 (green rectangle with a bubble) the same way as the IR transmitter. It receives IR through the bubble, which is actually a lens.

**MOUNTING THE IR DEVICES EXTERNAL TO THE PCB
(greater than 3 feet)**

Your kit may include one-LD123 (gray IR transmitter with three leads, the center one cut short) or two-TSUS5200 devices (gray IR transmitter with two leads). The LD123, designated LED1, contains two LEDs within one package. The TSUS5200s, designated as LED3 & LED4, are used as alternate IR sources. If you wish to mount the IR transmitter externally at a distance from the circuit board, you must use an external driver at the end of the wiring run. J3 can provide +5 volts, Ground, and the logic signal to control your external driver and IR transmitter(s).

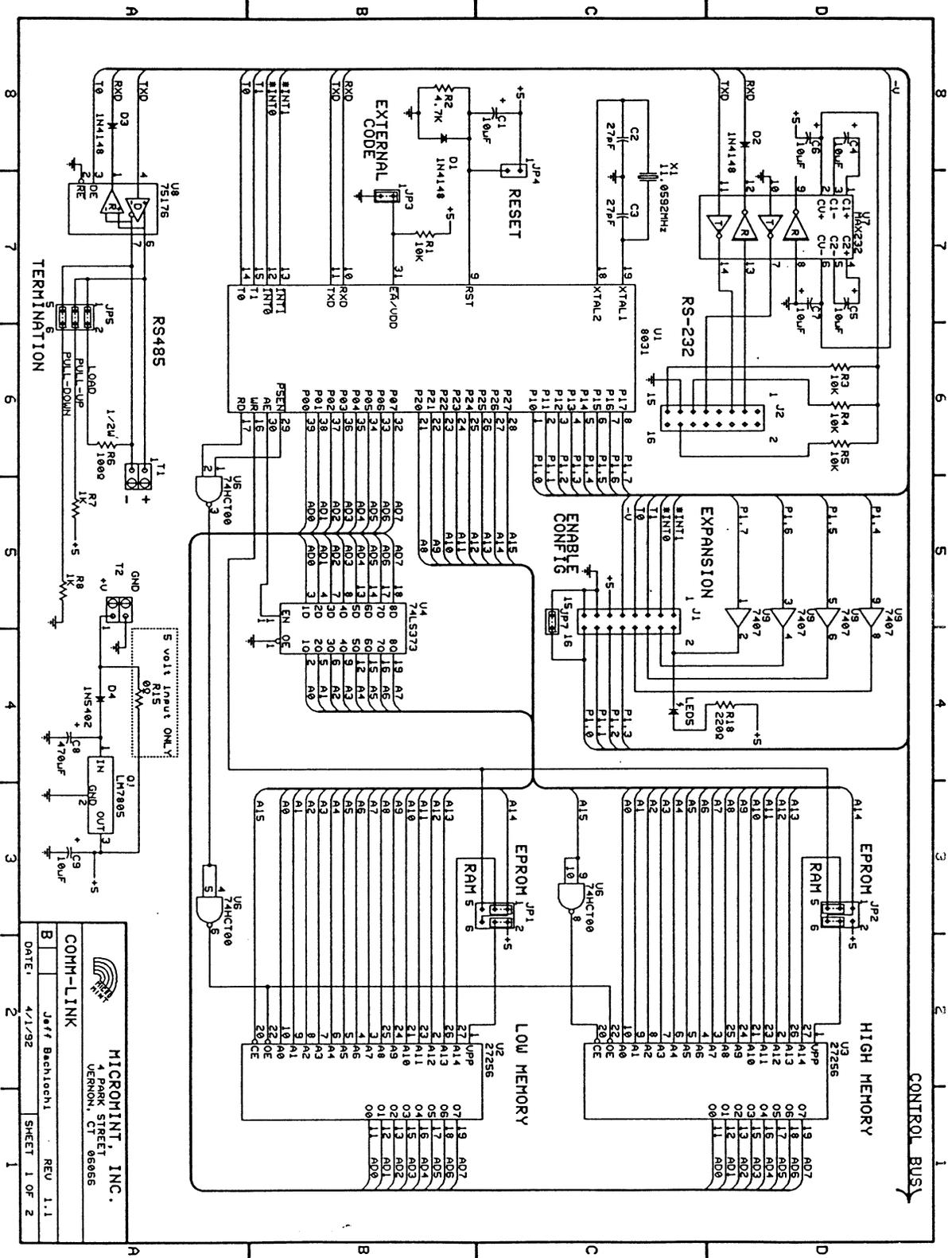
You must duplicate the circuitry including R12, Q3, R14, and LED1 (or LED3 & LED4). Extend the IS1U60 IR receiver Q5 (green rectangle with a bubble) using the signals on J3. It receives IR through the bubble, which is actually a lens. Check for correct connections to all external transmitters and receivers to prevent electrical damage.

