ARTISAN®
TECHNOLOGY GROUP

Your **definitive** source for quality pre-owned equipment.

Artisan Technology Group
(217) 352-9330 | sales@artisantg.com | artisantg.com

Full-service, independent repair center
with experienced engineers and technicians on staff.

We buy your excess, underutilized, and idle equipment
along with credit for buybacks and trade-ins.

Custom engineering
so your equipment works exactly as you specify.

- Critical and expedited services
- In stock / Ready-to-ship
- Leasing / Rentals / Demos
- ITAR-certified secure asset solutions

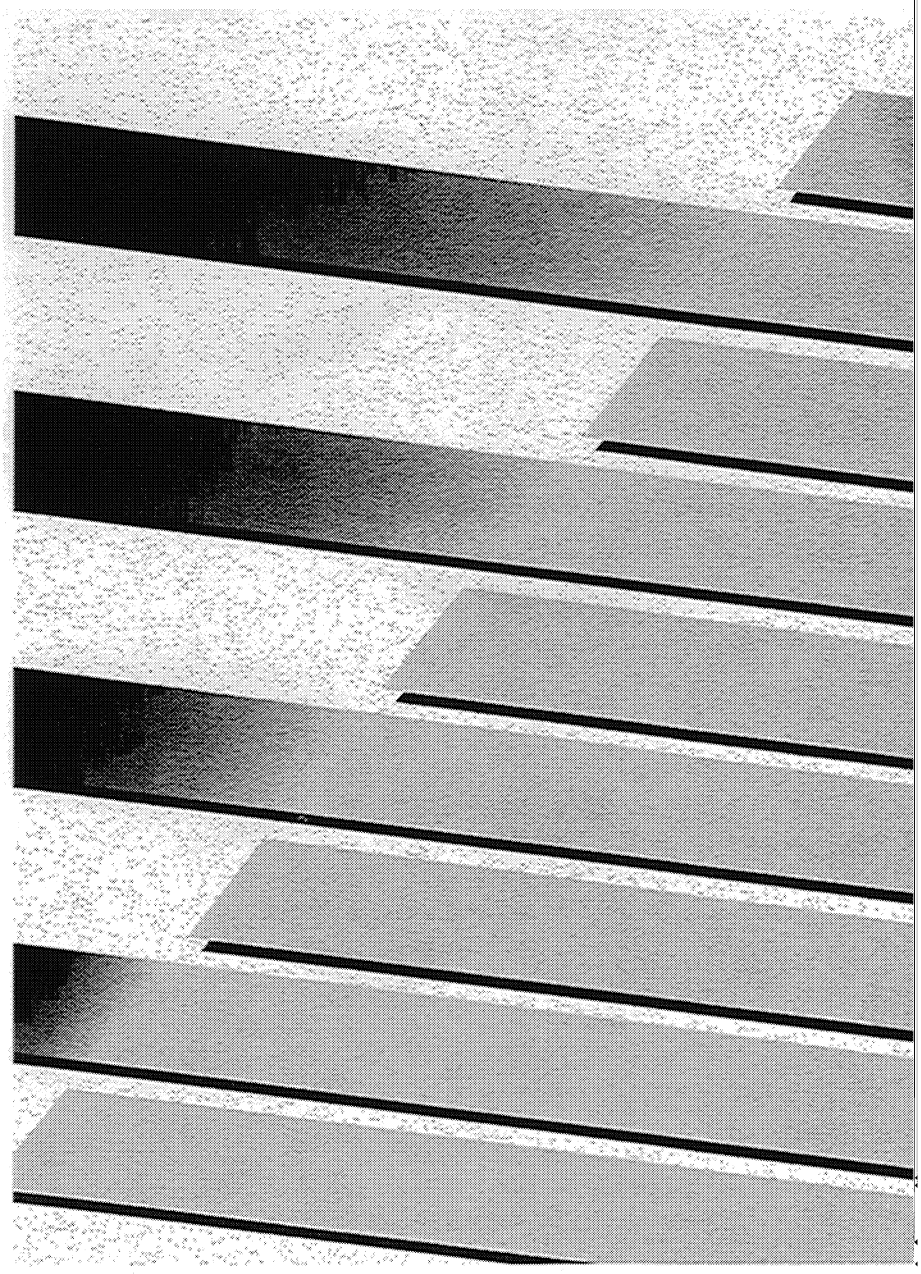
Expert team | Trust guarantee | 100% satisfaction

All trademarks, brand names, and brands appearing herein are the property of their respective owners.

Find the *Rockwell / Allen-Bradley PLC-5/V80B* at our website: ***Click HERE***

PLC-5/VME Programmable Control
(Cat. No. 6008-LTV)

Interface Manual



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Important User Information

Because of the variety of uses for this equipment and because of the differences between this solid state equipment and electromechanical equipment, the user of and those responsible for applying this equipment must satisfy themselves as to the acceptability of each application and use of the equipment. In no event will Allen-Bradley Company, Inc. be responsible or liable for indirect or consequential damages resulting from the use or application of this equipment.

The illustrations, charts, and layout examples shown in this manual are intended solely to illustrate the text of this manual. Because of the many variables and requirements associated with any particular installation, Allen-Bradley Company, Inc. cannot assume responsibility or liability for actual use based upon the illustrative uses and applications.

No patent liability is assumed by Allen-Bradley Company, Inc. with respect to use of information, circuits, equipment or software described in this text.

Reproduction of the contents of this manual, in whole or in part, without written permission of the Allen-Bradley Company, Inc. is prohibited.

© 1988 Allen-Bradley Company, Inc.

PLC is a registered trademark of Allen-Bradley Company, Inc.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Table of Contents

Chapter 1	Using Your Manual	
	Purpose of Manual	1-1
	Audience	1-1
	Suggestions for Learning this Product	1-1
Chapter 2	Overview of 6008-LTV	
	Features	2-1
	Mechanical Package	2-2
	Electrostatic Damage	2-3
	System Description	2-3
	Operation as a Programmable Controller	2-5
	I/O Subsystem	2-5
	Block Transfer	2-5
	Peer Communications Link	2-6
	Parity Checking	2-6
	VMEbus/6008-LTV Interface	2-7
	VMEbus Compliance Capabilities	2-8
Chapter 3	Getting Started with Copy Commands	
	Introduction	3-1
	Definitions	3-1
	Tutorial Overview	3-3
	Electrostatic Damage	3-4
	Setting Switches	3-5
	Connecting Equipment	3-6
	Defining the Transfer	3-7
	Entering Data	3-8
	Entering the PCL Station Number	3-8
	Creating the VME Status File	3-9
	Entering Copy Command Parameters	3-9
	Creating the LTV Data File	3-11
	Entering Data in LTV Data File	3-12
	Initiating Copy Commands and Observing the Results	3-13
	Conclusion	3-14
Chapter 4	Programming Copy Commands	
	Introduction	4-1
	Initiating Copy Commands from the LTV's Ladder Program	4-1
	Enable Bits	4-1
	Ladder Logic	4-1
	Address of Examine-On Instruction	4-2
	Address of Output Bit	4-2
	Changing Command Parameters from the LTV's Ladder Program	4-3
	Initiating and Changing Copy Commands from the Host CPU's Driver Program	4-5
	Driver Program	4-5
Chapter 5	VMEbus Global Memory	
	Overview of Global Memory	5-1
	Slave 0	5-1
	Slave 1	5-1
	External Memory Module	5-1

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Slave 0 Memory 5-3
Module ID Data Block 5-4
Bit/Byte Descriptions of Slave 0 Memory 5-5
Configuring Slave 0 Global Memory 5-8
Configuring Slave 1 or VMEbus Options 5-8
Configuring Slave 1, If Using It 5-8
Configuring VMEbus Options 5-8
Selecting Slave 1 Address Modifiers 5-10
Address Modifier Codes (Table 5.1) 5-11
Starting Address of Slave 0 Memory 5-12
Address Modifiers of Slave 0 Memory 5-13

Chapter 6 Commands, Command Blocks, & Sequences
VMEbus Commands 6-1
Copy Commands 6-2
Copy Command Block 6-2
Command Block Bit/Word Descriptions 6-2
Copy Command Sequence 6-8
Transferring Data 6-10
Selective Commands 6-11
Selective Command Block 6-11
Command Block Bit/Word Descriptions 6-12
Selective Command Sequence 6-13
Command Block Error Codes 6-15

Chapter 7 Selective Commands (Part I)
Introduction 7-1
Packet Format 7-2
Header Bit/Byte Descriptions 7-3
Selective Commands 7-5
Echo 7-6
Identify LTV, Report Status 7-7
Read-Modify-Write 7-9
Typed Read 7-11
Typed Write 7-13
Data Types 7-15
Data Type Field 7-15
Example Data Types 7-16
Integer 7-16
Floating Point 7-17
Control Structure 7-17
Counter 7-18
Timer 7-18
Array 7-19
Character String 7-20

