VMEbus Compatible

16-Line Asynchronous Communications Multiplexor

Product P/N 221200

Manual P/N 340001 Rev. D

Revision Date: August 27, 1998

Includes:
Installation Instructions
Programming Specification

Macrolink, Inc. 1500 North Kellogg Drive Anaheim, California 92807
Phone (714)777-8800 — FAX (714)777-8807
www.macrolink.com — info@macrolink.com

Table of Contents

Chapter 1 - Installation

Chapter 2 - Soft Configuration Options

Chapter 3 - Programming

CHAPTER A - Interface Signals

This page has no technical content.

Installation

The Macrolink MVC 16-line Communications multiplexor is the first VMEbus async commux designed for real performance. At the heart of the MVC is a 16 MIPS AMD 29K RISC processor combined with an efficient shared memory and DMA architecture. The RISC handles all character processing and buffering so driver calls and CPU intervention are minimized, and data overruns are eliminated. Unlike traditional controllers that rely heavily on the host for processing or 68XXX-based controllers that can't meet high throughput requirements, the MVC has the required processing muscle thanks to the AMD 29K. All VME data transfers and bus handshakes are controlled by a VIC068 VLSI ASIC. The MVC provides outstanding flexibility and one of the most comprehensive set of features available today – all at a very affordable price.

The MVC supports 16 serial lines on a single 6U VMEbus module. All lines are independently programmable for speeds up to 38.4 Kbps with split speed support, with overall throughput exceeding 61,000 characters per second. That's more than double the rate of most other VMEbus asynchronous controllers. Full modem control is available and individually programmable on each line. All modem signals are available on all lines with no performance penalty.

The MVC is ideal for applications with large amounts of high-speed data traffic including graphics, supervisory/robotics and PC-to-host file transfers as well as interactive terminal communications.

Our 4-port line modules allow the MVC to handle multiple line disciplines. A single MVC can be configured with any mix of RS-232, RS-422, RS-423, RS-485, or MIL-STD-188C in 4-line groups. On-board memory is modular and expandable to 1MB. Our approach ensures that the MVC will handle today's communications requirements and meet your future application needs.

The MVC includes advanced features that benefit both the system integrator and programmer. All VMEbus and port parameters are soft-configured and can be defined on a line-by-line basis. The bus address is selectable with rotary switches located on the front bezel. Once installed, the MVC never needs to be removed, regardless of how often your application or configuration changes.

1.1 Product Features

Software Definable Features

- Baud rate, character length, parity, and stop bits.
- Transmit/receive line buffer size.
- Host memory DMA buffer size.
- Flow control: XON/XOFF, XOFF/any, CTS/RTS, or special characters.
- Control over all modem lines.
- Special handling for user defined characters.
- Send or receive break characters.

- Break character timing.
- Error status data and handling.
- Special characters for end of transmission or end of buffer with optional interrupts.
- VMEbus interrupt/request level.
- VMEbus address modifiers.
- Character (PIO) or DMA block modes with scatter/gather support.

Other Features

- Front panel LEDs indicate serial I/O activity and board status.
- Built in Centronics-compatible parallel printer port interfaces through the VMEbus P2 connector.

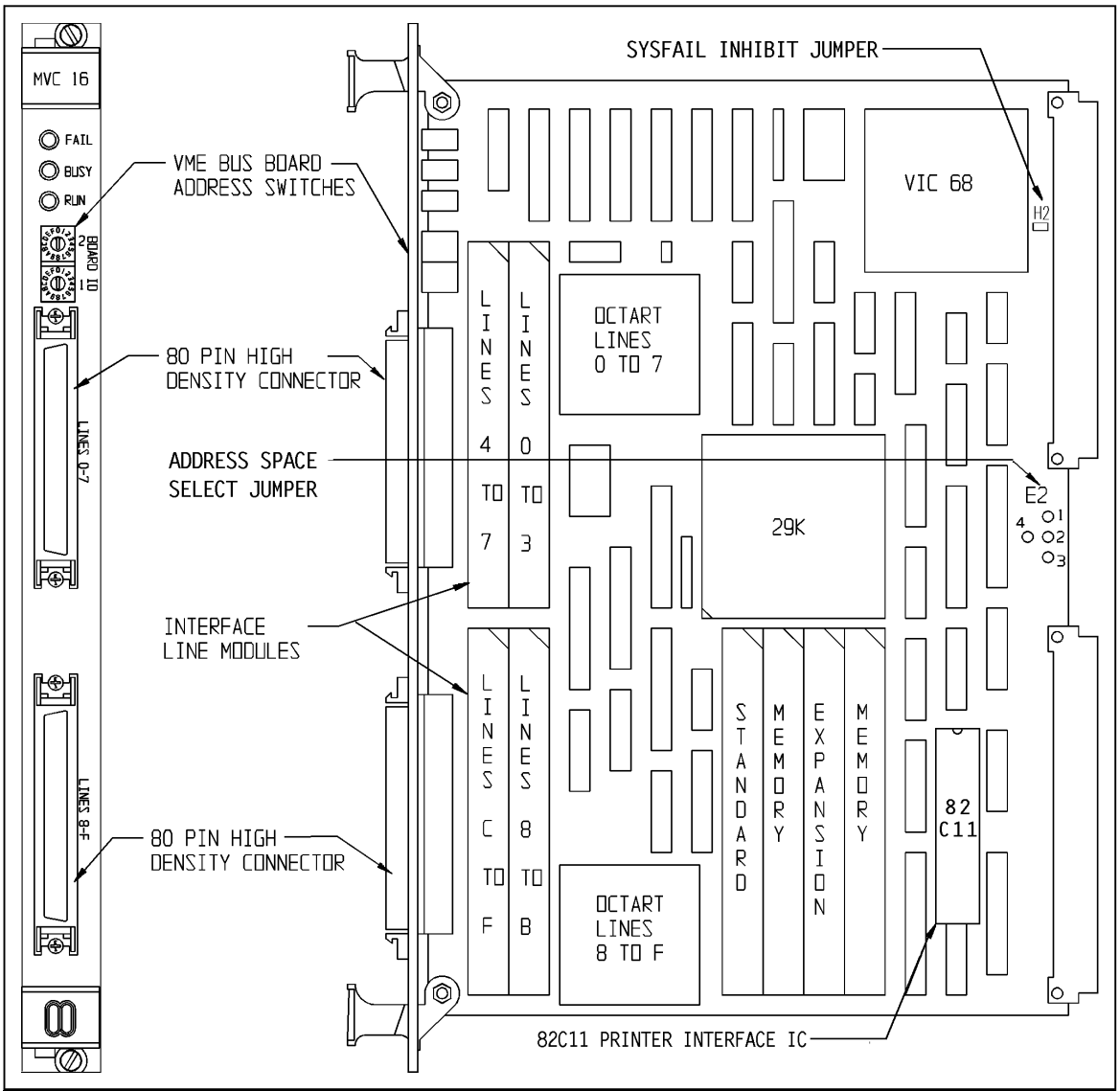


Figure 1 - 1 MVC Layout

1.2 Unpacking And Inspection

Each MVC is shipped in a sturdy, reusable padded carton. Optional accessories will be shipped in a separate carton and include I/O interface panels, interface cables, and an Installation and Programming manual. The shipping carton and packaging material should be retained for future use in moving, shipping, or storage of the board and accessories.

Carefully inspect the shipping carton for signs of damage. If any shipping damage is evident, do not open the package. Notify Macrolink and the freight carrier immediately to receive further instructions.

☞ The MVC uses sensitive electronic circuitry that can be easily damaged by electrostatic discharge. Personnel handling the board must exercise proper static control methods. Open the shipping carton only at an approved static-controlled workstation. An anti-static bag, anti-static bin, or the original packaging material must be used when transporting the board.

Open the shipping carton and remove the anti-static bag containing the MVC. The board should be removed from the bag carefully and inspected for damage prior to installation. Claims for shortage or damage must be filed within seven days of receipt of the shipment.

1.3 Installation

MVC Configuration

Rotary switches are used to set the most significant byte of the board's VMEbus base address. The BOARD ID 2 switch selects the most significant address digit. The BOARD ID 1 switch selects the second most significant digit. The address space (A32, A24, A16) is selected by jumper E2. It and the other MVC configuration jumpers are described in a table on the next page.

On power up, the shared RAM area of the MVC appears on the system memory map in the extended addressing area (Address modifier 0D or 09). If the system supports extended addressing, the MVC must be assigned a base address in the extended addressing area. If the system supports only short or standard addressing, the MVC may be assigned an address in those areas. Until the boot process has been completed, the board will *only* respond to an address modifier of 0D or 09, and a data size of 16 bits. If you are using the board in the short or standard addressing mode, be sure that no other boards in the system share the same extended mode address as the MVC.

Refer to Figure 1-1 to locate the Board ID switches and configuration jumpers.

MVC Jumpers	
Jumper	**Function**
H1	Inhibit SYSFAIL. In (default) = Normal operation. Out = Special application. The generation and reception of the VMEbus SYSFAIL signal is inhibited on the board.
E1	Memory Size. 1-2 = 64K modules installed. 2-3 = 16K modules installed. Factory configured. Do not alter the setting of this jumper.
E2	Address space. 2-4 (default) = A32 space. 3-4 = A24 space. 1-4 = A16 space. This jumper is in etch.

Extended mode addressing (32 bit)

In the extended addressing mode, address bits 24 thru 31 are used to select the MVC. For example, if the BOARD ID 2 switch is set to "D" and the BOARD ID 1 switch is set to "0," the board's base address will be 0xD0000000. This is the factory default configuration.

Standard mode addressing (24 bit)

In the standard addressing mode, address bits 16 thru 23 are used to select the MVC. For example, if the BOARD ID switches are set to "7" and "A" respectively, the board's base address will be 0x7A0000. To select this mode, cut the etch between E2 pads 2 and 4 on the solder side of the MVC, and install a jumper from pad 3 to pad 4. Refer to Figure 1-1 to locate jumper E2.

Short mode addressing (16 bits)

In the short addressing mode, address bits 9 thru 15 are used to select the MVC. For example, if the BOARD ID switches are set to "E" and "4" respectively, the board's base address will be 0xE400. Because the board requires 512 bytes of address space, BOARD ID 1 must be even (0,2,4,6,8,A,C, or E). To select this mode, cut the etch between E2 pads 2 and 4 on the solder side of the MVC, and install a jumper from pad 1 to pad 4. Refer to Figure 1-1 to locate jumper E2.

Addressing Examples		
Board ID	**Jumper E2**	**VMEbus Base Address**
D0	2 - 4	0xD0000000
7A	3 - 4	0x7A0000
E4	1 - 4	0xE400

Board Installation

After configuring the MVC to match your system, select a slot in which to install the board. With the power off, insert the MVC into the chassis. Tighten the board lock down screws to secure the board and guarantee proper RF shielding. If your backplane uses Bus Request and IACK jumpers, remove them from the card slot in which the MVC is installed. Be sure that the BR0-3 and IACK switches or jumpers are closed or installed for slots on your backplane which do not have boards installed. The system will not operate properly if there is an open in the Bus Request or IACK chain.

Cabling

Macrolink's I/O connector panels are fully enclosed FCC-compliant shielded units and may be mounted in the system, in an external I/O cabinet, or used stand alone. They are available in the following configurations:

- An 8-Line connector panel containing eight DB-25 connectors, Macrolink P/N 330061. See Figure 1-2. This panel requires a dual width 6U height I/O panel location. Two panels are required for 16-line installations.
- A 16-Line connector panel containing sixteen 10-pin RJ45 modular phone-style jacks, Macrolink P/N 330050. See Figure 1-3. This panel requires a triple width 6U height I/O panel location.

The MVC may cause interference to radio and television reception. To prevent this interference, FCC-compliant shielded cables, Macrolink P/N 320126, are recommended for use in all installations. The standard cable length is 6 feet. Cables in other lengths are available. An unshielded 3 foot ribbon cable, Macrolink part number 320138, is also available.

☞ The pins used on the connectors are delicate. Be sure the connectors are properly aligned when installing the cables to avoid bending or breaking the pins. NEVER FORCE THE CONNECTORS TOGETHER.

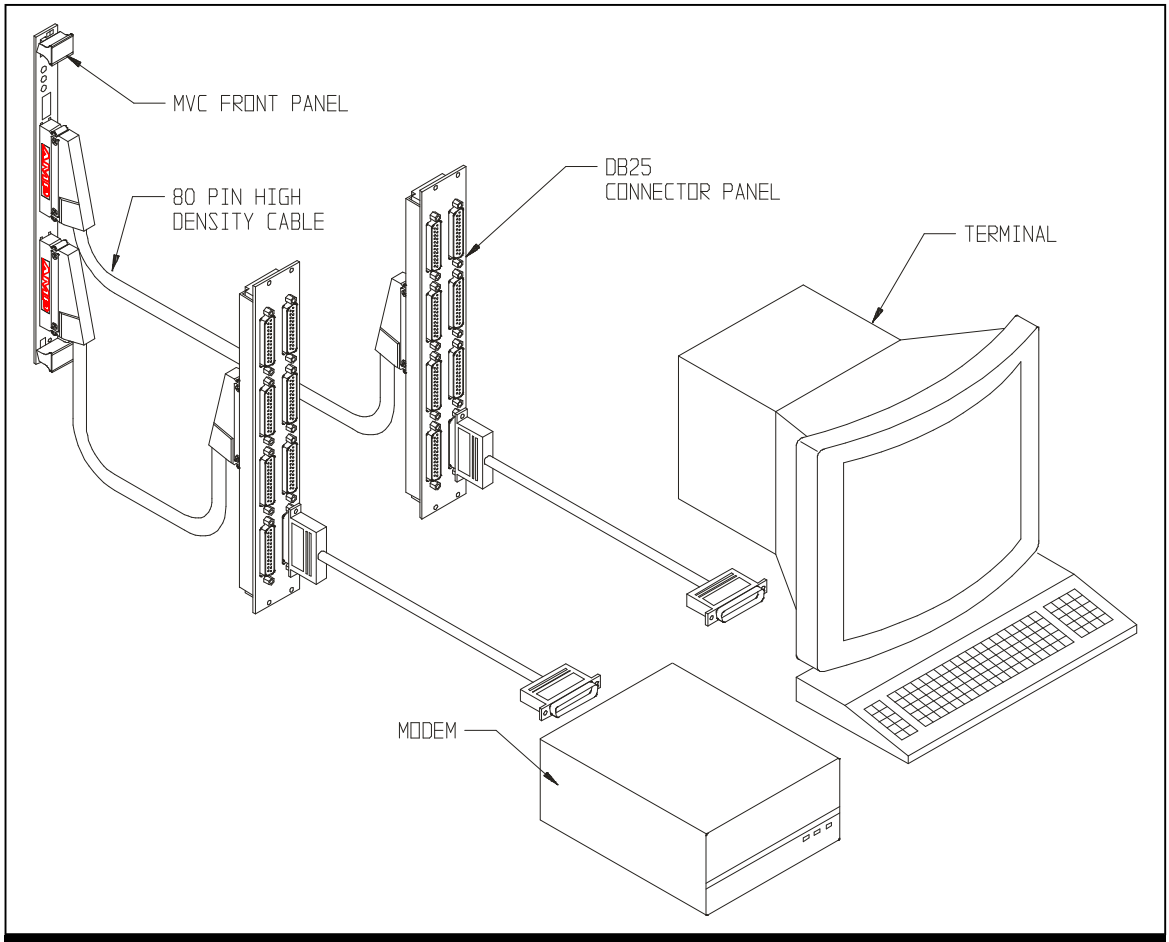


Figure 1 - 2 DB 25 Connector Panel Cabling

The cable connectors are keyed to prevent improper installation. The cable exits the connector shells at a right angle to allow for more orderly routing. Connect one end of the cable to the Port 0-7 connector of the MVC. Proper orientation of the connector is with the cable exiting the shell towards the bottom of the board. Connect the other end of the cable to the transition panel. Install the cable for ports 8-F in a similar manner.

Important

Unused modem status input signals on the MVC must either be terminated (tied to ground) or disabled in the line parameter #47 Valid Modem Input Mask (see Chapter 2 -* Programming*). Failure to do so will cause spurious interrupts to be generated, potentially causing a system crash. This problem occurs most frequently when unshielded flat ribbon cables are used.