Prior to installing the ICs, giving the PCB another inspection is a good idea. Look for unsoldered or untrimmed leads. Cleaning the solder side of the PCB with a flux remover will make inspection easier, not to mention less sticky.

Another suggestion is to use a volt-ohm meter and measure the resistance between Ground and any power supply input. A short circuit here could ruin your power supply and won't allow your board to operate properly. Power up the board before inserting the ICs to check for proper voltage.

Refer to the parts list and silkscreen for proper IC orientation and insertion. This concludes the assembly instructions.

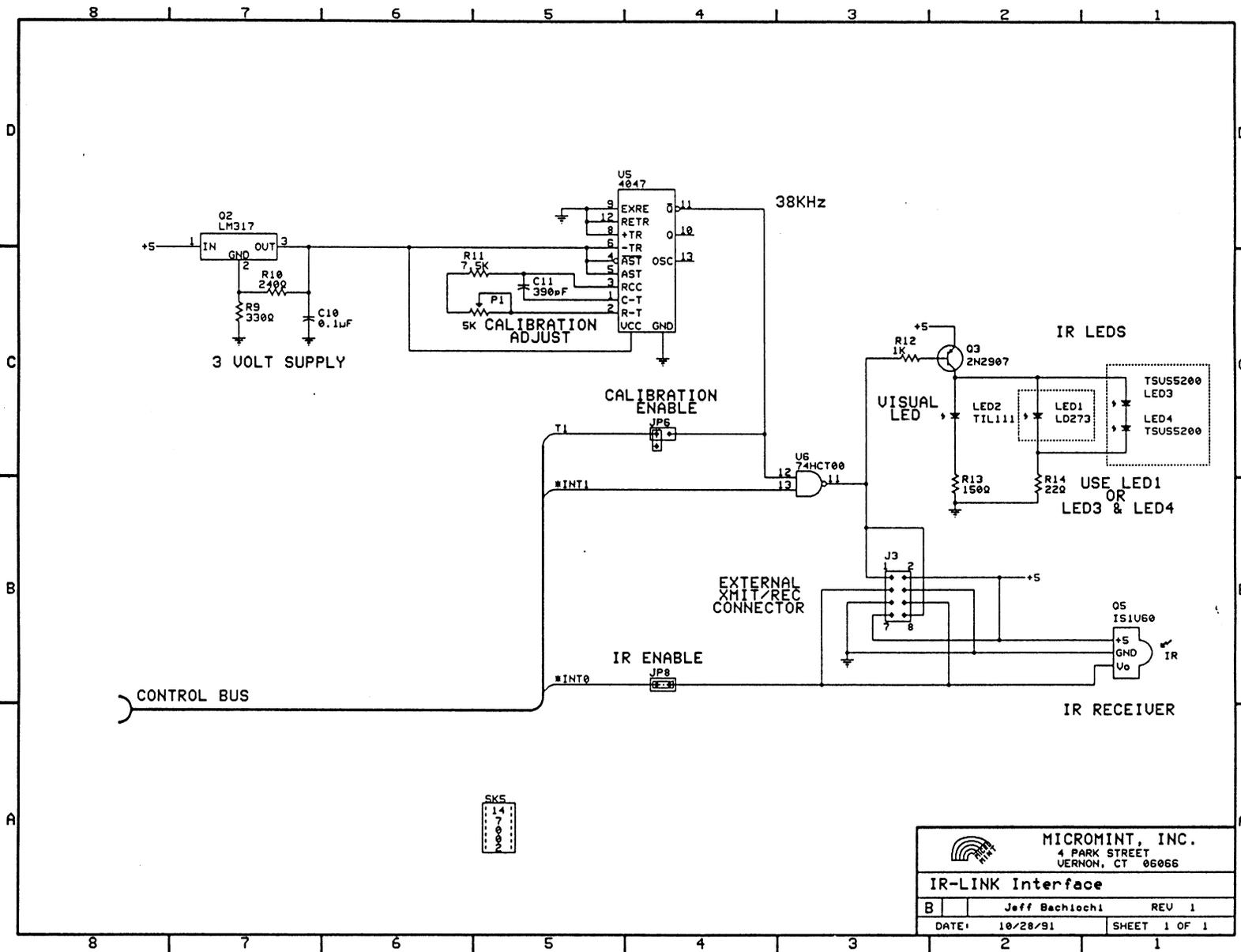
IR-LINK



(MC) IR-LINK SCHEMATICS (1 of 3)