Chapter 8 Selective Commands (Part II)
Introduction 8-1
Criteria for Upload/Download Commands 8-1
Verification 8-2
Completion 8-2
Command Discrimination 8-3
Upload/download Compatibility 8-3
Identify LTV, Report Status 8-4
Set Processor Mode 8-6
Upload All Request 8-7
Read Bytes Physical 8-9
Upload Completed 8-10

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Download All Physical Request 8-11
Writes Byte Physical 8-12
Download Completed 8-13

Chapter 9 PC and VME Status Files
Introduction 9-1
PC Status File 9-1
Screen Display of PC Status File 9-2
Memory Map of PC Status File 9-3
Bit/Word Descriptions 9-4
VME Status File 9-5
Screen Displays 9-5
VME Status 9-5
VME Copy Configuration 9-6
Memory Map of VME Status File 9-7
Word/Bit Descriptions 9-8
Error Codes for Copy Commands 9-12

Chapter 10 Switch Settings
Switch Assembly Locations 10-1
SW1 8-Switch Assembly 10-2
PCL Station Number 10-2
Memory Protection 10-4
SW2 8-Switch Assembly 10-4
SW3 4-Switch Assembly 10-4
PLC Terminating Resistor 10-4
Remote I/O Terminating Resistor 10-4
SW4 8-Switch Assembly 10-5
Starting Address of Slave 0 10-5
Address Modifiers of Slave 0 10-6
SW5 4-Switch Assembly 10-6
VMEbus Arbitration Level 10-6

Chapter 11 Installation Considerations
PLC-5 Family Assembly and Installation Manual 11-1
Power Requirement 11-2
Position in the VMEbus Subrack 11-2
Grounding 11-2
Power Monitor Module 11-2
Setup Tips 11-3

Chapter 12 Troubleshooting
Introduction 12-1
LED Status 12-1
Troubleshooting with LED Indicators 12-3
Driver Program Error Codes 12-4

Appendix A Example Driver Program
Set Processor Mode A-1

Appendix B Timing Considerations
Scan Timing B-1
Timing Considerations for Copy Commands B-2

Index I-1

Worksheets

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Purpose of Manual

The purpose of this manual is to acquaint you with the PLC-5/VME industry bus programmable controller (cat. no. 6008-LTV). We explain its operation so that you can use it in a VMEbus system. You must develop a software driver program that lets your host CPU or a VMEbus master write and read data to and from the LTV. You must also write a ladder diagram program so your LTV can monitor and control the I/O of your control system. This manual and related publications help you to do this.

IMPORTANT: We recommend that you follow the suggestions below to help you learn the LTV and operate it in a VMEbus system in the shortest possible time.

Audience

We assume that the audience for this product has programming and application experience in two primary areas: VMEbus and Programmable Controller products.

To control communication between the VMEbus master and the LTV, you must develop a software driver program in a language such as C or Assembly. Experience in systems development and integration, and in writing software programs for VMEbus products using 68000 family microprocessors would be very helpful. Use this manual.

You must design an application control program in ladder diagram logic for the LTV to monitor its inputs, control its outputs, and manipulate data. Experience with Allen-Bradley programmable controllers and programming terminals that use application software would be very helpful. Refer to PLC-5 Family Manuals.

Suggestions for Learning this Product

1) Review this manual and the following manuals to get an overall feeling for their organization and content:

   PLC-5 Family Processor Manual, publication 1785-6.8.2. This manual describes data table and memory organization, (excluding VMEbus dual ported global memory) instruction set for ladder diagram programming, communication concepts, and status/debugging tools.

   PLC-5 Programming Software User's Manual, Release Number 2.1 (or later), publication 6200-6.5.5 dated August 1987 and 6200-6.5.5-DU1 dated December 1987. This manual and update tell you how to use your

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

programming terminal and its PLC-5 programmable controller software to program (ladder diagram) the LTV, examine its data table and non-VMEbus memory, change its configuration, set its operating mode, and document the above.

Industrial Terminal User's Manual, publication 1784-6.5.1. This manual tells you how to install and start up your 1784-T50 programming terminal.

PLC-5 Family Assembly and Installation Manual, publication 1785-6.6.1. This manual shows you how to connect Allen-Bradley I/O products to your LTV, how to set switches, and how to relate physical I/O terminals to I/O image table locations in LTV memory.

User's manuals and data sheets of the Bulletin 1771 I/O products that you will use with your LTV. They will help you interface your LTV to the real-time signals of your plant operation.

Hardware manuals of your VMEbus computer system, modules, chassis, power supply, transition panels, and accessories. They describe jumper settings to configure modules (other than A-B), memory mappings in the system address space, installation and configuration parameters, and troubleshooting information.

Software manuals for your VMEbus system's operating system, drivers, compilers, assemblers, interpreters, debuggers, linkers, editors, data files, and other software. They will help you write the software for your application, debug, test, and place the final software into operation.

2) Determine the memory map of your system. Reserve a 1024 byte space of free addressing in the short global range from 0000 to FFFF. Set jumpers and switches of your various VMEbus modules to meet these mapping requirements. Check operating system manuals for specific configuration requirements. Install the modules in the VMEbus subrack. Verify that VMEbus backplane jumpers are correct.

3) Learn how to power up and boot your VMEbus system. Verify that the various ports and system utilities are operational. Using a system debugger, store values in the LTV's Slave 0 memory (starting at 52H bytes above the starting address of Slave 0). Verify that values are correctly read and written.

4) Connect the programming terminal to the LTV. Ensure that the 6200 Series software is loaded and operational. Practice creating and filling program and data table files, and examining status files.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

5) Determine whether your application requires copy commands or selective commands (page 3-1). Then determine what your software driver program must do to transfer data with either type of command (chapters 6 and 13).

6) Determine your global memory requirements and choose either the LTV's internal Slave 0 (1K bytes) or Slave 1 (4K bytes), or an external global memory module. Parameters for configuring global memory are found:

For	Refer to pages
Slave 0	5-1, 5-10, 5-11
Slave 1	5-1, 5-6, 5-12
external	5-2, 5-6, 5-12

7) If you select Slave 1, write a driver program to configure it.

8a) If your application requires copy commands, you may want to do the copy command tutorial in chapter 3. Then program the transfer of a file after studying the first half of chapter 6 followed by chapter 4.

8b) If your application requires selective commands, study chapters 6, 7, and 8 thoroughly. Review the sample programs in chapter 13. Start with a simple one-pass test to read and write data to an integer or floating-point data file. Expand your driver program in steps. We suggest that you avoid interrupts unless you are very familiar with your operating system, host CPU, and how to write interrupt-driven device drivers with your system.

9) Try to make your program crashproof by testing it thoroughly. Verify its correct operation under as many different input conditions as possible including error conditions.

- With loops, test the loop for zero, one, two and a maximum number of iterations.
- Make sure all possible input cases are covered, including no input. Verify each error by simulation.
- Check that the appropriate status is returned or the appropriate message is printed.
- Verify that timeouts and restarts work as desired.
- Check each subroutine or function against its calls and verify that the same number and types of arguments are passed. Check for buffer overflows.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

- Check memory allocations and returns, port and file access operations.
- Watch for stack creep, a condition where a program crashes the operating system for no apparent reason after many stack operations.
- Check for word, long word, and float alignment restrictions and for holes in data structures created by the compiler, interpreter, and/or assembler.
- Check pointer operations (indirect addressing) very carefully.

10) We suggest that you write modular code that performs only one given function in each block, and that you thoroughly document your program development effort by keeping a log. Document your code and explain its operation. Document your data structures with symbol names, contents, data format, and storage size.

    Use the forms at the back of this manual to help you configure the LTV and map your data files.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Features

The 6008-LTV (abbreviated LTV) is equivalent to a PLC-5/15 (Series B) programmable controller with a VMEbus (Revision C.1) interface. It resides in a VMEbus subrack and can be controlled by a host CPU or any other VMEbus master. Also, it can operate solely as a PLC-5/15 processor with no VME interface, drawing only its power from the VME subrack.