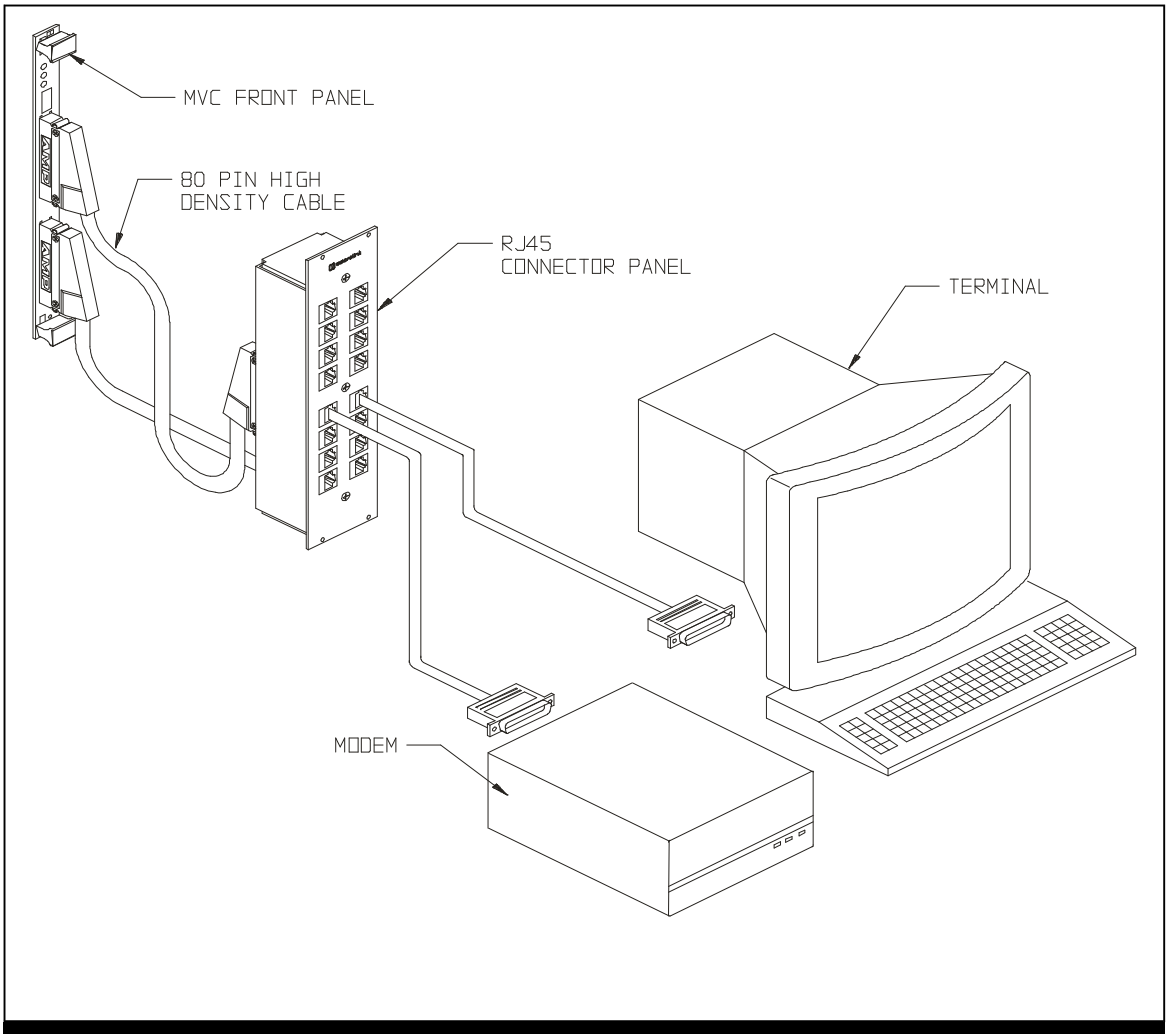


Figure 1 - 3 RJ45 Connector Panel Cabling

1.4 P2 Printer Adapter

A Centronics-compatible parallel printer is built in to the MVC. The interface to the printer port is via the VMEbus P2 backplane connector. An optional interface cable is available that connects between the backplane and an I/O panel equipped with a 36-pin Centronics-style connector. Order Macrolink part number 330042. The I/O panel is illustrated in Figure 1-4 below.

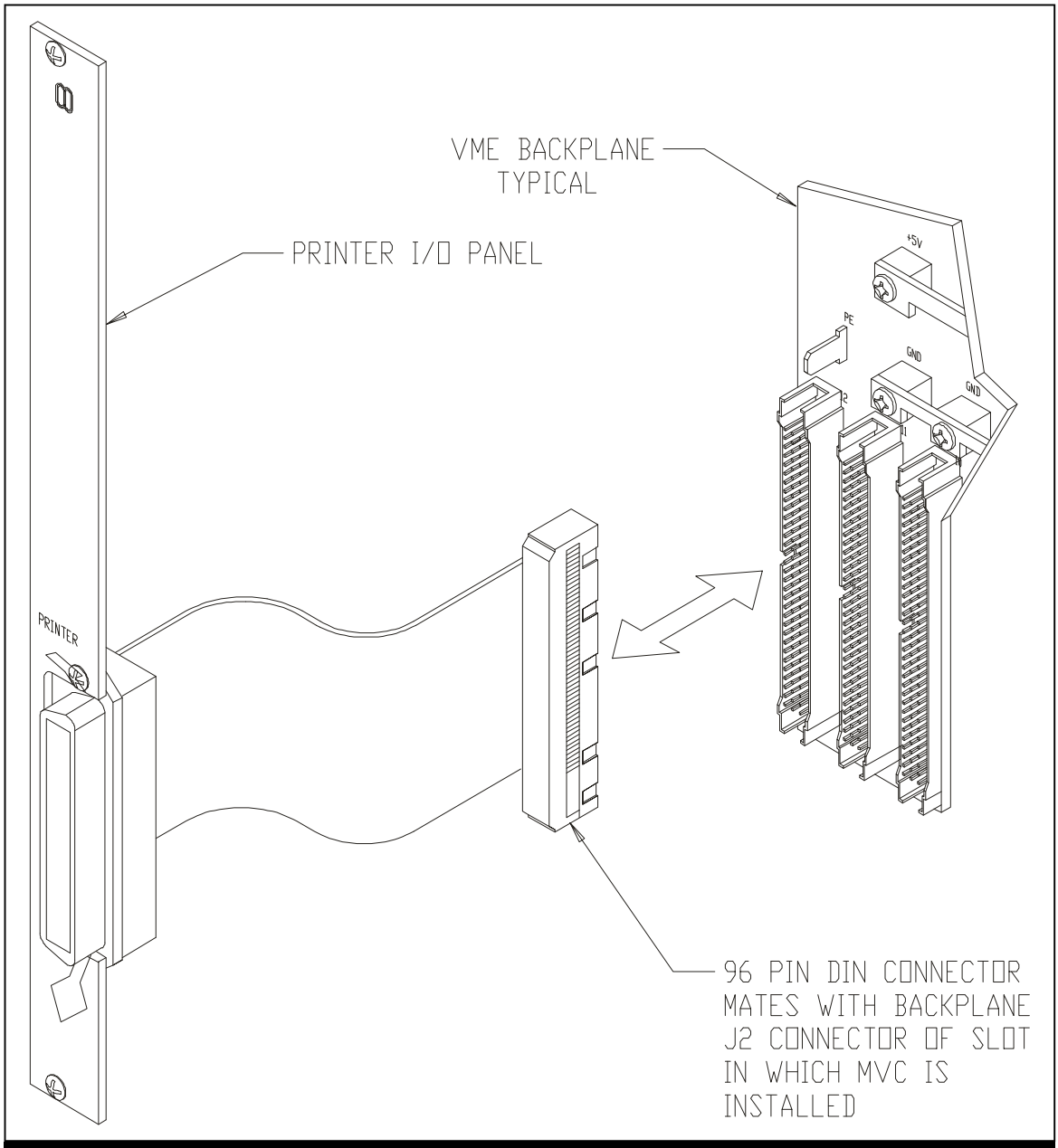


Figure 1 - 4 Printer Port Connector Panel

Disabling The Printer Port

Most systems do not make use of the P2 backplane connector for system specific functions. For those which do, the printer option cannot be utilized. In these instances, or when the printer port needs to be disabled, the following steps should be taken:

☞ The MVC contains circuitry which is extremely sensitive to electrostatic discharge. This procedure should be performed only by trained personnel utilizing proper static control procedures.

- Remove the board from the system. Locate and remove the 82C11 printer controller integrated circuit. See Figure 1-1.
- Disable the Printer Port Available flag, Controller Parameter #11. See Chapter 2 - *Programming* for more information. Disabling this flag will prevent the 29K processor from attempting to service the non-existent printer port.
- The internal FIFO buffer space which was previously allocated to the printer port should be distributed among the remaining lines on the MVC.

1.5 Driver Software

Specific instructions for installing the MVC driver software are contained in a README file included in the driver distribution. The driver software is available on several different media types and in a variety of data formats. Contact Macrolink for information on support for your platform.

The MVC driver distribution contains a hierarchical directory tree of source code files that have been converted to a single file by an archiving tool. Some archive formats also contain command scripts that run automatically when the distribution is installed. In all distribution formats, the README file can be accessed when the source files are loaded, without building the driver. Please read the README file before building the host adapter driver.

How To Find The README File

"TAR" format distributions contain pathnames that are relative to the current directory. They can be loaded into any directory. The README file is found in the top level directory of the archive. Note: the filename may be lowercase (*readme*) in some driver package releases. The file can be extracted separately into the current directory by running the following command:

```
tar xvf /dev/rfd0 README
```

This example assumes that /dev/rfd0 is the device file for the drive that contains the distribution media. Your device file name may vary.

"PKG" format distributions contain pathnames that are relative to a "base directory" instead of the current directory. The MVC driver's base directory under Solaris 2™ is /opt/MLINKmvac/src. "PKG" format is called "datastream" format on some platforms. To access the README file, the distribution files must be installed into the base directory without building the driver from the installed source files. For example, the following command would install the distribution source files into the base directory:

```
pkgadd -d /dev/rmt/0 MLINKmvac
```

The README file would be installed into the base directory, /opt/MLINKmvac/src. This example assumes that /dev/rmt/0 is the device file for the drive that contains the distribution media. Your device file name may vary.

1.6 Returning Products For Repair

Before returning a product for repair, you must obtain a Return Material Authorization (RMA) number from Macrolink. This number must appear on the outside of the shipping container. Products returned without an RMA number will be refused. When requesting an RMA number, please be prepared to provide the following information:

- Product name or part number.
- Serial number.
- Failure description. An inspection fee may be assessed on items returned without an adequate failure description.

Shipping expenses to Macrolink are to be paid by the customer. Macrolink will pay for standard return shipping for products under warranty.

1.7 FCC Compliance

This equipment generates, uses, and can radiate radio frequency energy, and if not installed and used in accordance with this manual, may cause interference to radio and television reception. This equipment has been tested and found to comply with the limits for a Class A computing device in accordance with the specifications in Subpart J of Part 15 of the FCC Rules, which are designed to provide reasonable protection against such interference when operated in a commercial environment. Operation of this equipment in a residential area is likely to cause interference requiring the user, at the user's own expense, to take whatever measures may be necessary to correct the interference.

If this equipment does cause interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference in one or more of the following ways:

1. Reorient the receiving antenna.
2. Relocate the computer with respect to the receiver.
3. Move the computer away from the receiver.
4. Plug the computer into a different outlet, so that the computer and receiver are on different circuit branches.
5. Ensure that the mounting screws, attachment connector screws, and grounding connections are securely tightened.
6. Ensure that good quality, shielded and grounded cables are used for data communications.

If necessary, the user should consult the dealer or an experienced radio/television technician for additional suggestions. The user may find the booklet *How To Identify And Resolve Radio-TV Interference Problems* prepared by the Federal Communications Commission helpful. This booklet is available from the U.S. Government Printing Office, Washington, D.C. 20420, Stock No. 004-000-0345-4.

1.8 Specifications

Specifications	
VMEbus	Compliant with the VMEbus revision C.1 specification IEC 821 & IEEE P1014/01.2. Supports both Master & Slave modes.
Master Data Transfer	A32/A24/A16 - D32/D16/D8. Programmable address modifiers.
Slave Data Transfer	A32/A24/A16 - D32/D16/D08 A16 Supervisory access (2D). A24 Supervisory data access (3D). A32 Supervisory data access (0D). A16 non-privileged access (29). A24 non-privileged data access (39). A32 non-privileged data access (09).)
Bus Request Lines	BR(0) through BR(3). Programmable selection of all lines.
Interrupt Request	IRQ1 through IRQ7. Programmable selection of all lines.
Channel Interface	8 or 16 channels, asynchronous serial data. 38.4Kbaud sustained on all 16 lines concurrently. RS232C/CCITT V.24/V.28, RS422/CCITT V.11, RS423, RS485, and/or MIL 188C. All lines independently programmable. Programmable line/controller parameters. Support for both hardware and software flow control. Modem control on all lines supported via CTS, DCD, DSR, RTS, DTR & RI.
Dimensions	6U Dual-height Eurocard (160mm x 233mm) Front Panel - Single width, dual-height VME panel (20mmx262mm)
Connectors	Front Panel: One 80-pin, shielded high-density connector for each 8-channel group. Backplane: P1 & P2 standard 96-pin DIN connectors.
Power	5VDC @ 5.0A (maximum) +12VDC @ .1A (maximum) –12VDC @ .075A (maximum)
Temperature	0° to 50°C (32° to 120° F) operating. –40° to 68° C (0° to 150° F) storage.
Humidity	10% to 95%, non-condensing.
Certification	FCC Part 15 Class A
Reliability (MTBF)	8 line: 148,281 P.O.H. per MIL-HDBK-217E 16 line: 147,608 P.O.H. per MIL-HDBK-217E

Soft Configuration Options

This chapter contains information for configuring the MVC. When the system power is turned on or after the VMEbus RESET signal has been activated, the MVC is in boot mode. The system first downloads the operational firmware to the board, followed by the configuration parameter blocks. The default parameters are stored in a configuration file on the system. The name of this file will vary depending upon the operating system in use. Consult the README file included with the software driver files for specific information. Each serial line has its own set of configuration parameters. There is also a block that defines options that affect the whole controller. These parameters are described in this chapter.

Macrolink supplies STREAMS drivers that support *stty* commands to change the baud rate and similar parameters while the board is operating. Those writing their own driver will find information on changing parameters under program control in Chapter 3 - *Programming*.

2.1 Controller Parameter Block

The controller parameter block is a set of sixteen 32-bit words that define controller related options. The parameters shown in the table below are defined in the sections that follow.

Controller Parameters

CP Number	Description
0	Baud Rate Group Flag for Lines 0 through 7
1	Baud Rate Group Flag for Lines 8 Through F
2	Receive Interrupt Level
3	Receive Interrupt Vector
4	Transmit Interrupt Level
5	Transmit Interrupt Vector
6	Bus Request Level
7	Slave Mode Address Size
8	Bus Release Control
9	Reserved
10	Reserved
11	Printer Enable
12	Receive DMA And Break Timebase Controller
13-15	Reserved

The tables that are included in the descriptions of the parameters are guides that show which bits in the longword are used to specify the parameter value. All undefined bits of the controller parameter words *must* be zero.

Baud Rate Group Flag For Lines 0 Through 7 (CP #0)

This parameter specifies the group of available baud rates for lines zero through seven. Each line may set to any of the 13 baud rates specified in the selected group, independently of the rates specified for the other lines.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																								G							

Valid arguments for this parameter are:

0x00 - Select group zero rates:
50, 110, 134.5, 200, 300, 600, 1200, 1050, 2400, 4800, 7200, 9600, 38.4K

0x80 - Select group one rates:
75, 110, 150, 300, 600, 1200, 1800, 2000, 2400, 4800, 9600, 19.2K, 38.4K

Baud Rate Group Flag For Lines 8 Through F (CP #1)

This parameter specifies the group of available baud rates for lines eight through fifteen. Each line may set to any of the 13 baud rates specified in the selected group, independently of the rates specified for the other lines. This parameter must be set to zero if the MVC has only 8 lines installed.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																								G							

Valid arguments for this parameter are:

0x00 - Select group zero rates:
50, 110, 134.5, 200, 300, 600, 1200, 1050, 2400, 4800, 7200, 9600, 38.4K

0x80 - Select group one rates:
75, 110, 150, 300, 600, 1200, 1800, 2000, 2400, 4800, 9600, 19.2K, 38.4K

Interrupt Levels and Vector Numbers (CP 2-5)

It is recommended to use the same interrupt request level and vector for transmit and receive interrupts to simplify handling of the line status word. If separate interrupt vector numbers will be used, different interrupt vector levels must also be used, with receive interrupts having a higher priority than transmit interrupts. Examples of valid and invalid interrupt level/vector number combinations are shown below:

Valid interrupt level and vector number examples:

Example 1:

Receive Interrupt Level (CP 2) = 4
Transmit Interrupt Level (CP 4) = 4
Receive Vector Number (CP 3) = 0x73
Transmit Vector Number (CP 5) = 0x73

Example 2:

Receive Interrupt Level (CP 2) = 4
Transmit Interrupt Level (CP 4) = 3
Receive Vector Number (CP 3) = 0x74
Transmit Vector Number (CP 5) = 0x73

Invalid interrupt level and vector number examples:

Example 1:

Receive Interrupt Level (CP 2) = 3
Transmit Interrupt Level (CP 4) = 4
Receive Vector Number (CP 3) = 0x73
Transmit Vector Number (CP 5) = 0x73

Example 2:

Receive Interrupt Level (CP 2) = 4
Transmit Interrupt Level (CP 4) = 4
Receive Vector Number (CP 3) = 0x74
Transmit Vector Number (CP 5) = 0x73

When separate levels and vector numbers are used, the driver transmit interrupt function should only clear Transmit Buffer Empty (TBE) in the corresponding status register, and the receive interrupt function should clear all bits except TBE. A Read-Modify-Write instruction should be used to clear the status register bits or disable interrupts to avoid status register corruption by a higher priority interrupt.