MICROMINT, INC.
 14 PARK STREET
 VERNON, CT 06065

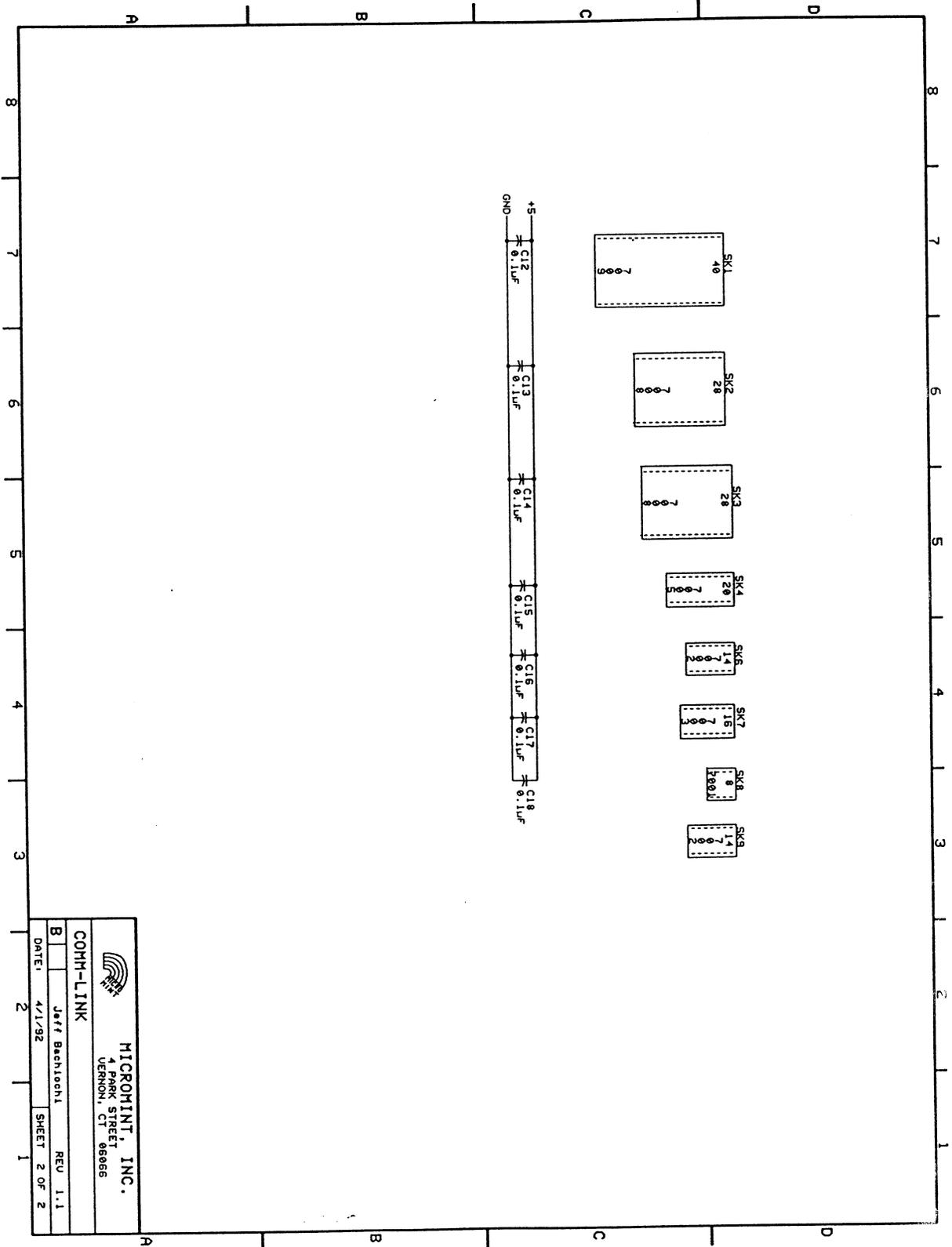
COMM-LINK
 Jeff Bachiochi
 REV 1.1
 DATE: 4/1/92 SHEET 1 OF 2



(MC) IR-LINK SCHEMATICS (2 of 3)