Features include:

- Advanced PLC-5 instruction set for ladder diagram programming.
- Ladder programs interchangeable with those of PLC-5/15 programmable controllers elsewhere on the plant floor.
- Programmed with either the 1784-T50 Industrial Terminal and PLC-5 Programming Terminal Software (6201-PLC5), or an IBM PC/AT or compatible computer with a combination of 1784-KTK1 PLC Interface Card and PLC-5 Programming Terminal Software (6211-PLC-5).
- Remote I/O scanner for distributing four remote I/O chassis up to 10,000 feet (total of 512 I/O points). A complementary I/O configuration can double the number of I/O points.
- Serial interface to A-B Peer Communication Link (PCL) providing a network of up to 10 stations (optimum), 64 stations (max), that share data and control information under user control.
- 14K words of battery-backed RAM for ladder diagram programming and data files.
- 1K bytes of VMEbus short global memory and 4K bytes of VMEbus short, standard, or user-defined global memory for communication with the host CPU.
- Commands for controlling data transfers across the VMEbus backplane via VMEbus global memory to or from the host CPU using a software driver program that you write.
- Select the operating mode manually with a keyswitch on the front panel or remotely from the host CPU or a programming terminal.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

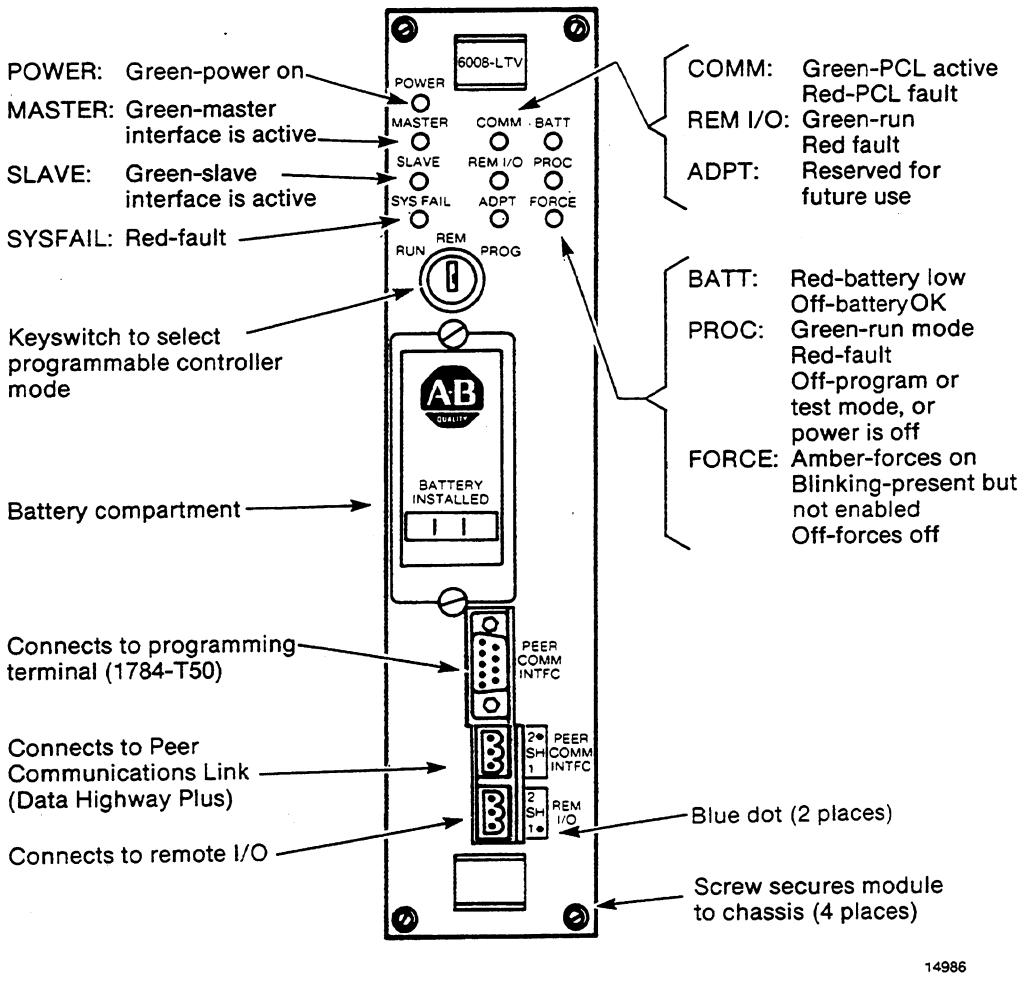
Mechanical Package

The LTV fits into a VMEbus subrack in the space of three double-height Eurocards. The module consists of three boards. Two outer boards fit into the VMEbus cardguides. These two boards use only the J1 (top) connector for electrical connection to the VMEbus backplane. The center board is mounted to one of the outer boards. All three boards are connected electrically by a mini-backplane, independent of the VMEbus backplane.

CAUTION: Do not disassemble the unit. Return it to the factory for service, if required.

The module has a single faceplate.

6008-LTV Faceplate



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Electrostatic Damage

Electrostatic discharge can damage integrated circuits and semiconductors in this module. Damage can be immediate or cumulative. The resulting malfunction can be immediate or not realized until some future time when the damaged portion of the device is required to perform. Electrostatic discharge can damage these devices when your fingers come in direct contact, by arcing from your fingers, or by arcing from adjacent conductors. We recommend that you take the following precautions to avoid electrostatic damage:

- Touch a grounded object to rid yourself of electrostatic charge before handling he module.
- Handle the module carefully by its edges or front panel.
- Avoid touching the solder side of the circuit board, or components on the component side.
- When not in use, keep the module in its static-shield bag.

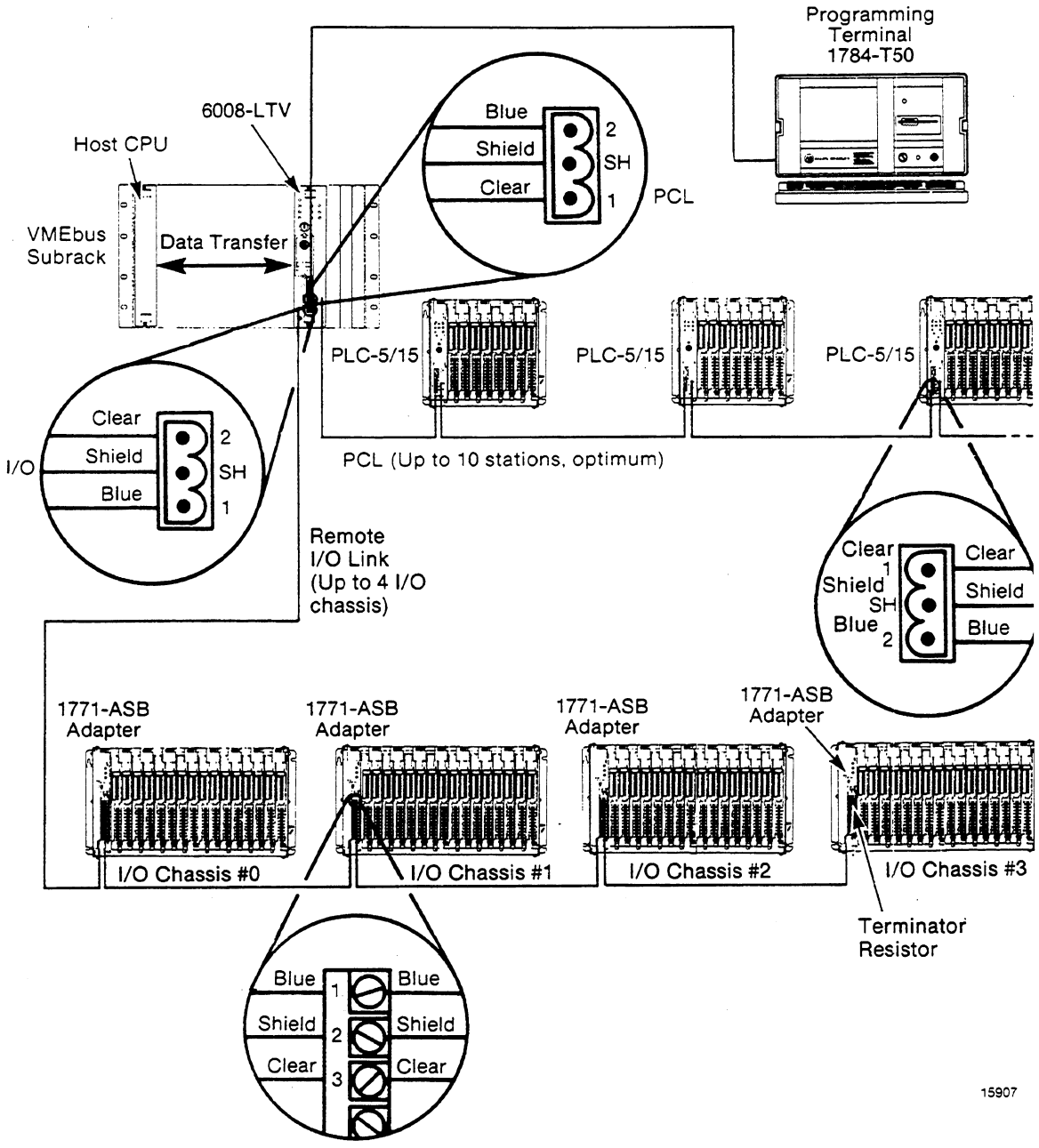
CAUTION: Electrostatic discharge can degrade performance or damage the module. Observe the precautions stated above.