The only interrupt status that uses the transmit interrupt vector and transmit interrupt mask is Transmit Buffer Empty (TBE). All other interrupt statuses, including TFWE, use the receive interrupt vector and receive interrupt mask.

Refer to Chapter 3 - *Programming* for further information.

Receive Interrupt Level (CP #2)

This parameter defines the interrupt level the controller will use to indicate a receive data or receive error interrupt. Valid interrupt levels are 0 through 7. A zero interrupt level disallows receive interrupts. This value does not have to, but can be the same value used for transmit interrupts. The receive interrupt level is contained in bits 1 through 7. Bit 0 *must* be set (1).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																								Level							1

Valid arguments are:

0x01 - Interrupt level 0 (no interrupts)
0x03 - Interrupt level 1
0x05 - Interrupt level 2
0x09 - Interrupt level 3
0x11 - Interrupt level 4
0x21 - Interrupt level 5
0x41 - Interrupt level 6
0x81 - Interrupt level 7

Receive Interrupt Vector (CP #3)

The receive interrupt vector number is the value the controller places on the bus during an interrupt acknowledge cycle. The host system normally uses this value to determine which device issued the interrupt. This value does not have to, but can be the same value as used for transmit interrupts.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																								Vector							

Transmit Interrupt Level (CP #4)

This parameter defines the interrupt level the controller will use to indicate a transmit complete or transmit error interrupt. Valid interrupt levels are 0 through 7. A zero interrupt level disallows transmit interrupts. This value does not have to, but can be the same value used for receive interrupts. The transmit interrupt level is contained in bits 1 through 7. Bit 0 *must* be set (1).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																								Level							1

Valid arguments are:

0x01 - Interrupt level 0 (no interrupts)
0x03 - Interrupt level 1
0x05 - Interrupt level 2
0x09 - Interrupt level 3
0x11 - Interrupt level 4
0x21 - Interrupt level 5
0x41 - Interrupt level 6
0x81 - Interrupt level 7

Transmit Interrupt Vector (CP #5)

The transmit interrupt vector number is the value the controller places on the bus during an interrupt acknowledge cycle. The host system normally uses this value to determine which device issued the interrupt. This value does not have to, but can be the same value as used for receive interrupts.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																								Vector							

Bus Request Level (CP #6)

This parameter specifies the bus request level to be used by the controller for DMA transfers to and from the VME bus.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																									level						

Valid bus request levels are:

0x00 - Level 0 (BR0)
0x20 - Level 1 (BR1)
0x40 - Level 2 (BR2)
0x60 - Level 3 (BR3)

☞ Some systems refer to the interrupt level as the “maximum bus request level.” This is a mistake in terminology as the interrupt and bus levels are two distinctly different values. The bus request level gives the requester a priority level for gaining bus access. The interrupt level gives the requester a priority for performing processor interrupts.

Note that many systems do not use all of the levels. The bus request level is usually set to three to give all devices equal chance at gaining ownership of the bus. If all devices have equal priority, proximity to the CPU/bus master determines bus request priorities. Slower devices should be placed closer to the CPU. They require less bus time, but should be given priority when they need access.

Slave Mode Address Size (CP #7)

This parameter defines the mode used to access the MVC's shared RAM during slave transfers. The default size is A32 (extended). This change goes into effect after the last line parameter has been transferred to the MVC during Initialization mode.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																											1	spc		0	1

Valid arguments are:

0x11 - extended space (A32)
0x15 - standard space (A24)
0x19 - short space (A16)
0x1D - invalid

☞ Changing the address space of the controller can have a major impact on system operations. The MVC will be communicating in one addressing mode, and then it will move to the new space, with no set time for this change to take effect. It is suggested that if the address space is to be changed, no accesses to the old or new address should be performed for at least 1 second after transferring the last line parameter to the MVC.

Bus Release Control (CP #8)

This parameter specifies the method used for allowing other devices to arbitrate for the bus. It is used when performing bursts with block mode DMA capable memory - see *Block Mode and Burst Count* (CP #9) below. The modifiers give the controller a fairness parameter when performing burst transfers. Bits 0-5 of the parameter word *must* be set (1).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																								mod			1	1	1	1	1	1

Valid arguments are:

0x3F - Release on request (ROR)
0x7F - Release when done (RWD)
0xBF - Release on BCLR (ROC)
0xFF - invalid

Controller Parameter #9 (Reserved)

This parameter is reserved for future use. It must be set to zero.

Controller Parameter #10 (Reserved)

This parameter is reserved for future use. It must be set to zero.

Printer Port Enable (CP #11)

This parameter must be disabled on boards not equipped with the printer port option, or for installations that will not use the printer port. This will allow the buffer space that was allocated to the printer to be distributed among the serial I/O lines.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																															P

Valid arguments are:

0x00 - Disable printer
0x01 - Enable printer

Receive DMA And Break Timebase Controller (CP #12)

This parameter allows selection of the timebase to be used for the DMA Break Timeout (LP #44) and Receive Timeout (LP #70) counters.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																														TB	

Valid parameters are:

0 1/20 of a second.
1 1/50 of a second.
2 1/100 of a second.

The firmware adds one tick to the specified interval to avoid the possibility of a short timeout when a timeout of 1 is specified. The timer could take from N to N + 1 ticks to expire. On average, a randomly started N tick timeout should expire in N + .5 ticks.

Controller Parameters 13-15

These parameters are reserved for future use. They must be set to zero.

2.2 Line Parameter Block

Each serial line is configured through its own line parameter block. The line parameter block is a set of one hundred and twenty-eight 32-bit parameter words. All undefined bits of the line parameter words *must* be zero. A total of sixteen blocks are written to the MVC, regardless of the number of lines installed. Parameter blocks for unused lines should contain all zeroes. The parameters and their definitions are described in the sections that follow.

Parameters marked with an asterisk (*) can be changed while the board is operating. See Chapter 3 - *Programming* for information on dynamically changing line parameters under program control.

Line Parameter Block	
LP Number	**Description**
0	Receive Baud Rate *
1	Transmit Baud Rate *
2	Character Length (Bits Per Character) *
3	Number Of Stop Bits (Length) *
4	Parity Flag *
5	Receive FIFO Buffer Size
6	Transmit FIFO Buffer Size
7	Auto Echo Flag *
8	Hardware Auto Flow Control Flag *
9	XOFF Character *
10	XON Character *
11	Number of Strip Characters
12 – 21	List of Strip Characters
22	Number of Receive Translate Characters
23 – 32	List Of Receive Translate Characters
33	Number of Transmit Translate Characters
34 – 43	List of Transmit Translate Characters
44	Line Break Timer
45	Modem Auto Answer Flag
46	Modem Half Duplex Control Flag
47	Valid Modem Input Mask
48	Valid Error Status Mask
49	Receive FIFO Almost Full Value
50	RS485 Flag
51	Printer Valid Status Mask
52	Line Type

Line Parameter Block	
LP Number	**Description**
53	Receive Mode Flag
54	Receive DMA Buffer Address
55	Receive DMA Buffer Length
56	Transmit Mode Flag
57	Transmit DMA Buffer Address
58	Transmit DMA Buffer Length
59	Number Of End Of Buffer Characters *
60 - 69	List of End Of Buffer Characters *
70	Receive Timeout Value *
71	Master Mode Address Modifier
72	Master Mode Data Length
73	Disable Receive Soft Flow Control *
74	Disable Transmit Soft Flow Control *
75	DMA Receive Timeout Block Mode *
76	Receive Characters Available Now *
77 - 127	Reserved For Future Use

Receive Baud Rate (LP #0)

This parameter specifies the default receive baud rate for the specified line. The group value is specified in Controller Parameter numbers 0 and 1. This parameter can be changed while the board is operating.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																												rate			

Valid arguments for the Receive Baud Rate field are:

	Group 0	Group 1
0x00	50	75
0x01	110	110
0x02	134.5	38.4K
0x03	200	150
0x04	300	300
0x05	600	600
0x06	1200	1200
0x07	1050	2000
0x08	2400	2400
0x09	4800	4800
0x0A	7200	1800
0x0B	9600	9600
0x0C	38.4K	19.2K

Transmit Baud Rate (LP #1)

This parameter specifies the default transmit baud rate for the specified line. The group value is specified in Controller Parameter numbers 0 and 1. This parameter can be changed while the board is operating.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																												rate			

Valid arguments for the Transmit Baud Rate field are:

	Group 0	Group 1
0x00	50	75
0x01	110	110
0x02	134.5	38.4K
0x03	200	150
0x04	300	300
0x05	600	600
0x06	1200	1200
0x07	1050	2000
0x08	2400	2400
0x09	4800	4800
0x0A	7200	1800
0x0B	9600	9600
0x0C	38.4K	19.2K

Character Length (LP #2)

This parameter specifies the default character length (bits per character) for the specified line. This parameter can be changed while the board is operating.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																														len	

Valid arguments for the Character Length field are:

0x00 - 5 bits
0x01 - 6 bits
0x02 - 7 bits
0x03 - 8 bits

Number of Stop Bits (LP #3)

This parameter specifies the default number of stop bits for the specified line. This parameter can be changed while the board is operating.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																												stop			

Valid arguments for the Number of Stop Bits field are:

0x07 - 1 stop bit
0x08 - 1.5 stop bits
0x0F - 2 stop bits

Parity Flag (LP #4)

This parameter specifies the default parity for the specified line. This parameter can be changed while the board is operating.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																													parity		

Valid arguments for the Parity Flag are:

0x00 - parity enabled, even parity
0x01 - parity enabled, odd parity
0x02 - force parity, even parity
0x03 - force parity, odd parity
0x04 - no parity

Receive (LP #5) and Transmit (LP #6) FIFO Buffer Size

These parameters determine the amount of the MVC's SRAM that is allocated to the specified line's receive and transmit FIFO buffers. The MVC can be equipped with a maximum of 1 megabyte of SRAM. The firmware and parameters that are downloaded to the board consume approximately 64K. The remainder is available for the FIFO buffers. The Memory Size register should be read to determine the total amount of memory available (see Chapter 3 - *Programming*). Each line should take a portion of the available memory.

Allocating too small a buffer will cause data starvation during transmits and data overruns on receives if flow control is disabled. Allocating more buffer space than needed has no adverse impact on the operation of the board. If more memory is allocated than is available, the firmware will first disable the printer port, then disable serial lines starting with the highest numbered line, until the allocated memory is less than or equal to the available memory.

Receive FIFO Buffer Size (LP #5)

The receive FIFO buffer consumes two bytes for each character received; one for the received character and one for the received character status. This parameter specifies the combined size, in bytes. The recommended minimum value is 2048 bytes, which equals 1024 characters and 1024 status bytes..

Transmit FIFO Buffer Size (LP #6)

This parameter specifies the number of bytes to be used for the specified line's transmit buffer. This value should be a minimum of 16 bytes. The recommended minimum value is 128 bytes.

Auto Echo Flag (LP #7)

This parameter instructs the controller to automatically echo data received to the transmit port for the specified line. Lines configured for RS485 interface may not use the Auto Echo option. This parameter can be changed while the board is operating by issuing a Change Line Parameter command.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																															E

Valid arguments for the Auto Echo Flag are:

0x00 - normal
0x01 - auto echo mode

Hardware Auto Flow Control Flag (LP #8)

This parameter instructs the MVC to control data transfers using the Request to Send and Clear to Send (RTS/CTS) hardware control signals. This parameter can be changed while the board is operating by issuing a Change Line Parameter command.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																															F

Valid arguments for the auto flow control flag are:

0x00 - do not use hardware flow control
0x01 - use hardware flow control

In hardware flow control mode, the MVC will discontinue transmitting data when it sees the Clear to Send (CTS) signal negate. It will resume transmitting when CTS is reasserted. If the MVC is busy (cannot accept a new character) it will negate Request to Send (RTS). The remote device will stop transmitting data until the controller reasserts RTS.

The CTS bit in the line status register will always report 0 while hardware flow control is active.

XOFF Character (LP #9)

This parameter defines the character to transmit to disable incoming data or recognize to stop transmitting data when using XON/XOFF software flow control. A value of zero disables software flow control. For lines configured as RS485, you may not use software flow control; the XOFF Character must be set to zero. This parameter can be changed while the board is operating by issuing a Change Line Parameter command.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																								character							

XON Character (LP #10)

This parameter defines the character to transmit to re-enable incoming data or recognize to re-enable transmit data when using software flow control. If this character is zero and the XOFF character is non-zero, then any character will be considered an XON character. This parameter can be changed while the board is operating by issuing a Change Line Parameter command.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																								character							

Number of Strip Characters (LP #11)

This feature is useful for ignoring unwanted characters, such as an escape or break character. When a strip character is encountered during transmit or receive, it is ignored. This parameter specifies how many characters are in the list of strip characters. There are ten available strip characters for each line.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																												count			

List of Strip Characters (LP #12 through 21)

This is the list of characters that, when encountered, will be stripped from the transmit and receive buffers of the specified line.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																								character							

Number of Receive Translate Characters (LP #22)

This parameter specifies the number of characters in the receive translate character list. When a character is received, it is compared to those in the translate list. If a match is found, the character will be translated to the new character before being placed in the receive buffer. This feature is useful when the received character may cause undesirable operation of the host system. For example, an escape (0x1B) can be translated to a null (0x00). There are ten available receive translate characters for each line.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																												count			

List of Receive Translate Characters (LP #23 through 32)

This is the list of receive translate characters for the specified line. “From” is the character to be translated, and “To” is the character to be substituted.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																to								from							

Number of Transmit Translate Characters (LP #33)

This parameter specifies the number of characters in the transmit translate character list. Before a character is transmitted, it is compared to those in the translate list. If a match is found, the character will be translated to the new character and then transmitted. There are ten available transmit translate characters for each line.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																												count			

List of Transmit Translate Characters (LP #34 through 43)

This is the list of transmit translate characters for the specified line. “From” is the character to be translated, and “To” is the character to be substituted.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																to								from							

Line Break Timer (LP #44)

This parameter tells the controller how many $1/20^{th}$ of a second counts to continue issuing a line break after receiving a Start Line Break command. The line break will stop when the timer expires. This is a 32-bit field.

Modem Auto Answer Flag (LP #45)

This flag, when enabled, causes the controller to set the Data Terminal Ready (DTR) signal when Ring Indicate (RI) is active.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																															A

Valid arguments for the Modem Auto Answer Flag are:

0x00 - do not set DTR on ring
0x01 - set DTR on ring

Modem Half-Duplex Control Flag (LP #46)

This flag specifies whether a connected modem is operated in full or half duplex mode. In half duplex mode, the controller will assert Request to Send (RTS) and wait for Clear to Send (CTS) before transmitting a character.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																															M

Valid arguments for the Modem Control Duplex Flag are:

0x00 - full duplex mode
0x01 - half duplex mode

This parameter can be changed while the board is operating by issuing a Change Line Parameter command. This change takes place immediately, even if there are characters being received. It is recommended that the line be idle (closed or flushed) when this parameter is changed.

Valid Modem Input Mask (LP #47)

This parameter specifies the set of valid modem status signals. These status bits will be displayed in the line status register. When a bit in this mask is cleared (0), the Line Status Register bit corresponding to the input line will also be cleared. For example, a value of 0x03 states that only the CTS and DSR lines will be active, and RI and DCD will be ignored. This parameter can be changed while the board is operating by issuing a Change Line Parameter command.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																												DCD	RI	CTS	DSR

The Valid Modem Input Mask is defined as follows:

Bit 0 (0x01) is Data Set Ready (DSR)
Bit 1 (0x02) is Clear To Send (CTS)
Bit 2 (0x04) is Ring Indicate (RI)
Bit 3 (0x08) is Data Carrier Detect (DCD)

Valid Error Status Mask (LP #48)

This parameter specifies the set of valid error status bits. When a bit in this mask is cleared (0), the Line Status Register bit corresponding to the error signal will also be cleared (see Line Status Register in Chapter 3 - *Programming*). This parameter can be changed while the board is operating by issuing a Change Line Parameter command. The input mask is defined as follows:

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																	RCAN	TFWE	BSTR					RBD	FRE	PE	OVR				

The Valid Error Status Mask is defined as follows:

0x0010 OVE is Overrun error
0x0020 PE is Parity error
0x0040 FRE is Framing error
0x0080 RBD is Receive break detect
0x1000 BSTR is Break started
0x2000 TFWE is Transmit FIFO Went Empty Status
0x4000 RCAN is Received Characters Available Now

Receive FIFO Almost Full Value (LP #49)

This parameter specifies the number of characters allowed in the receive FIFO before the RFAF status and interrupt are generated. If this value is zero, the RFAF status and interrupt are disabled. This value should be slightly less than the receive FIFO size. The recommended value is 16 less than the receive FIFO size.