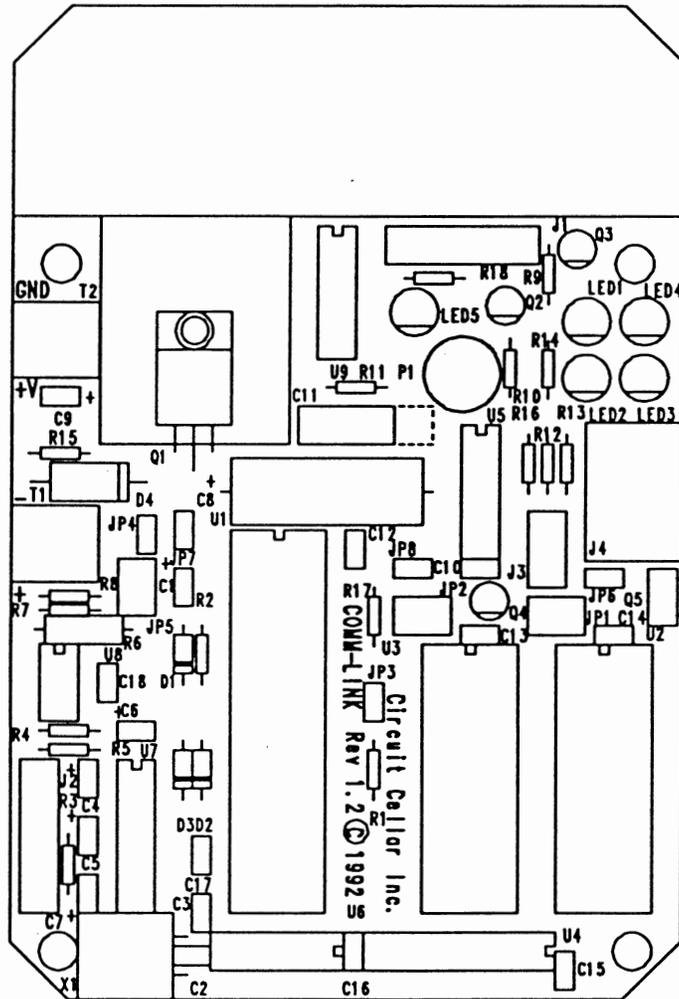
 MICROMINT, INC. 4 PARK STREET VERNON, CT 06066		
IR-LINK Interface		
B	Jeff Bachiochi	REV 1
DATE:	10/28/91	SHEET 1 OF 1

IR-LINK



 <p>MICROMINI, INC. 1000 STREET VERNON, CT 06066</p>		REV	1.1
		DATE	4/1/92
COHM-LINK		SHEET	2 OF 2
B		DESIGNED BY	JEFF BACHLOCH

(MC) IR-LINK SCHEMATICS (3 of 3)



Silkscreen for the (MC)IR-LINK

PARTS LIST for the (MC)IR-LINK

DESIGNATION	PART#	DESCRIPTION
<u>Printed Circuit Board</u>		
PCB1		COMM-LINK PC Board
<u>Integrated Circuits</u>		
U1	8031	Microcontroller
U2	27256	EPROM for IR-LINK or MCIR-LINK
U3	6264	RAM for IR-LINK
or	62256 w/SS	RAM in smartsocket for MCIR-LINK
U4	74LS373	Octal Latch
U5	CD4047	Monostable Multivibrator
U6	74LS00	Quad 2-Input NAND
U7	MAX232	Dual Transmitter/Receiver
U8	75176	Differential Transmitter/Receiver
U9	7407	Hex Buffer w/OC Outputs
<u>Resistors</u>		
R1, R3, R4, R5	10K	1/4W, 5% (brn-blk-org)
R2	4.7K	1/4W, 5% (yel-vio-red)
R6	100Ω	1/2W, 5% (brn-blk-brn)
R7 & R8	1K	1/4W, 5% (brn-blk-red)
R9	330Ω	1/4W, 5% (org-org-brn)
R10 & R18	240Ω	1/4W, 5% (red-yel-brn)
R11	7.5K	1/4W, 5% (gry-red-red)
R12	1K	1/4W, 5% (brn-blk-red)
R13	150Ω	1/4W, 5% (brn-grn-brn)
R14	22Ω	1/4W, 5% (red-red-blk)
R15	0Ω	1/4W, 5% (blk) or bus wire
<u>Capacitors</u>		
C1, C4, C5, C6, C7 & C9	10μF	Tantalum
C2 & C3	27pF	Monolithic
C8	470μF	Electrolytic
C10 & C12-C18	0.1μF	Monolithic
C11	390pF	Polystyrene