System Description

The LTV is a PLC-5/15 programmable controller with a VMEbus interface. As a programmable controller it monitors inputs and controls outputs over its remote I/O link to four (max) I/O chassis distributed through the control system. You must write a ladder logic program to do this. You program the LTV with a programming terminal connected to it peer communications link (PCL). The LTV can communicate with other programmable controllers over the PCL. The LTV can operate independent of VMEbus communication. The LTV's VMEbus interface lets it respond to commands from the host CPU or other VMEbus masters to transfer data and/or memory across the VMEbus backplane. You must write a software driver program to do this.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

LTV Communications



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Operation as a Programmable Controller

The LTV is equivalent to an Allen-Bradley PLC-5/15 (Series B) programmable controller with a VMEbus (Revision C.1) interface. As a programmable controller it monitors inputs and controls outputs via its remote I/O link, performs math and logic operations, performs process control (PID), and many other functions including communication with other Allen-Bradley PLC-5 family programmable controllers on the Peer Communications Link.

You create a ladder diagram program to control your application. The PLC-5 Family Programmable Controller User's Manual, publication 1785-6.8.2 describes the instruction set. You enter the ladder diagram program into memory with an Allen-Bradley 1784-T50 Industrial Terminal or an IBM PC/AT computer. We provide 6200 Series software and manuals for use with either programming terminal.

I/O Subsystem

The internal scanner of the LTV can scan up to four remote I/O chassis containing 8-point, 16-point, or 32-point discrete I/O modules. The maximum number of discrete I/O points (terminals) is 512, or 1024 by configuring your I/O in a complementary mode.

Typical update time to transfer discrete I/O data is 6ms per chassis at 57.6 Kbaud. While scanning remote I/O, the LTV can simultaneously transfer data files with the host CPU, with another PLC-5 family programmable controller on its peer communications link, or upload/download program or data files with a programming terminal.

Block Transfer

The internal scanner of the LTV transfers blocks of data (up to 64-word data files) to and from intelligent I/O modules. The scanner queues block transfer requests for each of the four chassis independently. A block transfer request must wait for the completion of all block transfer requests ahead of it for that chassis.

The transfer time between scanner and I/O modules depends on whether the LTV is reading or writing data, the number of I/O chassis, and the number of queued block transfer requests. If interested, refer to Processor Communication in the PLC-5 Family Processor manual for details.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Peer Communications Link

The LTV communicates with PLC-5 family programmable controllers and 1784-T50 Industrial Terminals via the Peer Communications Link (PCL) also referred to as Data Highway Plus. You program communication between the LTV and PLC-5 family programmable controllers using the message instruction in your ladder diagram program. Communication between an Industrial Terminal and an LTV is driven by the terminal's software program. The PCL (Data Highway Plus) can link up to 10 stations (optimum), 64 stations (max) via a twin-axial cable up to 10,000 cable feet.

Using the PCL, the LTV can

- transfer data files with up to 63 other stations
- upload/download ladder diagram programs and data files with a 1784-T50 Industrial Terminal
- simultaneously scan remote I/O while communicating on the PCL

When linking 10 stations or less, the PCL can be a faster mode of communication than Data Highway I.

The front panel of the LTV contains an interface port and screw terminals for connections to the PCL. The screw terminals and interface port are electrically connected together inside so you can connect the 1784-T50 terminal to the PCL via the LTV.

Parity Checking

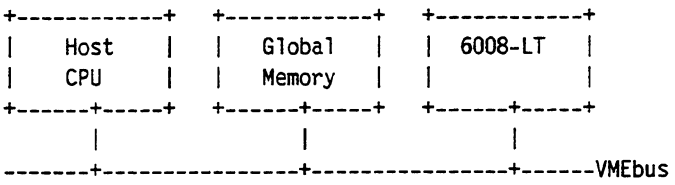
The LTV checks parity at power-up and every time it reads user RAM. If it detects a parity error, it turns on its processor fault LED and changes its operating mode from run or remote run to program mode. If it detects a parity error at power-up, the LTV also clears its RAM before going to program mode.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

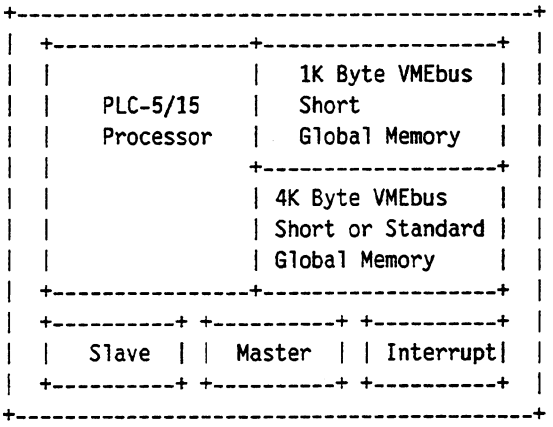
VMEbus/6008-LTV Interface

The VMEbus interface of the LTV consists of a slave, master, and interrupter. The LTV generally functions as a slave, but can initiate data-transfer commands when instructed by its ladder diagram logic or by the host CPU. The host CPU or VMEbus masters and the LTV pass commands and data through VMEbus global memory. You may provide a VMEbus global memory module for this purpose or use the 5K byte global memory inside the LTV. (If you do not require a VME interface, disregard this manual.)

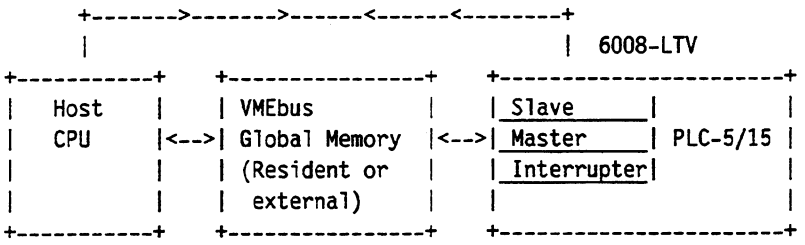
VMEbus Concept



6008-LTV Block Diagram



VMEbus Interface



Operation of the LTV defaults to slave operation. If you select internal global memory, the LTV never uses its master interface. If you select external global memory, the LTV uses its master interface to respond to commands from the host CPU. The LTV can initiate copy-to-VME and copy-from-VME commands from its ladder logic.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

VMEbus Compliance Capabilities

The VMEbus Specification, Revision C.1 describes various VMEbus capabilities of equipment operating in the VMEbus. The VMEbus capabilities applicable to the LTV are listed below followed by VMEbus notation definitions.

VMEbus Capabilities of the 6008-LTV

```
A16/D08(EO)      MASTER and BLT MASTER
A16/D08(O)       MASTER and BLT MASTER
A16/D16          MASTER and BLT MASTER
A24/D08(EO)      MASTER and BLT MASTER
A24/D08(O)       MASTER and BLT MASTER
A24/D16          MASTER and BLT MASTER
A16/ADO          SLAVE
A24/ADO          SLAVE
A16/D08(EO)      SLAVE, RMW SLAVE, and BLT SLAVE
A16/D16          SLAVE, RMW SLAVE, and BLT SLAVE
A24/D08(EO)      SLAVE, RMW SLAVE, and BLT SLAVE
A24/D16          SLAVE, RMW SLAVE, and BLT SLAVE
7-level ROAK interrupter with D08 or D16 Status/ID
Bus request levels 0-3, Bus release type RWD, ROR and ROC*

* You select RWD, ROR, and ROC by setting a software bit.
Refer to chapter 5, Bit/Byte Descriptions of Slave O
Memory, and chapter 6 Copy Command Bit/Word Descriptions.
```

Notation Definitions

A **Master** can generate simple read or write cycles.
A **BLT MASTER** can generate block transfer read or write cycles.

An **ADO SLAVE** can tolerate address-only cycles.
A **SLAVE** can respond to simple read or write cycles.
A **BLT SLAVE** can respond to block transfer cycles.

A **ROAK INTERRUPTER** can be configured via software to interrupt at any level listed (1, 2, 3, 4, 5, ,6, 7). The INTERRUPTER will release the interrupt line as a result of a STATUS/ID read cycle. Software programming determines if a D08 or D16 Status/ID is returned.

A **BUS REQUESTOR** can be configured via hardware to request control of the VMEbus on any level listed (1, 2, etc). Configure it by software to release the bus with the following:

RWD = Release when Done when cycle is completed

ROR = Release on Request from another requestor

ROC = Release on (bus) Clear

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Chapter 3 Getting Started with Copy Commands

Introduction

This tutorial chapter shows you how to copy data from an LTV data file into a global memory location (Copy-to-VME command), and copy data in the reverse direction (Copy-from-VME command). We introduce definitions and concepts as needed. We want you to learn how the LTV performs these commands by demonstrating their operation.

You must have Series 6200 Software (Rev 2.1 or later).

Definitions

We present the following definitions that we use in this tutorial and later in the manual.

Host CPU The host CPU is the primary device that communicates with the LTV. The host CPU's driver program controls this communication. The host CPU can be integral to the system controller or a separate VMEbus module placed adjacent to the system controller in the VMEbus subrack.

LTV The 6008-LTV module that contains a VMEbus interface and a PLC-5/15 programmable controller.