RS485 Flag (LP #50)

This parameter specifies whether an RS485 line module is installed for the specified line. Lines configured as RS485 may not use the Auto Echo (LP #7) or XON/XOFF software flow control (LP #9) options.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																															F

Valid arguments for the RS485 Flag are:

0x00 - No RS485 module installed for this line
0x01 - An RS485 module is installed for this line

Printer Valid Status Mask (LP #51)

This parameter specifies the set of valid printer status bits. When a bit in this mask is cleared (0), the Line Status Register bit corresponding to the input signal will also be cleared.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																										POUT	OFFL	ERR	0	0	PSV

The Printer Valid Status Mask is defined as follows:

0x01 PSV is Printer Status Valid
0x08 ERR is Printer Error
0x10 OFFL is Printer Offline
0x20 POUT is Paper Out

Line Type (LP #52)

This parameter is used to inform the driver of the type of device connected to the specified line.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																															type

Valid arguments for the Line Type value are:

0x00 is a local terminal
0x01 is a local printer
0x02 is a terminal connected to a modem
0x03 is a printer connected to a modem

Receive Mode Flag (LP #53)

This parameter specifies whether the MVC should use character (PIO), sequential DMA, or scatter/gather DMA mode for receive data transfers on the specified line.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																															RM

Valid arguments for the Receive Mode Flag are:

0x00 - character (PIO) mode
0x01 - sequential DMA mode
0x02 - scatter/gather DMA mode

Receive DMA Buffer Address (LP #54)

This parameter specifies either the DMA buffer address in host memory or the address of the receive buffer scatter gather table. This is a 32-bit field. This value is ignored in character (PIO) mode.

Receive DMA Buffer Length (LP #55)

This parameter specifies the length of the software driver's receive buffer when the controller is in DMA mode. This is a 32-bit field. If a value of 0 or a value greater than 256 is specified, the driver will default to 256 bytes. This value is ignored in character (PIO) mode.

Transmit Mode Flag (LP #56)

This parameter specifies whether the MVC should use character (PIO), sequential DMA, or scatter/gather DMA mode for transmit data transfers on the specified line.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																															TM

Valid arguments for the Transmit Mode Flag are:

0x00 - character (PIO) mode
0x01 - sequential DMA mode
0x02 - scatter/gather DMA mode

Transmit DMA Buffer Address (LP #57)

This parameter specifies either the DMA buffer address in host memory or the address of the transmit buffer scatter gather table. This is a 32-bit field. This value is ignored in character (PIO) mode.

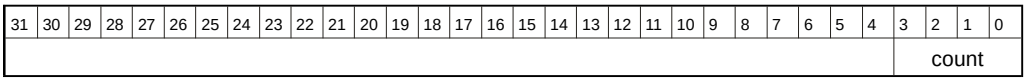
Transmit DMA Buffer Length (LP #58)

This parameter specifies the length of the software driver's transmit buffer when the controller is in DMA mode. This is a 32-bit field. If a value of 0 or a value greater than 256 is specified, the driver will default to 256 bytes. This value is ignored in character (PIO) mode.

Number of End of Buffer Characters (LP #59)

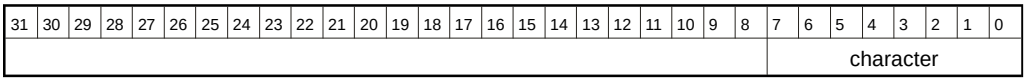
This parameter specifies how many characters are in the End Of Buffer Character list. These characters are used to mark the end of a DMA block. When the controller recognizes any of these characters, the number of characters actually received will be written to the receive count register, and the DMA cycle will be completed. There are a maximum of ten available end of buffer characters.

This parameter can be changed while the board is operating by issuing a Change Line Parameter command. This change takes place immediately, even if there are characters in the on-board receive FIFO. The characters in the FIFO will be compared against the new characters specified in the new count (which may be set to 0.) It is recommended that the line be idle (closed or flushed) when this parameter is changed. The user should be careful not to increase the character count before the new characters are valid.



List of End of Buffer Characters (LP #60 through 69)

This is the list of characters to signal the end of a DMA block. These parameters can be changed while the board is operating by issuing a Change Line Parameter command. These changes take place immediately, even if there are characters in the on board receive FIFO. The characters in the FIFO will be compared against the new characters specified in the new count (which may be set to 0.) It is recommended that the line be idle (closed or flushed) when these parameters are changed. The user should be careful not to increase the character count before the new characters are valid.



Receive Timeout Value (LP #70)

This parameter specifies how may $1/20^{th}$ of a second periods to wait without receiving any new characters before ending the DMA cycle. When the maximum count has been reached, the controller will end the DMA cycle with a normal termination and write the number of bytes actually transferred to the receive count register. This is a 32-bit field. This parameter can be changed while the board is operating by issuing a Change Line Parameter command.

Master Mode Address Modifier (LP #71)

This parameter specifies the address modifier that the controller should place on the VMEbus when transferring data in DMA mode to the receive data buffer or from the transmit data buffer.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																										modifier					

Valid arguments for the Master Mode Address Modifier value are:

0x3F - A24 supervisory block transfer *
0x3E - A24 supervisory program access
0x3D - A24 supervisory data access
0x3B - A24 non-privileged block transfer *
0x3A - A24 non-privileged program access
0x39 - A24 non-privileged data access
0x2D - A16 supervisory data access
0x29 - A16 non-privileged data access
0x0F - A32 supervisory block transfer *
0x0E - A32 supervisory program access
0x0D - A32 supervisory data access
0x0B - A32 non-privileged block transfer *
0x0A - A32 non-privileged program access
0x09 - A32 non-privileged data access
*To use block transfers, the Block Mode and Burst Count controller parameter must be non-zero.

Master Mode Data Length (LP #72)

This parameter specifies the maximum number of bytes to transfer in a single DMA operation.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																													len		

Valid arguments for the Master Mode Data Length value are:

0x01 - One byte
0x02 - Two bytes
0x04 - Four bytes

Disable Receive Soft Flow Control (LP #73)

This parameter in conjunction with LP #74 allows separate software flow control for the receive and transmit channels. Both channels use the same XON and XOFF characters. This parameter can be changed while the board is operating by issuing a Change Line Parameter command.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																															R

Valid arguments for the Disable Receive Soft Flow Control flag are:

0 – Enable Receive Soft Flow Control
1 – Disable Receive Soft Flow Control

Disable Transmit Soft Flow Control (LP #74)

This parameter in conjunction with LP #73 allows separate software flow control for the receive and transmit channels. Both channels use the same XON and XOFF characters. This parameter can be changed while the board is operating by issuing a Change Line Parameter command.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																															T

Valid arguments for the Disable Transmit Soft Flow Control flag are:

0 – Enable Transmit Soft Flow Control
1 – Disable Transmit Soft Flow Control

DMA Receive Timeout Block Mode (LP #75)

This parameter allows selection of timeout modes for DMA data transfers. The firmware adds one tick to the specified interval to avoid the possibility of a short timeout when a timeout of 1 is specified. The timer could take from N to N + 1 ticks to expire. On average, a randomly started N tick timeout should expire in N + .5 ticks. This parameter can be changed while the board is operating by issuing a Change Line Parameter command.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																															T

Valid parameters for the DMA Receive Timeout Block Mode flag are:

0 – Inter-character mode: Start timer after the first character is received, then timeout if a period of time between characters exceeds the timer value.
1 – Block mode: Timeout if the DMA operation is not complete before the timer expires.

Receive Characters Available Now (LP #76)

This parameter allows selection of how receive DMAs should be terminated. It has no effect on PIO transfers. This parameter can be changed while the board is operating by issuing a Change Line Parameter command.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																															R

Valid parameters for the Receive Characters Available Now value are:

0 - Normal termination.
1 - Terminate the receive DMA when there are no characters in the on board receive FIFO for the specified line. This is useful if the host only wants the characters that are immediately available, and does not want to wait for more characters to be received or a DMA timeout. The Receive Characters Available Now status will be set if enabled in the Valid Error Status Mask (LP #48).
2 - Terminate the receive DMA when there are no characters in the on board receive FIFO for the specified line. Then automatically set this parameter to 0. This will terminate the receive DMA only once. The status Receive Characters Available Now will be set if enabled in Valid Error Status Mask (LP #48).

Line Parameters 77-128

These parameters are reserved for future use. They must be set to zero.

Programming

The information in this chapter is for those who wish to write their own software driver or host application for the MVC.

The MVC has four distinct operating modes that control the definition of the controller's shared ram area and the method in which the host communicates with it. These modes are Boot, Firmware Load, Initialization, and Run and are documented in the sections that follow.

3.1 Boot Mode

The firmware for the MVC is stored in volatile SRAM on the MVC and must be downloaded to the board each time the power is cycled or the VME SYSRESET line is activated. After the system's power is turned on, the MVC is in Boot mode. The MVC is held in a reset state with the processor stopped and the FAIL and BUSY indicators on. The MVC requires the host to write a bootload program into the MVC's shared RAM and start the processor. The bootload program allows the MVC to download the actual execution firmware.

When the MVC is first powered up, it is in the extended addressing mode. It will *only* respond to an extended address modifier of 0D or 09, and a data size of 16 bits. Even if your system supports only short or standard addressing modes, an extended address modifier must be used to access the MVC in Boot mode. After the MVC has been initialized, the addressing mode and data word size may be changed. Make sure that no other boards are assigned to the same address in the extended space as the MVC. The base address of the MVC is defined by the Board ID 1 & 2 switches (see Chapter 1 - *Installation*).

The boot code is loaded into sequential locations in the shared RAM starting at the board's base address. The length of the boot code *must not* exceed 240 (0xF0) bytes. Once the boot code is loaded, the 29K is released by reading or writing the 29K Reset/Release register at location base address + 0x160. When released, the 29K will begin executing the boot code. The 29K will change the mapping of its shared RAM to a different section of MVC internal RAM. The data in the new area should be deemed corrupt. The MVC will now be in Firmware Load mode.

Shared RAM in Boot Mode

Address	Data Halfword Bits 15-8	Bits 7-0
Base + 0x00	Byte 0	Byte 1
Base + 0x02	Byte 2	Byte 3
⇳	⇳	
Base + 0xEE	Byte 238	Byte 239
Base + 0x160	29K Reset/Release register	

3.2 Firmware Load Mode

Once the MVC has been successfully booted, it enters Firmware Load mode. In this mode, the boot code loads the controller firmware using the shared RAM area as a communications section. The sequence for loading the firmware is:

- Wait 250 milliseconds for the MVC to complete the switch from Boot to Firmware Load mode.
- After the transition from Boot mode, there is a change in the MVC's shared RAM addressing. It is mandatory that the software write a NOP command (0x00000001) to the Command/Status register (offset 0xF0) *twice* to ensure that the MVC and the host software are communicating through the same shared RAM area. Next, read the Command/Status register. The MVC should clear it almost immediately after receiving the commands. If it does not, either the boot code or the controller has failed.
- Write the first longword of the firmware to the Data In register (offset 0xF4). Write the command Load Firmware (0x00000002) to the Command register. The MVC will clear the Command register, echo the longword just written to the Data Out register, and add the value to the Checksum register when it has completed the command. You may read the Data Out and Checksum registers to verify that the firmware is loading correctly.
- Repeat the above step for the rest of the firmware. Once the firmware is loaded, write a Start Firmware command (0x00000005) to the Command register. The MVC will switch to Initialization mode.

Firmware Load Mode Register Descriptions

The address of a register is its offset value added to the base address of the MVC. For example, if the base address is 0xC2000000, the Command register is at address 0xC20000F0. The registers available during Firmware Load mode are summarized in the table below and described in the sections that follow.

Firmware Load Mode Registers

Offset	Register
0x0 – 0xEC	Reserved
0xF0	Command
0xF4	Data In
0xF8	Data Out
0xFC	Checksum

Command Register (Base address + Offset of F0)

This longword register is used to pass commands to the MVC. It is cleared after the MVC has completed processing the last received command. A new command may not be issued until the previous command completes. The commands are described in the section below.

Command Longword

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x000000																								Command							

Bits 31-8 = Reserved, must be zero.
Bits 7-0 = Command code byte

The syntax for issuing a command to the MVC is:

- Write any parameter data to the Data In register.
- Write the command to the Command register.
- Read the Command register. When it clears, the command is complete.
- Read any returned data from the Data Out register.

Data In (Offset 0xF4) and Data Out (Offset 0xF8) Registers

The Data In register (Offset 0xF4) is written with the longword firmware data value prior to issuing a Write Firmware command. The MVC echoes the data received to the Data Out register (Offset 0xF8) as it is loaded to allow the control program to verify that it was written correctly.

Checksum Register (Offset 0xFC)

The Checksum Register contains a running additive value of the data values as they are written to the MVC. This is another means of verifying that the firmware was loaded correctly. In the event of a checksum error, the control program should attempt to reload the firmware.

Firmware Load Mode Commands

The following commands are available during Firmware Load mode:

Firmware Load Mode Commands

Command	Description
0x00	No Command
0x01	No Operation
0x02	Load Firmware
0x03	Set Firmware Address
0x04	Read Firmware
0x05	Start Firmware

No Command (0x00000000)

When the controller has completed a command, it clears the Command Register. The MVC recognizes a new command when the command register contains a non-zero data value. Issuing this command has no effect on the operation of the MVC.

No Operation (0x00000001)

The boot code should be tested to ensure that the 29K is indeed running and that the MVC is now in Firmware Load mode. This can be accomplished by writing a NOP to the Command register. When the MVC recognizes this command, it will respond by clearing the Command/Status register.

Load Firmware (0x00000002)

The firmware is loaded in sequential 32 bit words beginning at MVC local memory address 0x500. Write the first longword of the firmware to the Data In register (offset 0xF4). Write the Load Firmware command to the Command register. The MVC will clear the Command register, echo the longword just written to the Data Out register, and add the value to the Checksum register when it has completed the command. Repeat the preceding for the rest of the firmware longwords.

The longword in the Data Out register should be compared to the original data written to ensure the MVC received the intended data. Also, the checksum value should be used to ensure that the data was recorded correctly.

Once the firmware is loaded, write a Start Firmware command (0x00000005) to the Command register. The MVC will switch to Initialization mode. It is loaded into the shared RAM using the Load Firmware command.

Set Firmware Address (0x00000003)

The boot code loads the firmware into sequential locations in the MVC's SRAM starting at local memory address 0x500. This operation continues until the host instructs the boot code to start the loaded firmware. The Set Firmware Address command allows the user to change the current load/read address to reload pieces of the firmware, or verify the code was loaded correctly.

☞ The set firmware address command can allow the firmware to be loaded in non-contiguous memory, or at a starting address other than 0x500. This should not be attempted, as the firmware will not function properly.

To issue this command, write the new local memory address to the Data In register, and the Set Firmware Address command to the Command register. The MVC will clear the Command register after it has processed the command. The next firmware longword will load at the specified address.

Read Firmware (0x00000004)

The Read Firmware command allows the user to read and verify data in a specific location of the MVC's local memory. The address to be read must be set using the Set Firmware Address command before issuing the Read Firmware command. The data will be returned in the Data Out register.

This command should not be used to read MVC device registers. Doing so may cause a MVC local bus error and a boot code crash.

Start Firmware (0x00000005)

Once the firmware has been loaded, it is started by writing a Start Firmware command to the Command register. Execution begins at address 0x500. The MVC will switch to Initialization mode.

3.3 Initialization Mode

After the MVC firmware has been successfully loaded and started, the MVC switches to Initialization mode. This is evident by the illumination of the RUN and BUSY indicators on the front panel. If both LEDs are not lit, the firmware code was not loaded properly or may be bad.

When the MVC enters Initialization mode, it writes information to the Firmware Part Number & Revision, Line Count, Printer Port Available, and Memory Size registers. These registers may be read by the software driver and used to alter its operation. The Line Count register value can be used to determine the number of line parameter blocks that must be downloaded to the MVC. This number will depend on the number of lines available (8 or 16). The Memory Size register value is the maximum amount of memory available for data FIFO buffers. The registers available during initialization mode are shown in the table below:

Initialization Mode Registers

Address	Register
0x0 – 0xCC	Reserved
Base + 0xD0	VIC Revision ID
Base + 0xD4	Reserved
Base + 0xD8	Printer Available Flag
Base + 0xDC	Firmware Part Number and Revision
Base + 0xE0	Line Count
Base + 0xE4	Memory Size
Base + 0xE8	Reserved
Base + 0xEC	Reserved
Base + 0xF0	Command
Base + 0xF4	Data In
Base + 0xF8	Reserved
Base + 0xFC	Reserved

The controller and line parameters are downloaded using the shared RAM area as a communications section. Parameters are written to the Data In register and the appropriate command is written in the Command register. The firmware will signal that it has accepted the command by clearing the Command register. Initialization mode supports only two commands:

- Load controller parameters (0x7)
- Load line parameters (0x8)

All controller parameters *must* be loaded before the MVC will accept the Load Line Parameters command. All line parameters must be loaded before the MVC will switch to Run mode. The controller parameter table is a set of sixteen 32-bit parameters. The line parameter block is a set of one hundred and twenty-eight 32-bit parameters for each line. All undefined bits in the controller and line parameter words *must* be zero. The parameters are defined in Chapter 2 - *Soft Configuration Options.*

Load Controller Parameters (0x00000007)

The controller parameters must be loaded first. The size of the controller parameter block is fixed at sixteen entries. They are loaded by writing the 32-bit controller parameter words to the Data In register and a Load Controller Parameters command to the Command register. The MVC will expect the line parameters to immediately follow the last controller parameter word.