PARTS LIST for the (MC)IR-LINK (continued)

DESIGNATION	PART#	DESCRIPTION
<u>Semiconductors</u>		
D1-D3	1N4148	Small Signal Diode
D4	1N5402	Diode
Q1	LM7805	5.0-volt Linear Regulator
Q2	LM317LZ	Adjustable Linear Regulator
Q3	2N2907	PNP Transistor
Q5	IS1U60	IR Receiver
LED1	LD123	Dual IR LED
LED2 & LED5	TIL220	Visible LED

Connectors

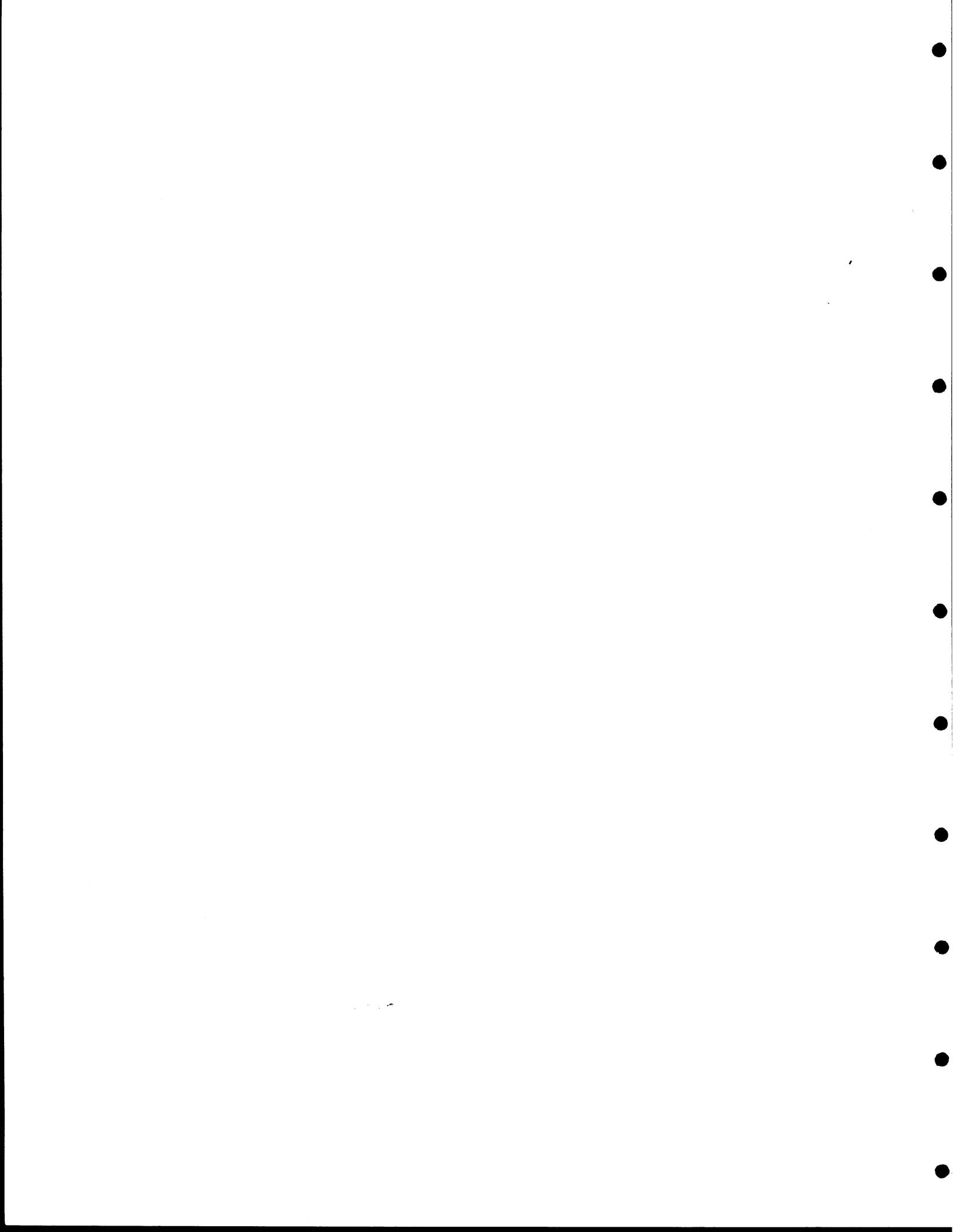
J2	2x8	Square-pin Header
JP1, JP2, JP5	2x3	Square-pin Header
JP3, JP4, JP6-JP7	1x2	Square-pin Header