Copy Commands Copy-to-VME and copy-from-VME commands data between host CPU and LTV (up to 500 words) in the direction indicated. Data is transferred every LTV program scan (typically every 10-100ms). The LTV or host CPU can start and stop these commands.

Selective Commands These commands let you transfer data and/or programs between host CPU and LTV, one command at a time. The transfer continues until all data reaches its destination. Completion is detected by polling or by interrupt.

Command Block The command block stores information that defines the command, such as source and destination addresses, address modifiers, and the number of words transferred. The LTV or host CPU writes the command block into global memory when it initiates a command. It also writes the location of the command block (a "pointer") into a specified address of Slave 0 memory to point to its location in global memory.

Global Memory All data and commands must pass through global memory when transferred between LTV and host CPU. You select the base address, either internal or external to the LTV. Use internal (5K byte) global memory to reduce traffic on the VMEbus. Use an external global memory module if your application requires more memory.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Slave 0 Global Memory This is a 1K byte memory internal to the LTV. You select its base address (short, only) and address modifier(s) with switches on the module. Up to 942 bytes are available for data transfers. The balance of 82 (52H) bytes is reserved for implementing commands and LTV identification.

Slave 1 Global Memory This 4K byte internal memory is separate from Slave 0 memory. You select Slave 1 base address (short, standard, or user-defined) and address modifier(s) by loading this data into specified storage locations in Slave 0 memory with your Host CPU's driver program. You initialize Slave 1 memory in this manner at power up. All of Slave 1 is available for data transfers.

LTV Data Files Each data file is defined by a starting address and number of elements (length). The LTV has up to 1000 data files for storing data associated with instructions in your ladder diagram program for controlling a machine or process. The first nine files are pre-assigned as follows:

File	File Type	File	File Type
0	output image (O)[1]	5	counter (C)[3]
1	input image (I)[1]	6	control (R)[3]
2	status (S)[1]	7	integer (N)[1]
3	bit (B)[1]	8	floating point (F)[2]
4	timer (T)[3]	9-999	B, T, C, R, N, and F

( ) = file's abbreviation [ ] = words per element

Each file can store up to 1000 elements of data.

LTV Data File Address You assign a file number (0-999) and type designator (O, I, S, B, etc. see above) to each file for its address. For example, N15 is integer file 15. You can also subdivide a file into data blocks by adding the starting element number to the address. For example, N15:24 or N15:0. N15 and N15:0 are the same address.

VMEbus File Address For this tutorial, you must use 16-bit short addresses because you will be using the LTV's Slave 0 memory which is a short-address memory.

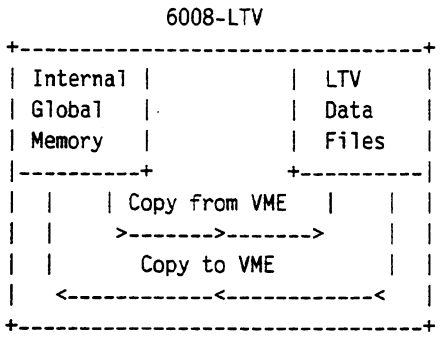
PCL (LTV) Station Number You must assign a station number to the LTV so your host CPU can differentiate it from other LTV modules, and so the programming terminal can differentiate it from other stations which can be connected serially to the terminal.

Programming Terminal When connected to the LTV over the PCL, you can enter ladder diagram programs, create data files, and perform other programming tasks. We give you the electrical connections, keystroke definitions, and display screens required in this tutorial.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Tutorial Overview

Copy commands transfer data between VMEbus global memory and programmable controller data files in the LTV. Global memory can be internal or external to the LTV. For this tutorial we use global memory inside the LTV, so data transfers will be internal to the LTV. It is the task of the host CPU'S driver program to fetch data from or place data into global memory regardless of the location of global memory. However for this tutorial, the transfer of data is only between VMEbus global memory and LTV data files. It is independent of the host CPU, so you will not need one. You will perform this tutorial with the LTV module and a programming terminal.



In the following tutorial, you will generate copy commands and observe their operation. You will create an LTV data file, copy its data into internal global memory (Copy-to-VME command) where you cannot monitor it with a programming terminal, then copy it back to a different LTV destination address (Copy-from-VME command) and observe identical data.

The tutorial consists of the following parts.

- Set LTV switches to determine the PCL (LTV) station number, base address and address modifier(s) of Slave 0 global memory.
- Connect the equipment.
- Define the transfer.
- Enter all required data into LTV memory with the programming terminal's display screens. You will create the VME Status File, the LTV Data File, and enter parameters.
- Initiate the commands and observe the results.

When necessary, we divide each part into steps.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Electrostatic Damage

Electrostatic discharge can damage integrated circuits and semiconductors in this module. Damage can be immediate or cumulative. The resulting malfunction can be immediate or not realized until some future time when the damaged portion of the device is required to perform. Electrostatic discharge can damage these devices when your fingers come in direct contact, by arcing from your fingers, or by arcing from adjacent conductors. We recommend that you take the following precautions to avoid electrostatic damage:

- Touch a grounded object to rid yourself of electrostatic charge before handling he module.
- Handle the module carefully by its edges or front panel.
- Avoid touching the solder side of the circuit board, or components on the component side.
- When not in use, keep the module in its static-shield bag.

CAUTION: Electrostatic discharge can degrade performance or damage the module. Observe the precautions stated above.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Setting Switches

Avoid electrostatic damage when handling the module. See **Electrostatic Damage** on the previous page.

You must set the switches of two switch assemblies:

- SW1 is the only 8-position switch assembly on top of the module. Use it for setting the PCL (LTV) station number.
- SW4 is one of two 8-position switch assemblies on the bottom of the module. Use it for setting the base address and address modifiers (AM)

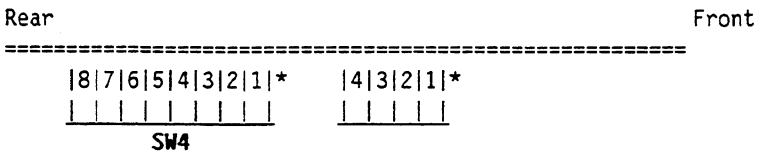
SW1 Set switches 1-6 of this 8-position switch on the top of the module to PCL station number 5 (you could use 0-64) by setting the binary code (low = true) as follows:

```
1   2   3   4   5   6 | 7   8
off - off  -   -   - | off off
where "-" must be "on" and
switches 7 and 8 must be off
```

IMPORTANT: You must select a station number so the programming terminal knows what PCL station it is talking to.

SW4 Locate this 8-position switch assembly on the bottom of the module, closest to the rear.

Bottom View of LTV



For this tutorial, use a Slave 0 base address of 4000 and address modifier 2D. Set SW4 switches as follows:

```
  Base Address   | AM
8  7  6  5  4  3 | 2  1
                  |
- off -  -  -  - |off -

where "-" must be "on"
```

Switch settings are also covered in chapter 10.

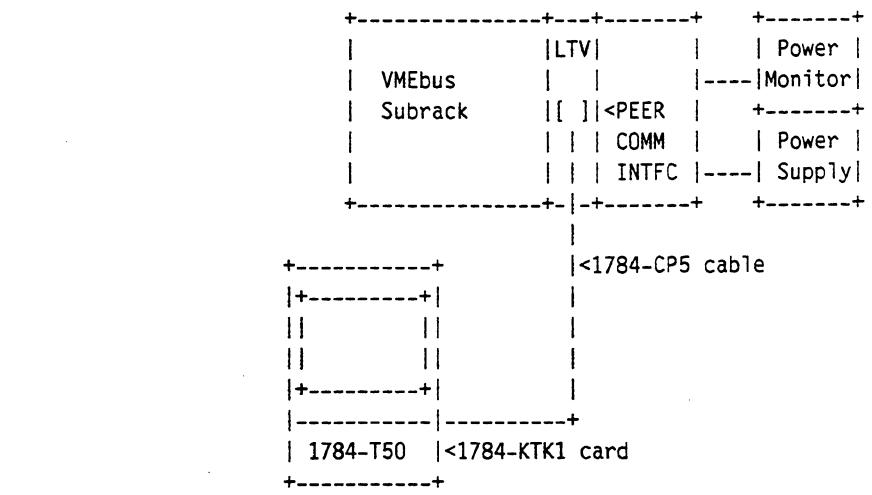
Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Connecting the Equipment

You need the following equipment for this tutorial:

6008-LTV
VMEbus subrack
VMEbus power supply
Power monitor module
Programming terminal (cat. no. 1784-T50)
Cable (1784-CP5)
6200 Series Software (Revision 2.1 or later)

Connect them as follows:



1) Slide the LTV into the VMEbus subrack. (Switches must be set.)

2) Connect the +5V DC power supply and power monitor to the VMEbus subrack.

3) Connect the "processor" end of the 1784-CP5 cable into the PEER COMM INTFC connector on the front of the LTV, and the "terminal" end of the cable into the connector in slot 8 (1784-KTK1 card) on the right side of the programming terminal.