Load Line Parameters (0x00000008)

The size of the line parameter block is fixed at 128 entries for each of lines zero through 16 (printer port). The 32-bit line parameter words are loaded by writing them to the Data In register, and writing a Load Line Parameter command to the Command register.

☞ If the requested size of the FIFO transmit and receive buffers exceeds the available memory, as specified in the memory size register, the firmware will first make the printer port unavailable, and then decrease the line count - and the value in the line count register - to the number of lines that will fit within the memory size value. The line count register should be checked after loading the line parameters and issuing a driver NOP command to ensure that the printer port and the desired number of lines are actually available.

The MVC will indicate that the controller and line parameters have been successfully loaded by turning off the BUSY LED and changing to Run mode.

3.4 Run Mode

After the controller and line parameters have been loaded, the MVC enters Run mode, indicated by the RUN indicator lighting and the BUSY indicator turning off. Run mode operation consists of issuing line commands, receiving interrupts, and issuing controller commands.

Line commands

Line commands act on a specific line. They include changing baud rates, changing word length, changing stop bits, etc. When a command requests a change in a line parameter that would affect the transmission of the characters in the FIFO, the request will be delayed until the transmit buffer empties and can be changed without altering the data. If another command is received that counteracts the first command while the MVC is waiting for the buffer to empty, the first command will be canceled.

Transmitting characters

Character transmission occurs in two modes: character (PIO) or DMA. This mode is determined during controller initialization by setting the appropriate flag in the line parameters.

Character mode (PIO)

In character mode, the character to be transmitted is placed in the transmit buffer of the appropriate line register, and the Transmit Buffer Full (TXF) bit in the line command word is set. The MVC will indicate the character has been transmitted by setting the Transmit Buffer Empty (TBE) bit in the status register and, if transmit interrupts are enabled, generating an interrupt.

DMA mode

In DMA mode, the transmit buffer area must be defined by setting the appropriate line and controller parameters specifying: sequential or scattered transfers; DMA mode; buffer or scatter table address; master mode address modifier; etc.

The data is placed in the appropriate buffer. If the MVC is in scatter/gather mode, the scatter/gather table must be set to point to the scatter buffer and the number of table entries written to the transmit buffer register.

The Transmit Buffer Full (TXF) bit in the command word is set, signaling the MVC to begin transmission. The MVC will indicate the character or buffer has been transmitted by setting the Transmit Buffer Empty (TBE) bit in the status register and, if transmit interrupts are enabled, generate an interrupt.

Receiving Characters

Character reception occurs in two modes: character (PIO) or DMA. This mode is determined during controller initialization by setting the appropriate flag in the line parameters.

Character Mode (PIO)

In character mode, the host must first signal the MVC that the receive buffer is empty by setting the Receive Buffer Empty (RXE) bit in the appropriate line command register. When the MVC receives a character, it is placed in the receive buffer of the appropriate line. The Receive buffer full (RBF) bit

in the status word is set, and if receive interrupts are enabled, an interrupt is generated. The host signals that the character has been removed from the buffer by setting the RXE bit in the line command register.

DMA mode

In DMA mode, the receive buffer must be defined by setting the appropriate line and controller parameters specifying: sequential or scattered transfers; DMA mode; buffer or scatter table address; master mode address modifier; etc.

In scatter/gather mode, the scatter/gather table must be set to point to the buffer and the number of table entries written to the receive buffer register.

The Receive Buffer Empty (RXE) bit is set in the line command word to signal that the MVC may begin to store data in the host's receive buffer.

When the MVC has filled its buffer, or a break character is received, the FIFO buffer is DMA'ed to the host, the number of characters DMA'ed is written to the Receive count register, the Receive Buffer Full (RBF) bit in the line status word is set, anf if receive interrupts are enabled, an interrupt is generated. The host signals the character has been removed from the buffer by setting the Receive Buffer Empty (RXE) bit in the line command register.

Scatter Gather Table

The scatter gather table is a sequential list of address modifiers, addresses, and byte counts. The scatter gather option is selected by the Scatter Gather Flag controller parameter. The pointers to this list are the line parameters DMA Receive Buffer Address and DMA Transmit Buffer Address.

Scatter Gather Table

Entry	Data Word
Entry 1	Address Modifier
	Address
	Count
Entry 2	Address Modifier
	Address
	Count
Entry n	Address Modifier
	Address
	Count

Address Modifier is the VMEbus address modifier for this buffer. *Address* points to the first byte in a contiguous block of memory. *Count* is the number of bytes in the block.

The scatter gather table MUST be longword aligned in host memory.

Processing Interrupts

Interrupts are generated according to the Interrupt Flags in the line parameters and the Interrupt Levels in the controller parameters. If an interrupt level of zero is specified, interrupts are disabled. If a non-zero value is specified for receive or transmit interrupt levels, the interrupt flags in the line parameters determine which conditions will allow an interrupt.

Controller Commands

Controller commands can be issued at any time after entering run mode. They are issued by writing the data into the data register, the address (if required) to the address register, and the command to the command register. The MVC has completed the command when it clears the command register. If data was to be returned, the data register is valid after the MVC has cleared the command word.

Receive Buffer

The Receive Buffer register has dual definitions depending on the mode set for the specific line (character, standard DMA, or scatter/gather DMA mode) in the line parameters. If the line is set to character mode (PIO), the line Receive Buffer is an eight bit character buffer for holding the received data byte until the host can retrieve it.

In DMA scatter gather mode, the Receive Buffer register contains the number of entries in the scatter gather table. The Receive Count register is incremented every time a character is transferred. The host can clear the Receive Count register before issuing a Receive Buffer Empty line command. The Receive Buffer register is not used in standard DMA mode.

Transmit Buffer

The Transmit Buffer has dual definitions depending on the mode set for the specific line (character or DMA mode) in the line parameters. If the line is set to character mode (PIO), the line Transmit Buffer is an eight bit character buffer for holding the transmit data byte until the MVC can transmit it.

In DMA scatter gather mode, the Transmit Buffer register contains the number of entries in the transmit scatter gather table. The MVC reads from the table until it is exhausted. In this mode, the count is set by the host and read by the MVC. The Transmit Buffer register is not used in standard DMA mode.

Receive Count

When the line is in DMA mode, this word contains the number of characters actually DMA'ed to the host's receive buffer. The count will continue to increment until it is cleared by the host software driver. This register is not used in the character mode.

3.5 Run Mode Registers

The registers available during Run mode are shown in the table below and explained throughout the remainder of this section.

Run Mode Registers			
Address	**Register**		
Base + 0x0	Line 0 Command		
Base + 0x4		Status	
Base + 0x6			Receive
Base + 0x7			Transmit
Base + 0x8	Line 0 Receive count		
Base + 0xC	Line 1 Command		
Base + 0x10		Status	
Base + 0x12			Receive
Base + 0x13			Transmit
Base + 0x14	Line 1 Receive count		
⇳	⇳		
Base + 0xB4	Line F Command		
Base + 0xB8		Status	
Base + 0xBA			Receive
Base + 0xBB			Transmit
Base + 0xBC	Line F Receive count		
Base + 0xC0	Printer port command		
Base + 0xC4		Status	
Base + 0xC6			Reserved
Base + 0xC7			Transmit
Base + 0xC8	Reserved		
Base + 0xD0	VIC Revision ID		
Base + 0xD8	Printer port available		
Base + 0xDC	Firmware part number and revision		
Base + 0xE0	Line Count		
Base + 0xE4	Memory Size		
Base + 0xE8	Receive Interrupt Mask		
Base + 0xEC	Transmit Interrupt Mask		
Base + 0xF0	Command/Status		
Base + 0xF4	Data In		
Base + 0xF8	Data Out		
Base + 0xFC	Reserved		

Line Registers

Each line has its own Command, Status, Receive, Transmit, and Receive Count registers. These register map as follows:

Line Registers			
Address	**Register**		
Base + 12*n	Line n Command		
Base + 12*n+4		Status	
Base + 12*n+6			Receive
Base + 12*n+7			Transmit
Base + 12*n+8	Receive count		

where ‘n’ is the line number from 0 through 16 (0x0 - 0x10).

Line Command Register

The Line Command register is 32 bits wide. After the MVC has read the new command, it will be cleared to inform the host that the command has been processed. The host must not issue a new command until the register has been cleared. ALL unused bits in the command word *must* be cleared (0) before posting a new command. The format of the Line Command register is:

Line Command Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CBK	SBK	ST1	ST0	CST	CL1	CL0	CCL	TB3	TB2	TB1	TB0	CTB	RB3	RB2	RB1	RB0	CRB	PR1	PR0	CRT	SRT	CDT	SDT	RXE	TXF	DRx	ERx	DTx	ETx	DI	EI
Stop transmitting line break	Begin transmitting line break	bit 1	bit 0	Change Number of Stop Bits	bit 1	bit 0	Change Bits Per Character	bit 3	bit 2	bit 1	bit 0	Change Transmit Baud Rate	bit 3	bit 2	bit 1	bit 0	Change Receive Baud rate	Parity mask (bit 1)	Parity mask (bit 0)	Clear Request To Send	Set Request To Send	Clear Data Terminal Ready	Set Data Terminal Ready	Receive Buffer Empty	Transmit Buffer Full	Disable receiver	Enable receiver	Disable transmitter	Enable transmitter	Disable Interrupts	Enable Interrupts

Bit	Definition
0	Enable interrupts: this bit enables interrupts for this line, and clears the line status register.
1	Disable interrupts: this bit disables interrupts for this line, and clears the line status register.
2	Enable transmitter: this bit enables the data transmitter.
3	Disable transmitter: this bit disables the data transmitter.
4	Enable receiver: this bit enables the data receiver.
5	Disable receiver: this bit disables the data receiver.

<table>
<tr><th>Bit</th><th>Definition</th></tr>
<tr><td>6</td><td>Transmit Buffer Full: indicates the transmit buffer is filled with data.</td></tr>
<tr><td>7</td><td>Receive Buffer Empty: indicates the receive holding buffer is empty.</td></tr>
<tr><td>8</td><td>Set Data Terminal Ready: causes the DTR output line to be asserted.</td></tr>
<tr><td>9</td><td>Clear Data Terminal Ready: clears the DTR output line.</td></tr>
<tr><td>10</td><td>Set Request To Send: causes the RTS output line to be asserted.</td></tr>
<tr><td>11</td><td>Clear Request To Send: clears the RTS output line.</td></tr>
<tr><td>12, 13</td><td>Parity mask: these bits signal the firmware to change the parity type.
<table>
<tr><th>Bit Values 13 12</th><th>Definition</th></tr>
<tr><td>0 0 (0x0)</td><td>No change</td></tr>
<tr><td>0 1 (0x1)</td><td>Odd parity</td></tr>
<tr><td>1 0 (0x2)</td><td>Even parity</td></tr>
<tr><td>1 1 (0x3)</td><td>No parity</td></tr>
</table></td></tr>
<tr><td>14-18</td><td>When bit 14 is set, the MVC is instructed to change the receive baud rate. Bits 15 through 18 select the new baud rate.
<table>
<tr><th rowspan="2">Bit Values 18 17 16 15</th><th colspan="2">Baud Rate</th></tr>
<tr><th>Group 0</th><th>Group 1</th></tr>
<tr><td>0 0 0 0 (0x0)</td><td>50</td><td>75</td></tr>
<tr><td>0 0 0 1 (0x1)</td><td>110</td><td>110</td></tr>
<tr><td>0 0 1 0 (0x2)</td><td>34.5</td><td>38.4K</td></tr>
<tr><td>0 0 1 1 (0x3)</td><td>1200</td><td>150</td></tr>
<tr><td>0 1 0 0 (0x4)</td><td>300</td><td>300</td></tr>
<tr><td>0 1 0 1 (0x5)</td><td>600</td><td>600</td></tr>
<tr><td>0 1 1 0 (0x6)</td><td>1200</td><td>1200</td></tr>
<tr><td>0 1 1 1 (0x7)</td><td>1050</td><td>2000</td></tr>
<tr><td>1 0 0 0 (0x8)</td><td>2400</td><td>2400</td></tr>
<tr><td>1 0 0 1 (0x9)</td><td>4800</td><td>4800</td></tr>
<tr><td>1 0 1 0 (0xA)</td><td>7200</td><td>1800</td></tr>
<tr><td>1 0 1 1 (0xB)</td><td>9600</td><td>9600</td></tr>
<tr><td>1 1 0 0 (0xC)</td><td>38.4K</td><td>19.2K</td></tr>
<tr><td>1 1 0 1 (0xD)</td><td>Not Valid</td><td>Not Valid</td></tr>
<tr><td>1 1 1 0 (0xE)</td><td>Not Valid</td><td>Not Valid</td></tr>
<tr><td>1 1 1 1 (0xF)</td><td>Not Valid</td><td>Not Valid</td></tr>
</table></td></tr>
<tr><td>19-23</td><td>When bit 19 is set, the MVC is instructed to change the transmit baud rate. Bits 20 through 23 select the new baud rate.
<table>
<tr><th rowspan="2">Bit Values 23 22 21 20</th><th colspan="2">Baud Rate</th></tr>
<tr><th>Group 0</th><th>Group 1</th></tr>
<tr><td>0 0 0 0 (0x0)</td><td>50</td><td>75</td></tr>
<tr><td>0 0 0 1 (0x1)</td><td>110</td><td>110</td></tr>
<tr><td>0 0 1 0 (0x2)</td><td>34.5</td><td>38.4K</td></tr>
<tr><td>0 0 1 1 (0x3)</td><td>1200</td><td>150</td></tr>
<tr><td>0 1 0 0 (0x4)</td><td>300</td><td>300</td></tr>
<tr><td>0 1 0 1 (0x5)</td><td>600</td><td>600</td></tr>
<tr><td>0 1 1 0 (0x6)</td><td>1200</td><td>1200</td></tr>
<tr><td>0 1 1 1 (0x7)</td><td>1050</td><td>2000</td></tr>
<tr><td>1 0 0 0 (0x8)</td><td>2400</td><td>2400</td></tr>
<tr><td>1 0 0 1 (0x9)</td><td>4800</td><td>4800</td></tr>
<tr><td>1 0 1 0 (0xA)</td><td>7200</td><td>1800</td></tr>
<tr><td>1 0 1 1 (0xB)</td><td>9600</td><td>9600</td></tr>
<tr><td>1 1 0 0 (0xC)</td><td>38.4K</td><td>19.2K</td></tr>
<tr><td>1 1 0 1 (0xD)</td><td>Not Valid</td><td>Not Valid</td></tr>
<tr><td>1 1 1 0 (0xE)</td><td>Not Valid</td><td>Not Valid</td></tr>
<tr><td>1 1 1 1 (0xF)</td><td>Not Valid</td><td>Not Valid</td></tr>
</table></td></tr>
</table>

Bit	Definition
24	When bit 24 is set, the MVC is instructed to change the number of bits per character. Bits 25 and 26 select the new value. Bit Values (26 25) — Definition 0 0 (0x0) — 5 bits 0 1 (0x1) — 6 bits 1 0 (0x2) — 7 bits 1 1 (0x3) — 8 bits
27	When bit 27 is set, the MVC is instructed to change the number of stop bits. A data character length of 5 bits requires at least 1.5 stop bits. Bits 28 and 29 set the new value. Bit Values (29 28) — Definition 0 0 (0x0) — 1 stop bit 0 1 (0x1) — 1.5 stop bits 1 0 (0x2) — 2 stop bits 1 1 (0x3) — Invalid
30	Begin transmitting line break: forces the TXD output line to the spacing condition.
31	Stop transmitting line break: the TXD output line will go to the marking condition and remain high for at least one bit time.

Line Status Word

Status information will *only* be posted when the status has changed. Current modem status bits are always given with new status information. Modem signal changes from set to clear will be indicated by a zero in the signal bit position with the valid status bit set. After the host has read the Status word, it must be cleared before the MVC will write any new status information or generate an interrupt.

If an interrupt was requested for a specific line, using one or more of the interrupt flags in the line parameters, an interrupt will be generated after the status has been posted. If an interrupt was requested for any error condition, the MVC will interrupt using the receive interrupt vector. Interrupts for a given modem status reflects a change in that status.