Sockets

SK1	40-Pin	0.6-inch IC-socket
SK2 & SK3	28-Pin	0.6-inch IC-socket
SK4	20-Pin	0.3-inch IC-socket
SK5-SK6	14-Pin	0.3-inch IC-socket
SK9		
SK7	16-Pin	0.3-inch IC-socket
SK8	8-Pin	0.3-inch IC-socket

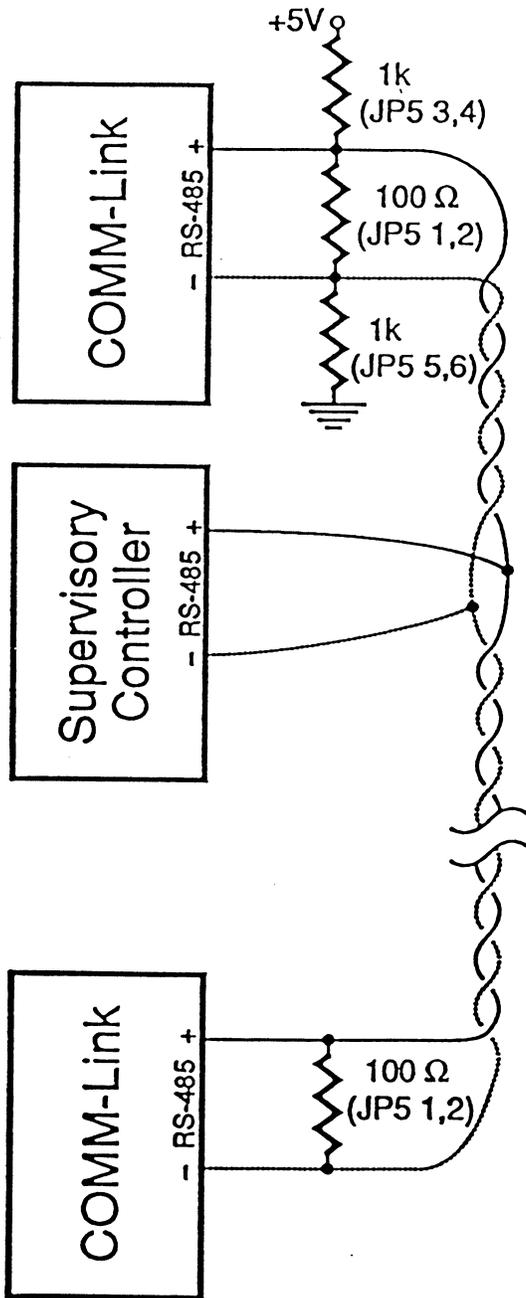
Miscellaneous

HS1		Heat Sink w/4-40 screw and nut
p1	5k	Potentiometer
SJ1-SJ9		Shorting Jumper
T1, T2	1x2	Screw Terminal Block
X1	11.059MHz	Crystal



Update

HCS II Network Termination Guidelines



In order for your HCS II network to operate as error-free as possible, the network cable must be properly terminated. The current documentation makes some references to termination, but needs clarifying. (If your system consists of only a Supervisory Controller and no Link modules, you don't have to worry about the network or termination.)

The network cable must be terminated in its characteristic impedance at each end. You should install the 100- Ω terminator resistor on the two modules at the ends of the cable. The resistor is installed on a COMM-Link board by inserting a jumper on JP5 pins 1 & 2. It is installed on the SC by inserting a jumper on JP1.

There should never be more than two 100- Ω resistors installed on the network!

In order to ensure the network floats to an inactive state when not being driven by any network module, a single pull-up resistor and a single pull-down resistor must be installed on the network. These resistors may be found on any PL-Link, IR-Link, LCD-Link, or DIO-Link module and are installed by inserting jumpers on JP5 pins 3 & 4 and 5 & 6. Early COMM-Link boards were shipped with 100k resistors installed in R7 and R8. These should be changed to 1k resistors. All current COMM-Link boards are shipped with 1k resistors.

There should never be more than one pull-up resistor and one pull-down resistor installed on the network!

Finally, as stressed in the manuals, twisted-pair cable should always be used for network connections.

Circuit Cellar, Inc.
4 Park St.
Vernon, CT 06066
(203) 875-2751