4) Connect power to the programming terminal and power supply.

5) Place the LTV in program mode with the keyswitch on the front panel.

You do not need a host CPU for this tutorial.

IMPORTANT: You need a power monitor module to reset the VMEbus subrack at power up, and to provide an AC Fail signal to save LTV memory in case of a power loss. The AC Fail signal must meet the requirements of The VMEbus Specification Rev C.1, sections 5.1, 6.21, and 6.22.

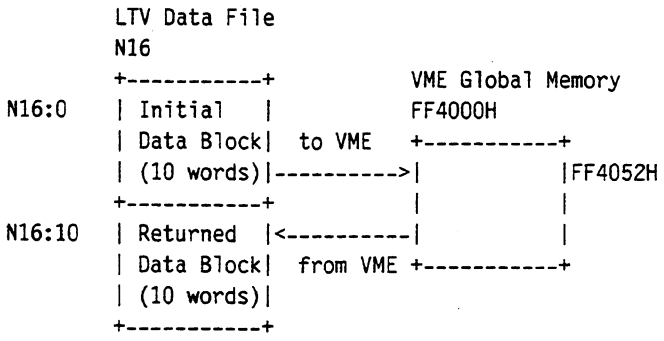
Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Defining the Transfer

In this tutorial, you will transfer a 10-word block of data from an LTV data file address to VMEbus global memory located inside the LTV using a Copy to VME command, then copy it back to a different LTV data file address using the Copy from VME command.

You will use two data blocks in integer file N16: the initial data block, words 0-9 of N16, starts at address N16:0; the returned data block, words 10-19 of N16 starts at address N16:10. You will enter and/or change data in words 0-9 and see it returned in words 10-19.

We are using two data blocks from the same file N16, so you can observe the initial data block and returned data block on the same screen. You could do this tutorial using two separate files, such as N16 and N17, but you would have to observe N16 on one screen and N17 on another.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Entering Data

You enter data with your programming terminal by advancing through a series of screens. You select the next screen by pressing one of several soft function keys described at the bottom of each screen. Soft function keys are [F1] through [F10]. Their function may change from screen to screen.

Entering the PCL Station Number

The first part of our journey is to tell the programming terminal the PCL station number that you assigned to the LTV so the terminal can talk to it.

1) Switch the LTV to program mode using the keyswitch.

2) Turn on power to the VMEbus subrack.

3) Turn on the programming terminal using the switch on the right-hand side in front of the power cable.

4) Load Series 6200 Software (Revision 2.1) into your programming terminal. Refer to publication 6200-6.5.5 for the procedure. If you haven't done this yet, you will have to set this tutorial aside until it is done.

5) From the terminal's main menu, select option 3

   Press [3] PLC-5 Industrial Terminal Software

6) Press [F5] WHO - a new screen appears

7) Press [F5] WHO ACTIVE - a new screen appears

   The screen should show a list of station numbers with a blinking [5VME] marker next to the station number you assigned the LTV by setting switch assembly SW1.

   Press [F5] SELECT STATION

   This informs the terminal that the displayed station number is the one you want to talk to.

8) Press [ESC][ESC] to exit.

   The screen returns to the menu function key options [F1] - [F8] and [F10].

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Creating the VME Status File

Next you create a VME status file by entering its number in the PC Status File Screen.

1) Press [F1] ONLINE PRG/DOC on the menu options screen.

2) You must give the processor a name.

   Press [F1] PROCESSOR FUNCTIONS

   Press [F3] CHANGING PROCESSOR NAME

   Type [**L**][**T**][**V**][**enter**]

3) Create a program file

   Press [F6] CREATE FILE

   Type [**2**][**enter**]

4) Press [F8] MONITOR FILE - a new screen appears

5) Press [F8] DATA MONITOR - a new screen appears

6) Type [s][**enter**] at the cursor

   This displays the PC status file.

7) Enter the VME status file number by cursoring up (one up-arrow stroke) to the "VME status file:" line.

   Type [**9**][**enter**] (You could enter any number 9-999.)

Entering Copy Command Parameters

Next you display the VME Copy Configuration Screen so you can enter copy command parameters into the VME Status File (data file 9) that you just created.

1) Press [F4] EXTENDED STATUS from the previous screen.

   The VME Copy Configuration Screen is displayed.

2) Enter the following values in the Copy from VME and Copy to VME sections of the screen. Use arrow keys as required.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

```
    +-------------------------------------------------------------------------+
 0  |                        VME COPY CONFIGURATION                           |  0
 1  |                                                                         |  1
 2  |        COPY FROM VME                        COPY TO VME                 |  2
 3  |                                                                         |  3
 4  |Enable (off=0,on=1)              0   Enable (off=0,on=1)              0 |  4
 5  |X-fer pt (before=0,after=1)      0   X-fer pt (before=0,after=1)      0 |  5
 6  |Scan mode (discontin=0,contin=1) 0   Scan mode (discontin=0,contin=1) 0 |  6
 7  |Error                          00H   Error                          00H|  7
 8  |Source address modifier        00H   Dest address modifier          00H|  8
 9  |VME address                 000000H  VME address                 000000H|  9
10  |Length (words)                   0   Length (words)                   0 | 10
11  |Data size(D16=0,D08(E0)=1,D08(0)=2 0  Data size(D16=0,D08(E0)=1,D08(0)=2 0| 11
12  |Destination DT file              0   Source DT file                    0 | 12
13  |Element                          0   Element                         0 | 13
14  |Interrupt level (disable=0)      0   Interrupt level                  0 | 14
15  |Status/ID                     0000H  Status/ID                     0000H| 15
16  |RWD/ROR (RWD=0, ROR=1)           0   RWD/ROR (RWD=0, ROR=1)           0 | 16
17  |ROC (ignore=0, ROC=1)            0   ROC (ignore=0, ROC=1)           0 | 17
18  |                                                                         | 18
19  |Press a key or lenter value.                                             | 19
20  |N55:23 = [                                                               | 20
21  |remote program  no forces     decimal data  decimal addr  PLC-5/VME Addr 76| 21
22  |         PROC             VME    SPECIFY                                 | 22
23  |        STATUS          STATUS   ADDRESS                                 | 23
24  |          F2              F4       F5                                    | 24
    +-------------------------------------------------------------------------+
```

IMPORTANT: The VME Copy Configuration screen contains these errors in Revision 2.1 of 6200 Series Software. Correct the display as follows:

Line	Caption	Incorrect Revision 2.1		Correct Revision 2.2
5	X-fer pt	before = 0	--->	before = 1
		after = 1	--->	after = 0
6	Scan mode	discontin = 0	--->	synch = 1
		contin = 1	--->	contin = 0
16,17	absent	absent		as shown above

COPY FROM VME

Source Address Modifier - Type [2][D][enter]
(because you selected the 2D address modifier for Slave 0 memory by setting switches)

VME address - Type [F][F][4][0][5][2][enter]
(This 6-digit address is the address in VMEbus global memory from which this command transfers data. You must add an offset of 52H bytes or more to your base address of 4000H for Slave 0 which you selected by setting switches. The offset represents a reserved area of Slave 0 memory.)

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Length - Type [1][0][enter]
(Transfer 10 words.)

Destination DT file - Type [1][6][enter]

Element - Type [1][0][enter]
(Destination and element define the LTV data block address N16:10 to which this command transfers data.)

COPY TO VME

Source DT file - Type [1][6][enter]

Element - Type [0][enter]
(Source and element define the LTV data block address N16:0 from which this command transfers data.)

Length - Type [1][0][enter]
(Same as for the Copy from VME command)

Destination address modifier - Type [2][D][enter]
(Same as for the Copy to PCL command)

VME address - Type [F][F][4][0][5][2][enter]
(This is the address in VMEbus global memory to which this command transfers data. Otherwise, it is the same as for the Copy to PLC command.)

3) Press [ESC] - the screen displays the PC Status File

4) Press [ESC] - the screen displays the menu function key options

Creating the LTV Data File

Now you need to create the integer file N16 that you specified in the copy command parameters.

1) Press [F7] GENERAL UTILITY

2) Press [F1] MEMORY MAP
(Screen displays data table map)

3) Press [F6] CREATE DT FILE

4) At the cursor, type [n][1][6][:][1][9][enter]
(The :19 specifies elements 0-19 for this file.)

Prompt: Create data table address?

Press [F1] YES

The screen displays "N16:19" with a size of "20".

The LAST ADDRESS is the address of the last word in the specified file.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

5) Press [ESC] to exit.

Entering Data in the LTV Data File

Next, you need to load data into the LTV data file N16 words 0-9 (10-word data block starting at N16:0) because you will transfer this data block to VMEbus global memory with a Copy to VME command, and then copy it back into words 10-19 (10-word data block starting at N16:10) with a Copy from VME command.