The line status word is a 16-bit word defined as follows:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MSV	RCAN	RES	BSTR	BERR	TBE	RBF	RFAF	RBD	FRE	PE	OVE	DCD	RI	CTS	DSR

Line Status Word

Bit	Name	Description
0	DSR	Data Set Ready: Reflects the state of the DSR input signal.
1	CTS	Clear To Send: Reflects the state of the CTS input signal.. This bit will always report 0 while hardware flow control (RTS/CTS) is active for the specified line.
2	RI	Ring Detected: Reflects the state of the RI input signal.
3	DCD	Data Carrier Detect: Reflects the state of the DCD input signal.
4	OVE	Overrun Error: An octart overrun error has occurred. Data has been lost.

Line Status Word		
Bit	**Name**	**Description**
5	PE	Parity Error: A character was received with incorrect parity.
6	FRE	Framing Error: An unexpected stop bit was detected for a received character. This error usually indicates that the wrong baud rate or character length is being used.
7	RBD	Receive Break Detect: An all zeros character was received without a stop bit.
8	RFAF	Receive FIFO Buffer Almost Full: The receive FIFO buffer changed from a not full to an almost full state. Receiving a new character will result in an Overrun Error. This status may be disabled by setting LP #49 (Receive FIFO Almost Full) to 0.
9	RBF	Receive Buffer has been Filled: The receive holding buffer has data that is ready to be input to the system. In DMA block mode, this bit indicates that the receive DMA block operation is complete.
10	TBE	Transmit Buffer has been Emptied: All data has been removed from the transmit holding buffer, and it is available for new data. In DMA block mode, this bit indicates that the transmit DMA block operation is complete.
11	BERR	Bus Error (DMA mode): The 29K was unable to access the device or area of memory pointed to by the driver. This status may not be disabled. The board automatically retries a bus error and only reports this status for two consecutive bus errors.
12	BSTR	Break Started: The transmit buffer has drained, and a pending break has started.
13	TFWE	Transmit FIFO Went Empty: This bit indicates that all characters have been transmitted out to the serial line. There are no characters in the host, MVC, or UART line buffers needing to be sent. This status is valid for the printer if the TFWE bit 13 is set in the printer LP #48. This status will cause an interrupt via the Receive Interrupt Masks.
14	RCAN	Receive Characters Available Now: A DMA mode receive command finished because there were no characters in the on board receive FIFO.
15	MSV	Modem status information valid: The modem status bits are valid.

Receive Buffer

The Receive Buffer register has dual definitions depending on the mode set for the specific line (character, standard DMA, or scatter/gather DMA mode) in the line parameters. If the line is in character mode (PIO), the line Receive Buffer is an eight bit character buffer for holding the received data byte until the host can retrieve it. In DMA scatter gather mode, the Receive Buffer register contains the number of entries in the scatter gather table. The MVC fills the table until it is exhausted or all characters have been transferred. The number of characters DMA'ed is written to the Receive Count register. In this mode, the count value is set by the host and read by the MVC. The Receive Buffer register is not used in standard DMA mode.

Transmit Buffer

The Transmit Buffer has dual definitions depending on the mode set for the specific line (character, standard DMA, or scatter/gather DMA mode) in the line parameters. If the line is set to character mode (PIO), the line Transmit Buffer is an eight bit character buffer for holding the transmit data byte until the MVC can transmit it. In DMA scatter gather mode, the Transmit Buffer register contains the number of entries in the transmit scatter gather table. The MVC reads from the table until it is exhausted. In this mode, the count is set by the host and read by the MVC. This register is not used in standard DMA mode.

Receive Count

When the line is in DMA mode, this word contains the number of characters actually DMA'ed to the host's receive buffer. The count will continue to increment until it is cleared by the host software driver. This register is not used in character mode.

Configuration registers

The configuration registers may be read by the software driver and used to alter its operation.

VIC Revision ID (Base + 0xD0)

This value contains the revision code of the VIC068 IC installed on the MVC. If you contact Macrolink for technical support, you may be asked to provide this value.

Printer Available Flag (Base + 0xD8)

This flag will contain the value 0x00000001 if the printer port is installed and enabled. It will be cleared otherwise.

Firmware Part Number and Revision (Base + 0xDC)

This register contains the firmware part number and revision. If you contact Macrolink for technical support, you may be asked to provide this value.

Line Count (Base + 0xE0)

This register contains the number of lines available for use. This value is adjusted during Initialization mode, with the number of lines available reduced if the total size of the transmit and receive buffers exceeds the amount of available buffer memory.

Memory Size (Base + 0xE4)

This register value is the maximum amount of memory available for data FIFO buffers.

Interrupt Mask Registers

When a line status bit change requires a corresponding interrupt, the MVC will set the bit associated with the line number in the interrupt flag register (i.e. bit zero for line zero, etc.) and post the interrupt.

After the host has read either the transmit or receive interrupt mask words, it must be cleared before the MVC will write new interrupt flags or generate new interrupts. New interrupts for a specific line will be blocked when status information from a previous interrupt was not processed (read and cleared).

Transmit Interrupt Mask (Base + 0xE8)

As the MVC transmits data, it sets an internal flag indicating that an interrupt is required for a specific line. These flags are queued until a group of lines have been serviced. After the lines have been serviced, the MVC sets the appropriate bits in the transmit interrupt mask and posts an interrupt. The MVC will not allow a new interrupt to occur until the host has read and cleared the previous interrupt information.

Each bit of this register is associated with a specific line:

Transmit Interrupt Mask

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																Line F	Line E	Line D	Line C	Line B	Line A	Line 9	Line 8	Line 7	Line 6	Line 5	Line 4	Line 3	Line 2	Line 1	Line 0

Receive Interrupt Mask (Base + 0xEC)

As the MVC receives data, it sets an internal flag indicating that an interrupt is required for a specific line. These flags are queued until a group of lines have been serviced. After the MVC has serviced the lines, it sets the appropriate bits in the receive interrupt mask and posts an interrupt. The MVC will not allow a new interrupt to occur until the host has read and cleared the previous interrupt information.

Each bit of this register is associated with a specific line:

Receive Interrupt Mask

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																Line F	Line E	Line D	Line C	Line B	Line A	Line 9	Line 8	Line 7	Line 6	Line 5	Line 4	Line 3	Line 2	Line 1	Line 0

Controller Registers

Controller Command/Status Register (Base + 0xF0)

The controller Command/Status register is used to issue run mode controller commands and receive status for commands issued. See section 3.6 - *Controller Commands* for a complete list of MVC commands. The format of the command longword is:

Run Mode Commands

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bus Request/Reserved								LP Number								Line Number								Command							

Bits 31-24 = Bus request level code for Power Fail Restore, otherwise must be set to all zeroes
Bits 23-16 = Line parameter number for the Change Line Parameter command
Bits 15-8 = Line number(for commands which require one)
Bits 7-0 = Command code byte

Some commands require a value be written to the Data In register before issuing the command. Others will return a value to the Data Out register. The value will be valid after the command register clears.

The MVC indicates that it has accepted and is processing the command by clearing the command register. If an invalid command is issued to the MVC, bit 31 of the Command/Status register will be set and bits 0-30 will be left unchanged. The driver may then analyze bits 0-30 to determine the error.

Controller Data In Register (Base + 0xF4)

The 32-bit Data In register is used to pass parameter data to the MVC for commands which require it. This register is written by the host before issuing the appropriate command.

Controller Data Out Register (Base + 0xF8)

The 32-bit Data Out register is used to return data to the host after the completion of certain commands. The data is valid after receiving a Command Complete response from the MVC in the status register.

3.6 Controller Commands

The commands available during Run mode are shown in the table below and detailed in the sections that follow. Unused command codes are reserved for future definition.

Run Mode Commands	
Command	**Code**
No Operation (NOP)	0x00000001
Read byte	0x0000000B
Enable Auto Echo	0x0000LL0C
Disable Auto Echo	0x0000LL0D
Change Line Parameter	0x00PPLL0E
Power Fail Memory Size	0x00000010
Power Fail Save	0x0000AA11
Power Fail Restore	0xBBRRAA12
Flush	0x0000LL13
Disable Receiver	0x0000LL14

No Operation (0x00000001)

This command causes no action to occur on the MVC other than it clearing the Command/Status register after accepting the command. This command can be issued to test that the MVC command interface is operational.

Enable Auto Echo (0x0000LL0C)

Disable Auto Echo (0x0000LL0D)

These commands are used to override the setting of LP #7 (Auto Echo Flag). The LL byte (bits 8-15) contains the line number that will be affected. Upon completion, Auto Echo is enabled or disabled on the specified line and the Status register is cleared.

The Change Line Parameter command is the preferred implementation of this function. Support for these commands is provided for compatibility with drivers which use these commands.

Change Line Parameter (0x00PPLL0E)

This command is used to override the settings of line parameters. The new value for a parameter is written the Data In register before issuing this command. The LL byte (bits 8-15) contains the line number that will be affected. The PP byte (bits 16-23) contains the parameter number to be changed. Upon completion, the parameter will be updated and the Status register cleared.

The line parameters that may be changed using this command are shown in the table on the next page.

Changeable Line Parameters	
LP #	Description
7	Auto Echo Flag
8	Hardware Auto Flow Control Flag
9	XOFF Character
10	XON Character
47	Valid Modem Input Mask
48	Valid Error Status Mask
59	DMA End Of Block Character Count
60	DMA End Of Block Character 0
61	DMA End Of Block Character 1
62	DMA End Of Block Character 2
63	DMA End Of Block Character 3
64	DMA End Of Block Character 4
65	DMA End Of Block Character 5
66	DMA End Of Block Character 6
67	DMA End Of Block Character 7
68	DMA End Of Block Character 8
69	DMA End Of Block Character 9
70	DMA Receive Timeout
73	Disable receive software flow control
74	Disable transmit software flow control
75	DMA Receive Timeout Block Mode

Power Fail Memory Size (0x00000010)

This command will cause the MVC to place the amount of memory bytes required for the Power Fail Save command into the Data Out register. The next command to the board MUST be a Power Fail Save command. If a different command is issued (e.g. NOP), the MVC will post a Bad Command status to the Status register and return to normal operation. Upon completion, the number of memory bytes required is written to the Data Out register and the Command register is cleared.

Power Fail Save (0x0000LL11)

This command will cause the MVC to save its state in host memory. The host memory must be longword aligned and longword accessible. The LL byte (bits 8-15) contains the address modifier for the DMA operation. A host memory pointer must be placed in the Data In register before issuing this command. The Data Out register will contain a checksum after this command completes. The host may use this checksum to verify a good restore using the

Power Fail Restore command. This command may only be issued after a Power Fail Memory Size command.

The amount of host memory required may be found by using the Power Fail Memory Size command. The MVC will light the FAIL, BUSY, and RUN indicators to indicate it has finished saving its state and must be rebooted.

The host may receive one more interrupt from the MVC after this command is issued. The host should acknowledge the interrupt, but process it only after the Power Fail Restore command.

Power Fail Restore (0xBBPPLL12)

This command will cause the MVC to restore its state using host memory. The host memory must be longword aligned and longword accessible. The Data In register contains the host memory pointer. The LL byte (bits 8-15) contains the address modifier for the DMA operation. The PP byte (bits 16-23) contains the bus release control code for the DMA operation.

0x3F - Release on request (ROR)
0x7F - Release when done (RWD)
0xBF - Release on BCLR (ROC)

The BB byte (bits 24-31) contains the bus request level code for the DMA operation.

0x00 - Level 0 (BR0)
0x20 - Level 1 (BR1)
0x40 - Level 2 (BR2)
0x60 - Level 3 (BR3)

A host memory pointer is written to the Data In register before issuing this command. After this command is issued the MVC should not be accessed for 1 second. After the 1 second delay the driver may then check for completion of this command.

The Data Out register will contain a checksum after this command completes. The host may use this checksum to verify a good restore by comparing this value to the checksum obtained during the Power Fail Save command. If the controller detects a bad checksum during the restore it will halt with the FAIL, BUSY, and RUN indicators lit.

If the host received one more interrupt from the MVC after the Power Fail Save command, the driver should now process the interrupt.

This command should only be issued at MVC boot time as a replacement for the Load Controller Parameter and Load Line Parameter commands.

Flush (0x0000LL13)

This command will cause the MVC to flush the receive channel, transmit channel, or both channels of the line specified by the LL byte (bits 8-15). The flush parameter is written to the Data In register before issuing this command:

1 = Flush the receive channel.
2 = Flush the transmit channel.
3 = Flush both the receive and transmit channels.

If the receive channel is flushed while processing a Receive Buffer Empty command, the command will be canceled. The host must clear the Receive Count register after the flush completes.

If the transmit channel is flushed while processing a Transmit Buffer Full command, the command will be canceled. The TBE and TFWE statuses will NOT be reported.

After a channel has been flushed, a new Receive Buffer Empty and/or Transmit Buffer Full command will have to be issued to start a transfer.

Disable Receiver (0x0000LL14)

This command disables the line's receiver without flushing the receiver buffer and without resetting the octart. The LL byte (bits 8-15) specify the line number.

3.7 Application Notes

The information in this section pertains to unique applications or configurations of the MVC.

Interrupt levels and vector numbers

If you intend to use separate receive and transmit interrupt levels or vector numbers, then both the interrupt levels and vector numbers must be set differently. In this mode, the driver transmit interrupt function should only clear Transmit Buffer Empty (TBE) in the corresponding status register, and the receive interrupt function should clear all bits except TBE. A Read-Modify-Write instruction should be used to clear the status register bits or disable interrupts to avoid status register corruption by a higher priority interrupt.

Valid interrupt level and vector number examples:

Example 1:

Receive Interrupt Level (CPB 2) = 4
Transmit Interrupt Level (CPB 4) = 4
Receive Vector Number (CPB 3) = 0x73
Transmit Vector Number (CPB 5) = 0x73

Example 2:

Receive Interrupt Level (CPB 2) = 4
Transmit Interrupt Level (CPB 4) = 3
Receive Vector Number (CPB 3) = 0x74
Transmit Vector Number (CPB 5) = 0x73

Invalid interrupt level and vector number examples:

Example 1:

Receive Interrupt Level (CPB 2) = 3
Transmit Interrupt Level (CPB 4) = 4
Receive Vector Number (CPB 3) = 0x73
Transmit Vector Number (CPB 5) = 0x73

Example 2:

Receive Interrupt Level (CPB 2) = 4
Transmit Interrupt Level (CPB 4) = 4
Receive Vector Number (CPB 3) = 0x74
Transmit Vector Number (CPB 5) = 0x73

Receive and Transmit Buffer Management

The MVC uses a round-robin approach for servicing the serial lines. Each line is checked, in numerical order, to see if characters are to be sent or have been received. This will occur for the total number of lines that are installed and enabled, regardless as to whether they are in use.

A method that can be used to enhance board performance is to allocate buffer space for unused lines to lines that are active. For example, assume that only five out of sixteen serial lines will be used. Allocating all of the buffer space to those five lines causes the MVC to disable the remaining lines, which means that the firmware will not have to service them. The firmware only polls the five remaining lines, thus speeding up their operation.

Lines are disabled in descending order, beginning with the printer port, then proceeding from lines F (15) through 1.

Flow Control

Flow control is a data management system that stops serial characters from being transmitted when the receiving device is not ready to accept them. Hardware flow control uses the Request To Send and Clear To Send (RTS/CTS) signal lines on the serial interface. Software flow control uses special characters, called XON and XOFF (short for Transmitter On and Transmitter Off) to temporarily suspend data transmission. This temporary suspension does not affect the transmission of XON and XOFF characters. Each serial line on the MVC can use hardware or software-based flow control or both in combination. Simultaneous use of both flow control types is rare, because hardware flow control can interfere with transmission of XON and XOFF characters. One use for both types in combination is to avoid sending to a device that only supports XON/XOFF flow control when it is turned off. Line parameters 8, 9, 10, 49, 73, and 74 are used to select and manage the type of flow control in use for each line. Refer to section 2.2 – *Line Parameter Block* in Chapter 2 for more information.

Software flow control and ldterm

Unix systems with STREAMS based serial port drivers exhibit unexpected flow control behavior for receive data when ldterm is pushed on the stream. Some flavors of unix cannot pause the remote transmitter, and some cannot pause the remote transmitter when only hardware flow control is selected, because the standard module ldterm does not use the STREAMS queue flow control method for received data. This is an issue for anyone implementing a connection to a device that sends bursts of data longer than 256 bytes.

Details for the driver writer

The MVC receives data and passes it upstream until the stream capacity is filled. The Driver stops issuing RXEs to preserve the last data DMAed but not queued to the stream, and because the receiver is still enabled, the data will continue to fill the board's receive FIFO buffer until the board issues an XOFF, which should pause the remote transmitter. If ldterm is present, then receive data will never back up into the receive side of the low level driver because of STREAMS queue flow control. The upstream message queue will always take receive data from the driver, because ldterm throws away all of its receive data as soon as it accumulates MAX_INPUT bytes in raw mode or exceeds MAX_CANON bytes in canonical mode. The IMAXBEL termio input mode affects this behavior, but not in a transparent way. The values of MAX_INPUT and MAX_CANON vary from platform to platform, but are at least 256. 512 is common.