1) Press [F8] DATA MONITOR

2) At the cursor, type [**n**][**1**][**6**][**enter**]

3) Select HEX DATA by pressing [F1][F4]
(File words are displayed across the top of screen.)

```
address  0    1    2    3    4    5    6    7    8    9
N16:0    0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
N16:10   0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
```

Using the left/right arrow keys, Enter values into words 0-9 as follows:

```
address  0    1    2    3    4    5    6    7    8    9
N16:0    0000 1111 2222 3333 4444 5555 6666 7777 8888 9999
N16:10   0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
```

Do not load data into words 10-19 because this is the location that will receive the data block returned from VMEbus global memory location starting at 4052H.

4) Press [ESC] to exit.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Initiating Copy Commands and Observing the Results

Initiate copy commands by setting the enable bits and visually comparing the outgoing data that you loaded into N16:0 words 0-9 with the data block returned to N16:10 words 10-19.

1) Press [F8] DATA MONITOR - a new screen appears

2) At the cursor, type [**s**][**enter**]

   This displays the PC status file.

3) Advance to the VME Copy Configuration Screen by cursoring up to "VME status file:" line and pressing [F4] EXTENDED STATUS.
   (The VME Copy Configuration Screen is displayed.)

4) With the cursor on the "Enable (off=0,on=1)" line, Type [**1**][**enter**] in the number field of both commands.

5) Switch the LTV from program to run mode using the keyswitch on the LTV's front panel. Both commands should now be transferring data.

6) Advance to the file display of LTV data file N16.

   Press [F5] SPECIFY ADDRESS

   Type [**n**][**1**][**6**][**enter**]

   (Words 0-19 are displayed at the top of the screen.)

7) Select HEX DATA by pressing [F1][F4].
   (The screen displays data in Hex.)

```
address  0    1    2    3    4    5    6    7    8    9
N16:0    0000 1111 2222 3333 4444 5555 6666 7777 8888 9999
N16:10   0000 1111 2222 3333 4444 5555 6666 7777 8888 9999
```

You should see the data that started in LTV data block N16:0 (first row), transferred to the VMEbus global memory with the Copy to VME command, and then transferred back to data block N16:10 (second row) with a Copy from VME command.

8) Now have fun with this tutorial by changing values in the original data block (first row) and seeing the image returned (second row).

   Move the cursor to the first row using arrow keys. Type your own characters. Press [**enter**].

   Your characters are returned in the second row.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Conclusion

You should now be familiar with copy commands, LTV data file structure and addressing, and using the programming terminal. You may want to read more about programmable controller memory organization and addressing in the PLC-5 Family Processor Manual, publication 1785-6.8.2.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Introduction

In this chapter we show you how to enable and disable copy commands, and change command parameters from the

- LTV's ladder diagram program
- host CPU's driver program

Initiating Copy Commands from the LTV's Ladder Program

You enable or disable copy commands from the LTV's ladder diagram program by setting or resetting a bit in the VME Status File for each command. You did this in chapter 3's tutorial with the programming terminal. In this section, we show you how to do it with the LTV's ladder diagram program.

We assume that you have created the necessary files for transferring data, and loaded VME copy command parameters with your programming terminal. If not, refer to chapter 3 for the procedure to do this.

Enable Bits

Enable bits of the copy command are found in control words in the VME Status File:

word 6, bit 15 for Copy from VME
word 15, bit 15 for Copy to VME

setting this bit (to 1) starts the command
resetting this bit (to 0) stops the command

You write ladder logic to enable or disable these bits.

Ladder Logic

The LTV's programmable controller uses ladder logic to monitor input and output conditions (left side of the rung) and control outputs (right side of the rung). An "examine ON" instruction has the symbol -] [- while an "energize output" instruction has the symbol -( )-. Together they comprise a logical rung:

```
examine ON                                  energize output
--] [-------------------------------------------( )--
```

When the "examine ON" instruction sees an ON or closed condition, its logic goes true and the output instruction is enabled. When the "examine ON" instruction sees an OFF or open condition, its logic goes false and the output instruction is disabled.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

You can latch ON an output by using a latched output instruction -(L)-. It requires an unlatched output instruction -(U)- having the same address to turn it off. Rungs with latched and unlatched output instructions must be used in pairs.

Assign an address to each instruction so it knows where to store its status in LTV I/O image files. Refer to PLC-5 Family Processor Manual, publication 1785-6.8.2.

Address of "Examine ON" Instruction

The address of an "examine ON" instruction contains numbers representing the rack, I/O group, module slot, and terminal. The numbers correspond to the physical address of the input module in an I/O chassis, and have an equivalent address in the LTV's input image file (Data File 1). The address **I:10/08** represents input image word 10 bit 08; and also represents I/O rack 1, I/O group 0, input terminal 08.

Address of Output Bit that Enables the Copy Command

Output latch and unlatch instructions turn on and off the copy command's enable bit. The copy command's enable bit is located in the control word in the VME Status File (N9 from the previous tutorial). The address would be:

N9:6/15 for Copy from VME
N9:15/15 for Copy to VME

Therefore, rungs to start and stop copy commands in this example could be

```
Start Copy from VME
  I:10                                          N9:6
--] [-----------------------------------------(L)--
    08                                            15

Stop Copy from VME
  I:10                                          N9:6
--] [-----------------------------------------(U)--
    09                                            15

Start Copy to VME
  I:10                                         N9:15
--] [-----------------------------------------(L)--
    10                                            15
Stop Copy to VME
  I:10                                         N9:15
--] [-----------------------------------------(U)--
    11                                            15
```

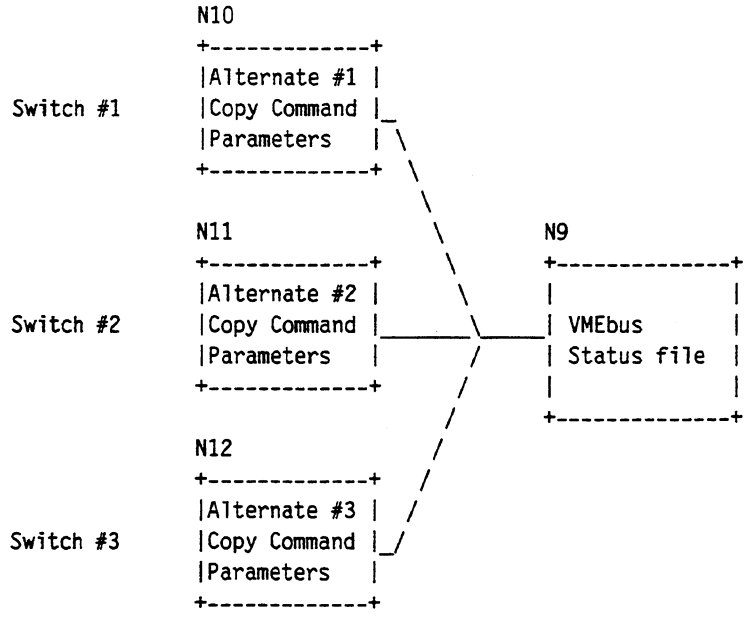
IMPORTANT: As long as the enable bit remains set, the command transfers data every LTV program scan.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Changing Command Parameters from the LTV's Ladder Program

We recommend that you change command parameters by copying a new set of copy command parameters from a storage file into the VMEbus status file. You can do this with the File Copy (COP) instruction.

Suppose you store three different sets of copy command parameters in files N10, N11, N12, with each one enabled by a different switch input. For example, when you turn on switch 2, the LTV moves copy command parameters from file N11 into the VMEbus Status File. We chose N9 as the VMEbus Status File in this example.



You would store alternate sets of copy command parameters in data files with your programming terminal.

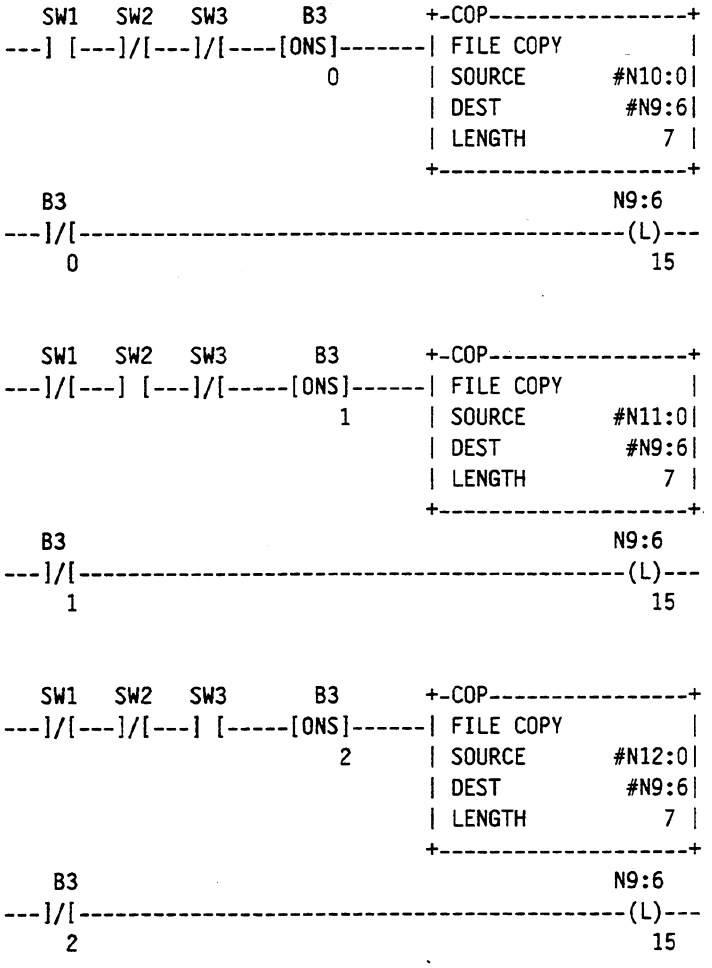
You can program the transfer of alternate copy command parameters into the VMEbus Status File with a File Copy (COP) instruction and a One-shot (ONS) for each set of values. The COP instruction writes its data over the data already stored in the VMEbus Status File. The new data should reset the copy command enable bit. The ONS instruction ensures that the enable bit is reset for one program scan (see ladder logic, below).