Ldterm receive flow control is implementation dependent and depends on flow control messages sent from ldterm to the driver to suspend and resume input (M_STARTI and M_STOPI). Not all flavors of ldterm send these messages. Also, chances are that if the driver tells ldterm that the driver will handle ixoff (via M_CTL response), then ldterm will not generate the messages. If you can coax ldterm into sending these message types to the driver, then you have an event for the driver to stop queueing receive data. The maximum length of any single received data message buffer from the driver must be less than the space left to fill when ldterm sends M_STOPI, or the maximum count will be reached and the data will be thrown away anyway. Ldterm's raw received data buffer is not a message queue. There is no proper way for the driver to peek and thereby avoid hitting the limit. That local buffering is

also what prevents the use of stream queue flow control to manage this. This is one of the areas of the unix tty subsystem that still assumes that the driver is sending each received data byte upstream as it is received.

In practice, using a maximum received data message buffer size of 16 bytes works well. Note that this says nothing about how large the MVC's DMA receive buffer can be.

DMA–Character Mode

The MVC allows the user to set the transmit handler to DMA mode and the receive handler to Character mode. This allows incoming information to be processed on a character basis while still maintaining a high data throughput on transmit. Auto Echo should be used in this mode. It may also be necessary to enable auto flow control to protect against overrunning the terminal or device.

Auto Echo Disable

It may be desirable to disable the Auto Echo feature during Run mode, for example, while a password is being entered, and then re-enable it. The Change Line Parameter command will provide this functionality. The Change Line Parameter command is used to override the settings of line parameters.

Change Line Parameter Command (0x00PPLL0E)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
00								Parameter								Line Number								0E							

Write a 1 to the Data In register to enable Auto Echo, or a zero to disable Auto Echo on the specified line. Next, write the Change Line Parameter command information to the Command register. The LL byte (bits 8-15) contains the line number, from 0-F, that will be changed. The PP byte (bits 16-23) contains the parameter number, which is seven (07) for the Auto Echo Flag. The Command register will clear (become 0) after the MVC has processed the command.

Troubleshooting VME or VIC Bus Errors

VME and VIC Bus errors are typically caused by incorrect system VME configuration. Check for the following:

- Missing backplane BG[0-3] jumpers on a vacant VME slot between MVC and host CPU board.
- Host VME slave space is not enabled.
- Another VME board's slave address space overlaps Host slave space.
- Host VME slave space does not support the address modifier the MVC is using (A24 vs. A32).
- Host VME slave space does not support VME mode the MVC is using (non-block vs. BLT).
- Host VME slave space is configured larger than host hardware can support.
- Corrupted VME DTACK signal. This causes the board to attempt access at address 0xFFFFFFFE.
- Broken or missing VME terminator. This causes board to attempt access at random addresses.

RS-485 Interface

The RS-485 standard specifies the electrical characteristics of drivers and receivers that could be used to implement a balanced multi-point line, also known as a "party line." The devices attached to this line will support 32 driver/receiver connect points. A line length of up to 50 meters is supported by this interface. The transmission line in use must be 120 Ohm twisted pair terminated at both ends with a 120 Ohm resistor. Terminators are provided on the RS485 line modules and must be connected by the user if they are required.

There is no implied protocol supported by RS-485 nor is there any hardware flow control. RS-485 as implemented by Macrolink does not have any collision recovery. It is intended to be used in a system employing one "talker or initiator" and multiple "listeners or targets."

The RS485 interface on the MVC is of the 2-wire type, where transmit and receive data are multiplexed on the same wire pair. The RTS signal on the MVC is used to determine the data direction for each line. When set by writing a one to bit 10 in the Line Command Register, the specified line will be in transmit mode. When RTS is cleared by writing a one to bit 11 of the LCR, the specified line is in receive mode. Macrolink-supplied drivers automate the handling of transmit/receive switching.

The jumpers on the I/O transition panel must be strapped for the RS-422 setting. The RS-485 pinout is shown in the table below. The J1/J2 connector signals are based on the RS-422 pinout and repeat every 10 pins. For example, TRDA for channel 0 is on pin 2, for channel 1 it is on pin 12, for channel 4 it is on pin 42, etc.

RS-485 Pinout

DB-25 Pin	RJ-45 Pin	J1/J2 Pin	Signal	Description
1	4	x1	CGND	Chassis Ground
2	5	x2	TRDA	Transmit/Receive Data +
14	10	x9	TRDB	Transmit/Receive Data –
3	6	x3	RT+	120 Ohm Terminator Resistor +
16	9	x8	RT-	120 Ohm Terminator Resistor –
7	7	x7	SGND	Signal Ground

Line parameter number 50 is used to indicate that the specified line is configured as RS485. Valid parameter values are TRUE (1) or FALSE (0). This parameter should be set to TRUE for each line that uses the RS-485 interface.

RS485 cannot use XON/XOFF software flow control. XON/XOFF assumes a full duplex path. RS485 is a hardware half-duplex single differential pair, so there is no "back channel" for the XON/XOFF characters to travel on. Typically, RS485 uses packets of a known maximum length and an ACK/NAK/Time-out protocol to cause the next packet to be sent or the last packet to be resent. The transmitting node asserts RTS to enable its transmitter onto the differential wire pair. All receiving nodes turn RTS off to enable their receivers on the same differential wire pair to hear what is being sent. For lines configured as RS-485, Line Parameter #7 Auto Echo and Line Parameter #9 XOFF Character must both be set to zero.

This page has no technical content.

Interface Signals

This appendix provides pin numbering and signal identification information for the interface connectors used on the MVC.

A.1 8 Line DB-25 I/O Interface Panel

The pinout of the DB-25 connectors on the I/O panel is shown in Figure A-1. Pin 1 is nearest the bottom of the I/O panel when it is installed in a chassis.

Jumper plugs on the I/O panel are used to select either RS232C or RS422 modes for each group of 4 lines (0-3, 4-7, 8-B or C-F). The jumper plug settings must match the line interface module types installed on the MVC. See Figure A-2 for a configuration diagram.

RS232C or MIL 188C Line Modules

The jumpers on the I/O panel must be strapped for the RS-232 setting. Install jumper plugs at E1, E7, E9 & E13 for Lines 0-3 or 8-B. Install jumper plugs at E3, E5, E11 & E15 for Lines 4-7 or C-F. See Figure A-2.

RS-232C/MIL 188C Pinout		
DB-25 Pin	**Signal**	**Description**
1	CGND	Chassis Ground
2	TXD	Transmit Data
3	RXD	Receive Data
4	RTS	Request To Send
5	CTS	Clear To Send
6	DSR	Data Set Ready
7	SGND	Signal Ground
8	DCD	Data Carrier Detect
20	DTR	Data Terminal Ready
22	RI	Ring Indicate

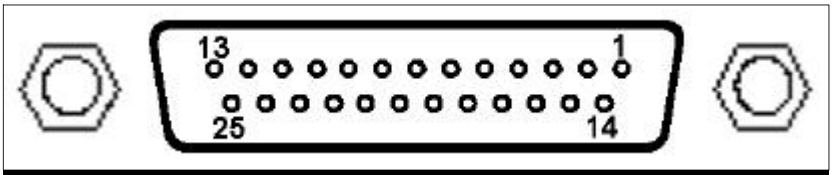


Figure A-1 – DB-25 Connector Pinout

RS422 or 423 Line Modules

The jumpers on the I/O panel must be strapped for the RS-422 setting. Install jumper plugs at E2, E8, E10 & E14 for Lines 0-3 or 8-B. Install jumper plugs at E4, E6, E12 & E16 for Lines 4-7 or C-F. See Figure A-2.

RS-422/423 Pinout		
DB-25 Pin	**Signal**	**Description**
1	CGND	Chassis Ground
2	TXDA	Transmit Data A
3	RXDA	Receive Data A
4	RTSA	Request To Send A
5	CTSA	Clear To Send A
7	SGND	Signal Ground
13	CTSB	Clear To Send B
14	TXDB	Transmit Data B
16	RXDB	Receive Data B
19	RTSB	Request To Send B

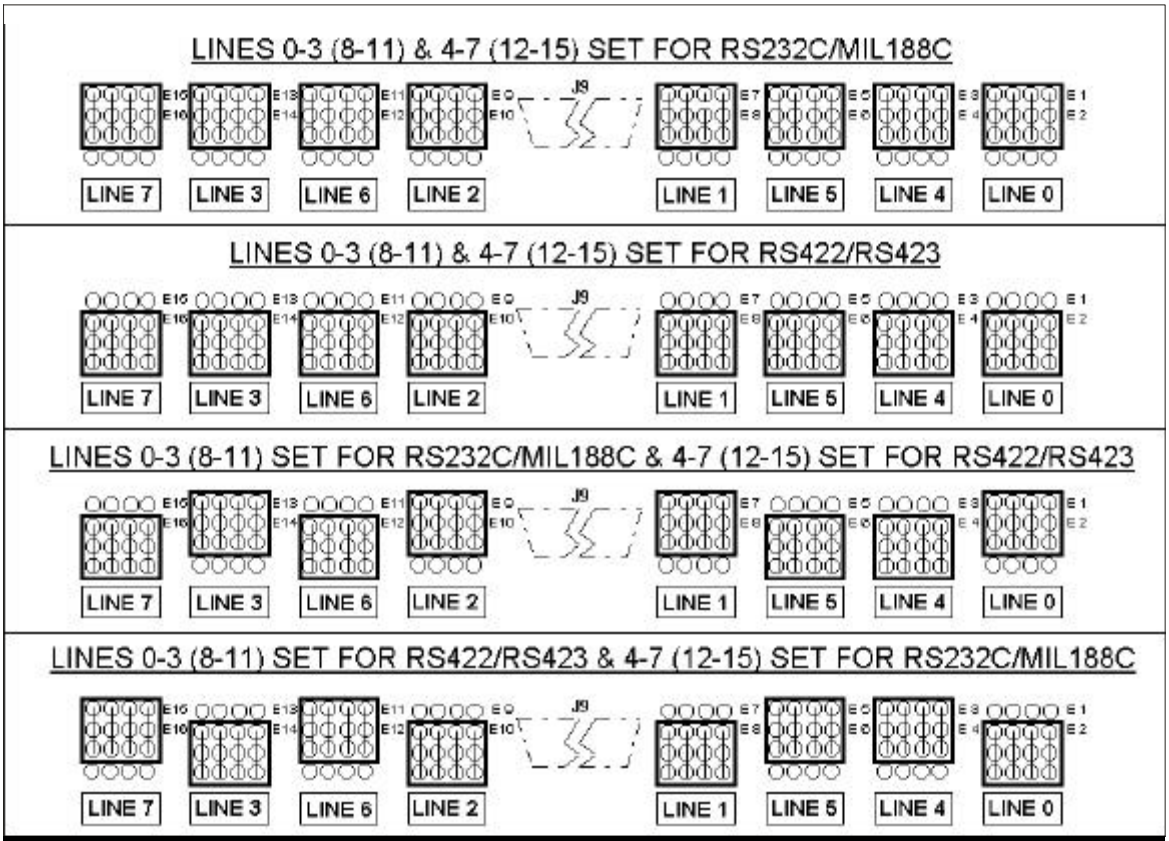


Figure A - 2 – DB-25 I/O Panel Configuration

RS485 Line Modules

The jumpers on the I/O panel must be strapped for the RS-422 setting. Install jumper plugs at E2, E8, E10 & E14 for Lines 0-3 or 8-B. Install jumper plugs at E4, E6, E12 & E16 for Lines 4-7 or C-F. See Figure A-2.

The RS485 line modules include 120 ohm termination resistors for the data lines. They must be connected externally. See the example drawing in Figure A-3 below.

RS485 Pinout		
DB-25 Pin	**Signal**	**Description**
1	CGND	Chassis Ground
2	TRDA	Transmit/Receive Data A
3	RTA	120 Ohm Terminator Resistor A
7	SGND	Signal Ground
14	TRDB	Transmit/Receive Data B
16	RTB	120 Ohm Terminator Resistor B

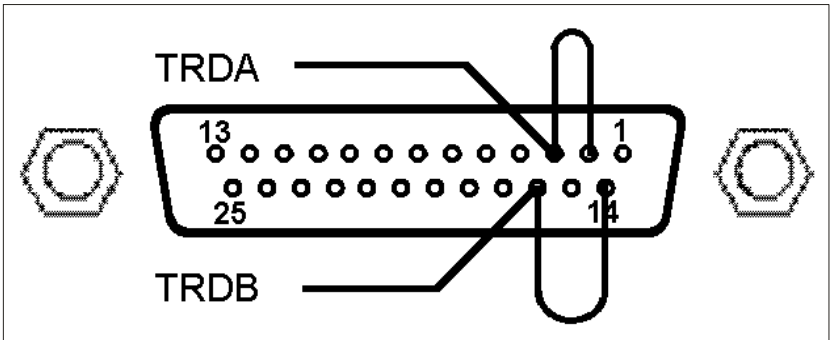


Figure A-3 – RS485 Connection Example

A.2 16 Line RJ45 I/O Panel

Unlike the DB-25 I/O panel, the RJ45 panel does not use configuration jumpers. See Figure A-4 for a pin number identification drawing of the RJ45 connector.

Mating 10-pin connectors are Stewart Connector Systems part number 937-SP-301010 or equivalent.

RS232C or MIL 188C Line Modules

RS-232/MIL 188C Pinout		
RJ45 Pin	**Signal**	**Description**
1	RI	Ring Indicate
2	DSR	Data Set Ready
3	RTS	Request To Send
4	CGND	Chassis Ground
5	TXD	Transmit Data
6	RXD	Receive Data
7	SGND	Signal Ground
8	CTS	Clear To Send
9	DTR	Data Terminal Ready
10	DCD	Data Carrier Detect

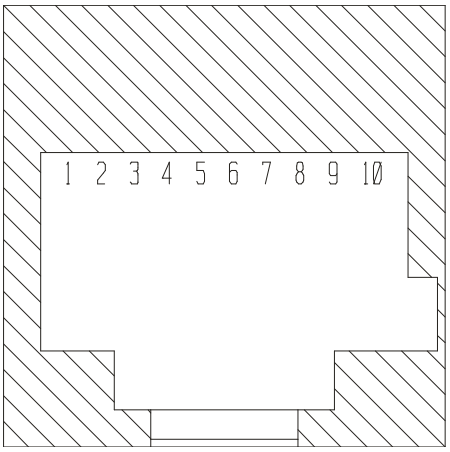


Figure A - 4 – RJ45 Connector Front View

RS422 or 423 Line Modules

RS-422/423 Pinout		
RJ45 Pin	**Signal**	**Description**
1	CTSB	Clear To Send B
2	RTSB	Request To Send B
3	RTSA	Request To Send A
4	CGND	Chassis Ground
5	TXDA	Transmit Data A
6	RXDA	Receive Data A
7	SGND	Signal Ground
8	CTSA	Clear To Send A
9	RXDB	Receive Data B
10	TXDB	Transmit Data B

RS485 Line Modules

The RS485 line modules include 120 ohm termination resistors for the data lines. They must be connected externally.

RS485 Pinout		
RJ45 Pin	**Signal**	**Description**
4	CGND	Chassis Ground
5	TRDA	Transmit/Receive Data A
6	RTA	120 Ohm Terminator Resistor A
7	SGND	Signal Ground
9	RTB	120 Ohm Terminator Resistor B
10	TRDB	Transmit/Receive Data B

A.3 Serial I/O Connectors

The 80-pin high-density serial I/O interface connectors used on the MVC are AMP part number 749075-8. The part number of the mating connector is 749621-8. The angled backshell part number is 749205-1. A pin number identification drawing is presented in Figure A-4. Signal pin number information is presented in the table below.

J1/J2 Pin	RS232/MIL188C Interface		RS422/423 Interface		RS485 Interface	
	Name	Description	Name	Description	Name	Description
1	CGND	Protective Ground	CGND	Protective Ground	CGND	Protective Ground
2	TXD0	Line 0/8 Transmit Data	TXDA	Line 0/8 Transmit Data A	TRDA	Line 0/8 Data A
3	RXD0	Line 0/8 Receive Data	RXDA	Line 0/8 Receive Data A	RTA	Line 0/8 Term. Resistor A
4	RTS1	Line 0/8 Request To Send	RTSA	Line 0/8 Request To Send A	—	Not used
5	CTS1	Line 0/8 Clear To Send	CTSA	Line 0/8 Clear To Send A	—	Not used
6	DSR1	Line 0/8 Data Set Ready	RTSB	Line 0/8 Request To Send B	—	Not used
7	SGND	Signal Ground	SGND	Signal Ground	SGND	Signal Ground
8	DTR1	Line 0/8 Data Terminal Ready	RXDB	Line 0/8 Receive Data B	RTB	Line 0/8 Term. Resistor B
9	CARR1	Line 0/8 Data Carrier Detect	TXDB	Line 0/8 Transmit Data B	TRDB	Line 0/8 Data B
10	RING1	Line 0/8 Ring Indicate	CTSB	Line 0/8 Clear To Send B	—	Not used
11	CGND	Protective Ground	CGND	Protective Ground	CGND	Protective Ground
12	TXD0	Line 1/9 Transmit Data	TXDA	Line 1/9 Transmit Data A	TRDA	Line 1/9 Data A
13	RXD0	Line 1/9 Receive Data	RXDA	Line 1/9 Receive Data A	RTA	Line 1/9 Term. Resistor A
14	RTS1	Line 1/9 Request To Send	RTSA	Line 1/9 Request To Send A	—	Not used
15	CTS1	Line 1/9 Clear To Send	CTSA	Line 1/9 Clear To Send A	—	Not used
16	DSR1	Line 1/9 Data Set Ready	RTSB	Line 1/9 Ready To Send B	—	Not used
17	SGND	Signal Ground	SGND	Signal Ground	SGND	Signal Ground
18	DTR1	Line 1/9 Data Terminal Ready	RXDB	Line 1/9 Receive Data B	RTB	Line 1/9 Term. Resistor B
19	CARR1	Line 1/9 Data Carrier Detect	TXDB	Line 1/9 Transmit Data B	TRDB	Line 1/9 Data B
20	RING1	Line 1/9 Ring Indicate	CTSB	Line 1/9 Clear To Send B	—	Not used
21	CGND	Protective Ground	CGND	Protective Ground	CGND	Protective Ground
22	TXD0	Line 2/A Transmit Data	TXDA	Line 2/A Transmit Data A	TRDA	Line 2/A Data A
23	RXD0	Line 2/A Receive Data	RXDA	Line 2/A Receive Data A	RTA	Line 2/A Term. Resistor A
24	RTS1	Line 2/A Request To Send	RTSA	Line 2/A Request To Send A	—	Not used
25	CTS1	Line 2/A Clear To Send	CTSA	Line 2/A Clear To Send A	—	Not used
26	DSR1	Line 2/A Data Set Ready	RTSB	Line 2/A Ready To Send B	—	Not used
27	SGND	Signal Ground	SGND	Signal Ground	SGND	Signal Ground
28	DTR1	Line 2/A Data Terminal Ready	RXDB	Line 2/A Receive Data B	RTB	Line 2/A Term. Resistor B
29	CARR1	Line 2/A Data Carrier Detect	TXDB	Line 2/A Transmit Data B	TRDB	Line 2/A Data B
30	RING1	Line 2/A Ring Indicate	CTSB	Line 2/A Clear To Send B	—	Not used
31	CGND	Protective Ground	CGND	Protective Ground	CGND	Protective Ground
32	TXD0	Line 3/B Transmit Data	TXDA	Line 3/B Transmit Data A	TXDA	Line 3/B Data A
33	RXD0	Line 3/B Receive Data	RXDA	Line 3/B Receive Data A	RTA	Line 3/B Term. Resistor A
34	RTS1	Line 3/B Request To Send	RTSA	Line 3/B Request To Send A	—	Not used
35	CTS1	Line 3/B Clear To Send	CTSA	Line 3/B Clear To Send A	—	Not used
36	DSR1	Line 3/B Data Set Ready	RTSB	Line 3/B Ready To Send B	—	Not used
37	SGND	Signal Ground	SGND	Signal Ground	SGND	Signal Ground
38	DTR1	Line 3/B Data Terminal Ready	RXDB	Line 3/B Receive Data B	RTB	Line 3/B Term. Resistor B
39	CARR1	Line 3/B Data Carrier Detect	TXDB	Line 3/B Transmit Data B	TRDB	Line 3/B Data B
40	RING1	Line 3/B Ring Indicate	CTSB	Line 3/B Clear To Send B	—	Not used

J1/J2 Pin	RS232/MIL188C Interface		RS422/423 Interface		RS485 Interface	
	Name	Description	Name	Description	Name	Description
41	CGND	Protective Ground	CGND	Protective Ground	CGND	Protective Ground
42	TXD0	Line 4/C Transmit Data	TXDA	Line 4/C Transmit Data A	TRDA	Line 4/C Data A
43	RXD0	Line 4/C Receive Data	RXDA	Line 4/C Receive Data A	RTA	Line 4/C Term. Resistor A
44	RTS1	Line 4/C Request To Send	RTSA	Line 4/C Request To Send A	—	Not used
45	CTS1	Line 4/C Clear To Send	CTSA	Line 4/C Clear To Send A	—	Not used
46	DSR1	Line 4/C Data Set Ready	RTSB	Line 4/C Ready To Send B	—	Not used
47	SGND	Signal Ground	SGND	Signal Ground	SGND	Signal Ground
48	DTR1	Line 4/C Data Terminal Ready	RXDB	Line 4/C Receive Data B	RTB	Line 4/C Term. Resistor B
49	CARR1	Line 4/C Data Carrier Detect	TXDB	Line 4/C Transmit Data B	TRDB	Line 4/C Data B
50	RING1	Line 4/C Ring Indicate	CTSB	Line 4/C Clear To Send B	—	Not used
51	CGND	Protective Ground	CGND	Protective Ground	CGND	Protective Ground
52	TXD0	Line 5/D Transmit Data	TXDA	Line 5/D Transmit Data A	TRDA	Line 5/D Data A
53	RXD0	Line 5/D Receive Data	RXDA	Line 5/D Receive Data A	RTA	Line 5/D Term. Resistor A
54	RTS1	Line 5/D Request To Send	RTSA	Line 5/D Request To Send A	—	Not used
55	CTS1	Line 5/D Clear To Send	CTSA	Line 5/D Clear To Send A	—	Not used
56	DSR1	Line 5/D Data Set Ready	RTSB	Line 5/D Ready To Send B	—	Not used
57	SGND	Signal Ground	SGND	Signal Ground	SGND	Signal Ground
58	DTR1	Line 5/D Data Terminal Ready	RXDB	Line 5/D Receive Data B	RTB	Line 5/D Term. Resistor B
59	CARR1	Line 5/D Data Carrier Detect	TXDB	Line 5/D Transmit Data B	TRDB	Line 5/D Data B
60	RING1	Line 5/D Ring Indicate	CTSB	Line 5/D Clear To Send B	—	Not used
61	CGND	Protective Ground	CGND	Protective Ground	CGND	Protective Ground
62	TXD0	Line 6/E Transmit Data	TXDA	Line 6/E Transmit Data A	TRDA	Line 6/E Data A
63	RXD0	Line 6/E Receive Data	RXDA	Line 6/E Receive Data A	RTA	Line 6/E Term. Resistor A
64	RTS1	Line 6/E Request To Send	RTSA	Line 6/E Request To Send A	—	Not used
65	CTS1	Line 6/E Clear To Send	CTSA	Line 6/E Clear To Send A	—	Not used
66	DSR1	Line 6/E Data Set Ready	RTSB	Line 6/E Ready To Send B	—	Not used
67	SGND	Signal Ground	SGND	Signal Ground	SGND	Signal Ground
68	DTR1	Line 6/E Data Terminal Ready	RXDB	Line 6/E Receive Data B	RTB	Line 6/E Term. Resistor B
69	CARR1	Line 6/E Data Carrier Detect	TXDB	Line 6/E Transmit Data B	TRDB	Line 6/E Data B
70	RING1	Line 6/E Ring Indicate	CTSB	Line 6/E Clear To Send B	—	Not used
71	CGND	Protective Ground	CGND	Protective Ground	CGND	Protective Ground
72	TXD0	Line 7/F Transmit Data	TXDA	Line 7/F Transmit Data A	TRDA	Line 7/F Data A
73	RXD0	Line 7/F Receive Data	RXDA	Line 7/F Receive Data A	RTA	Line 7/F Term. Resistor A
74	RTS1	Line 7/F Request To Send	RTSA	Line 7/F Request To Send A	—	Not used
75	CTS1	Line 7/F Clear To Send	CTSA	Line 7/F Clear To Send A	—	Not used
76	DSR1	Line 7/F Data Set Ready	RTSB	Line 7/F Ready To Send B	—	Not used
77	SGND	Signal Ground	SGND	Signal Ground	SGND	Signal Ground
78	DTR1	Line 7/F Data Terminal Ready	RXDB	Line 7/F Receive Data B	RTB	Line 7/F Term. Resistor B
79	CARR1	Line 7/F Data Carrier Detect	TXDB	Line 7/F Transmit Data B	TRDB	Line 7/F Data B
80	RING1	Line 7/F Ring Indicate	CTSB	Line 7/F Clear To Send B	—	Not used

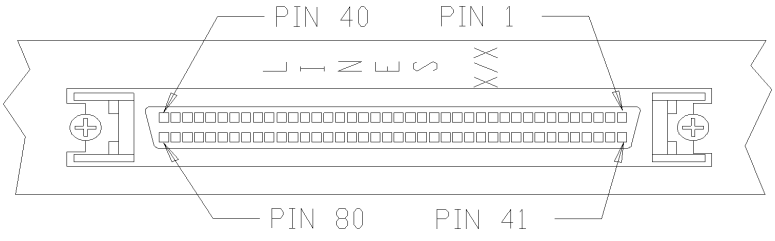


Figure A-5 – High-Density 80 Pin Connector

A.4 Internal Printer Cable

Pin	Name	Description	P2 Pin
1	DSTROB0	Data Strobe	C1
2	PDB1	Data Bit 1	C2
3	PDB2	Data Bit 2	C3
4	PDB3	Data Bit 3	C4
5	PDB4	Data Bit 4	C5
6	PDB5	Data Bit 5	C6
7	PDB6	Data Bit 6	C7
8	PDB7	Data Bit 7	C8
9	PDB8	Data Bit 8	C9
10	ACK0	Acknowledge	C10
11	BSY	Busy	C11
12	PE	Paper End	C12
13	SLCT	Select	C13
14	AUTOFD	Auto Feed	C14
15	—	Not Used	
16	—	Not Used	
18	—	Not Used	
19	DS Return	Data Strobe Return	A1
20	PDB1 Return	Data Bit 1 Return	A2
21	PDB2 Return	Data Bit 2 Return	A3
22	PDB3 Return	Data Bit 3 Return	A4
23	PDB4 Return	Data Bit 4 Return	A5
24	PDB5 Return	Data Bit 5 Return	A6
25	PDB6 Return	Data Bit 6 Return	A7
26	PDB7 Return	Data Bit 7 Return	A8
27	PDB8 Return	Data Bit 8 Return	A9
28	ACK Return	Acknowledge Return	A10
29	BSY Return	Busy Return	A11
30	—	Not Used	
31	INIT	Initialize	A13
32	Fault	Printer Error	A14
33	Undefined	Not Used	
34	Undefined	Not Used	
35	Undefined	Not Used	
36	SLCTN	Select In	A18

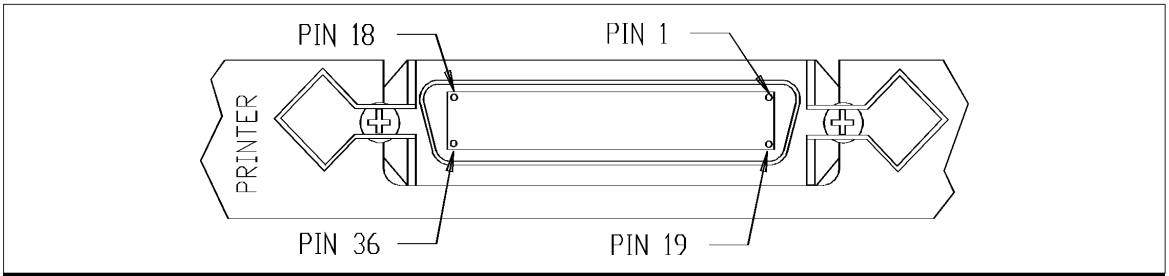


Figure A-6 – Centronics Printer Connector

A.5 VMEbus Interface Signals

VMEbus Interface Signals

Name	Pin	Description
A01	P1A-30	Address Bus
A02	P1A-29	Address Bus
A03	P1A-28	Address Bus
A04	P1A-27	Address Bus
A05	P1A-26	Address Bus
A06	P1A-25	Address Bus
A07	P1A-24	Address Bus
A08	P1C-30	Address Bus
A09	P1C-29	Address Bus
A10	P1C-28	Address Bus
A11	P1C-27	Address Bus
A12	P1C-26	Address Bus
A13	P1C-25	Address Bus
A14	P1C-24	Address Bus
A15	P1C-23	Address Bus
A16	P1C-22	Address Bus
A17	P1C-21	Address Bus
A18	P1C-20	Address Bus
A19	P1C-19	Address Bus
A20	P1C-18	Address Bus
A21	P1C-17	Address Bus
A22	P1C-16	Address Bus
A23	P1C-15	Address Bus
A24	P2B-4	Address Bus
A24	P2B-5	Address Bus
A26	P2B-6	Address Bus
A27	P2B-7	Address Bus
A28	P2B-8	Address Bus
A29	P2B-9	Address Bus
A30	P2B-10	Address Bus
A31	P2B-11	Address Bus
D00	P1A-1	Data Bus
D01	P1A-2	Data Bus
D02	P1A-3	Data Bus
D03	P1A-4	Data Bus
D04	P1A-5	Data Bus
D05	P1A-6	Data Bus
D06	P1A-7	Data Bus
D07	P1A-8	Data Bus
D08	P1C-1	Data Bus
D09	P1C-2	Data Bus
D10	P1C-3	Data Bus
D11	P1C-4	Data Bus
D12	P1C-5	Data Bus
D13	P1C-6	Data Bus
D14	P1C-7	Data Bus
D15	P1C-8	Data Bus

VMEbus Interface Signals

Name	Pin	Description
D16	P2B-14	Data Bus
D17	P2B-15	Data Bus
D18	P2B-16	Data Bus
D19	P2B-17	Data Bus
D20	P2B-18	Data Bus
D21	P2B-19	Data Bus
D22	P2B-20	Data Bus
D23	P2B-21	Data Bus
D24	P2B-23	Data Bus
D25	P2B-24	Data Bus
D26	P2B-25	Data Bus
D27	P2B-26	Data Bus
D28	P2B-27	Data Bus
D29	P2B-28	Data Bus
D30	P2B-29	Data Bus
D31	P2B-30	Data Bus
Strobes, Active Low (*=0)		
AS*	P1A-18	Address Strobe
DS0*	P1A-13	Data Strobe Zero
DS1*	P1A-12	Data Strobe One
DTACK*	P1A-16	Data Transfer Acknowledge
Master Mode Signals, Active Low		
BBSY*	P1B-1	Bus Busy
BCLR*	P1B-2	Bus Clear
BERR*	P1C-11	Bus Error
BG0IN*	P1B-4	Bus Grant In Zero
BG1IN*	P1B-6	Bus Grant In One
BG2IN*	P1B-8	Bus Grant In Two
BG3IN*	P1B-10	Bus Grant In Three
BG0OUT*	P1B-5	Bus Grant Out Zero
BG1OUT*	P1B-7	Bus Grant Out One
BG2OUT*	P1B-9	Bus Grant Out Two
BG3OUT*	P1B-11	Bus Grant Out Three
BR0*	P1B-12	Bus Request Zero
BR1*	P1B-13	Bus Request One
BR2*	P1B-14	Bus Request Two
BR3*	P1B-15	Bus Request Three
Interrupt Lines, Active Low		
IRQ1*	P1B-30	Interrupt Request One
IRQ2*	P1B-29	Interrupt Request Two
IRQ3*	P1B-28	Interrupt Request Three
IRQ4*	P1B-27	Interrupt Request Four
IRQ5*	P1B-26	Interrupt Request Five
IRQ6*	P1B-25	Interrupt Request Six
IRQ7*	P1B-24	Interrupt Request Seven
IACK*	P1A-20	Interrupt Acknowledge
IACKIN*	P1A-21	Interrupt Acknowledge In
IACKOUT*	P1A-22	Interrupt Acknowledge Out

VMEbus Interface Signals

Name	Pin	Description
Miscellaneous Signals, Active Low		
LWORD*	P1C-13	Longword, 32 Bit XFER
WRITE*	P1A-14	Write, High For Read
SYSRESET*	P1C-12	System Reset
Power		
A5V	P1A-32	Plus Five Volts DC
A5V	P1B-32	Plus Five Volts DC
A5V	P1C-32	Plus Five Volts DC
A5V	P2B-1	Plus Five Volts DC
A5V	P2B-13	Plus Five Volts DC
A5V	P2B-32	Plus Five Volts DC
A12V	P1C-3	Plus Twelve Volts DC
-12V	P1A-31	Minus Twelve Volts DC
GND	P1A-9	Signal Ground
GND	P1A-11	Signal Ground
GND	P1A-15	Signal Ground
GND	P1A-17	Signal Ground
GND	P1A-19	Signal Ground
GND	P1B-20	Signal Ground
GND	P1B-23	Signal Ground
GND	P2B-2	Signal Ground
GND	P2B-12	Signal Ground
GND	P2B-22	Signal Ground
GND	P2B-31	Signal Ground
GND	P1C-9	Signal Ground

This page has no technical content.