IMPORTANT: To change copy commands, you must transfer new copy command parameters into the VMEbus Status File and toggle the enable bit. The VMEbus Status File accepts new parameters at any time but the LTV cannot act on them until your program toggles the enable bit (from set to reset to set). The enable bit must remain reset for at least one LTV program scan.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

IMPORTANT: Each time you change copy command parameters, the LTV's ladder program scan is delayed approximately 20ms for one scan. Then it resumes normal scan time.

NOTE: The following ladder logic changes the parameters of a Copy from VME command for three different sets of parameters stored in N10:0, N11:0, and N12:0. You could use it to change values of a Copy to VME command by copying values into VME Status File starting at word 15 (change #N9:6 to #N9:15 and N9:6 to N9:15). We used a file length of 7 in this example to omit the interrupt. For the VME Status File, you could use any integer file (9-999). Refer to **Memory Map of VME Status File** in chapter 9.



We recommend that you study the PLC-5 Family Processor Manual, publication 1785-6.8.2, to become familiar with data file organization, program file organization, and the instruction set used to monitor, control, and manipulate data.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Initiating and Changing Copy Commands from the Host CPU's Driver Program

You enable, disable, or change copy comm
host CPU's driver program by sending com
the LTV that set or reset the enable bit a
command values. The command block inc
bits, source and destination addresses, ad
modifiers, and file length. The enable bit
word 7 in the copy command block. Refer t
Block in chapter 6 for details.

We assume that you have created the nece
including the VME Status File for transfe
that you have loaded VME copy command
LTV with your programming terminal. If
chapter 3 for the procedure to do this.

Driver Program

To initiate a copy command, your driver p
should perform the following functions:

1) Access VMEbus global memory:
Place the command block at a valid a
command block should contain the re
including a reset enable bit.

2) Access the LTV's Slave 0 memory:
Test the LTV's semaphore bit (byte 4
the starting address). When it is foun
(LTV ready for this command), set thi
the command block's VME address in
Then place the command block's VME
in byte 41H. Follow this order.

3) Re-send the command block with the e
and monitor the response word (comm
1) to verify completion of the comman

To change a copy command, your driver pr

1) Issue a copy command block as descrit
new values and the copy command en

2) Upon confirmation that the LTV recei
command with new values and stoppe
issue another command block with the

We provide you with an example driver pr
initiate a Copy to VME command. The com
data between a source LTV data file and a
VME file in VMEbus global memory every

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

We assume that you can modify this program to change copy command values, or to initiate or change a Copy from VME command.

Refer to **Copy Commands** in chapter 6 for additional information on the copy command block and copy command sequence.

```
/*****************************************************************************
*
* Synopsis
*     This program will initialize the 6008-LTV to do continuous copies
*     from the LTV to VME global memory. The parameters in the command
*     block are set once in the main program. To change the
*     characteristics of the copy process, the values in the command block
*     would need to be changed and resent to the LTV. The copy enable bit
*     would also have to be toggled in order for the LTV to recognize the new
*     parameters.
*
*****************************************************************************/

/* Define variables and structure. */

typedef unsigned char       UTINY;    /* 8 bits no sign   */
typedef char                BOOL;     /* 8 bits with sign  */
typedef unsigned int        USHORT;   /* 16 bits no sign   */
typedef int                 SSHORT;   /* 16 bits with sign */


/* Timeouts   */

#define SEMA4_TIMEOUT       5000      /* Reset semaphore about 5 sec */
#define BLK_TIMEOUT         5000      /* Command block received */
#define RESPONSE_TIMEOUT    20000     /* Command response */
#define SEMA4_OFFSET        64         /* Semaphore offset from base addre

/* Structure for copy command blcok */

struct cmd_blk {
        USHORT com_word;              /* Command word */
        USHORT resp_word;             /* Response word */
        USHORT dat_int_lev;           /* Data received interrupt level */
        USHORT dat_vector;            /* Vector for data rec. interrupt */
        USHORT res1;
        USHORT res2;
        USHORT res3;
        UTINY  cont_bits;             /* Control bits in upper byte */
        UTINY  add_mod;               /* Address modifier for command blocl
        USHORT addr_msbyte;           /* High byte part of data address */
        USHORT addr_mid_low;          /* Mid and low bytes of data address *
        USHORT number;                /* Number of words to be transferred *
        USHORT file;                  /* LTV data file number */
        USHORT element;               /* Element number in LTV data file */
        USHORT done_int;              /* Copy done interrupt */
        USHORT done_vector;           /* Vector for copy done interrupt */
};
```

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

```
main()
{

struct cmd_blk *cm_bk;
USHORT word, i;
UTINY semaphore, bite;
UTINY *slv0_addr;
long temp_addr;

/* Set pointer equal to the starting address of Slave 0 */

        slv0_addr = 0xFF4000;

/* Initialize the copy command block */

        cm_bk                   = 0x50000;

        cm_bk->com_word     = 1;        /* Copy to VME code */
        cm_bk->resp_word    = 0;        /* Initialize to zero */
        cm_bk->dat_int_lev  = 0;        /* Interrupt not used, set to zero */
        cm_bk->dat_vector   = 0;        /* No vector needed */
        cm_bk->res1         = 0;        /* Reserved */
        cm_bk->res2         = 0;
        cm_bk->res3         = 0;
        cm_bk->cont_bits    = 0x48;     /* No Enable, Before I/O, Continuous, ROC */
        cm_bk->add_mod      = 0x39;     /* Standard address modifier, no-priv*/
        cm_bk->addr_msbyte  = 0x5;      /* High byte of data address */
        cm_bk->addr_mid_low = 0x0100;
        /* Mid and low bytes of data address */
        cm_bk->number       = 15;       /* Copy 15 words */
        cm_bk->file         = 11;       /* Copy from LTV data file 11 */
        cm_bk->element      = 20;       /* Starting from element 20 */
        cm_bk->done_int     = 0;        /* Interrupt not used, set to zero */
        cm_bk->done_vector  = 0;        /* No vector needed */


/* Before sending command block pointer to LTV, see if the LTV's semaphore is
   set.  If it is not, then set it to declare ownership of the command block
   pointer area.  If it never resets for a given amount of time, return an
   error code.  Use a true test-and-set if there are multiple masters in the VME sysem.*/

   for (i=SEMA4_TIMEOUT; i; i--){
        semaphore = *(slv0_addr + SEMA4_OFFSET); /* Read semaphore byte      */
        semaphore = semaphore & 0x80;            /* Check bit 7              */
        if (semaphore != 0x80)                   /* If not set, then it      */
                break;                           /* can be set.              */
   }

   if (!i){
        printf("\nSemaphore not reset, cannot send command block...");
        exit(1);
   }

     *(slv0_addr + SEMA4_OFFSET) = 0x80;         /* Set semaphore            */
```

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

```
/* Send the command block address into the Slave 0 pointer area */

    temp_addr = cm_bk;

    bite = ((temp_addr & 0x00FF0000) >> 16);          /* Addr bits 16-23      */
    *(slv0_addr + SEMA4_OFFSET + 3) = bite;

    bite = ((temp_addr & 0x0000FF00) >> 8);           /* Addr bits 8-15       */
    *(slv0_addr + SEMA4_OFFSET + 4) = bite;

    bite = (temp_addr & 0x000000FF);                  /* Addr bits 0-7        */
    *(slv0_addr + SEMA4_OFFSET + 5) = bite;

    *(slv0_addr + SEMA4_OFFSET + 1) = 0x39;           /* AM of command blk    */


/* Command block has been sent to LTV  now wait for the LTV's response by   */
/* monitoring the response word.                                             */

    for (i=RESPONSE_TIMEOUT; i; i--) {
        if (cm_bk->resp_word != 0x0000)
          if(cm_bk->resp_word == 0x00FF)
            break;
        else {
          printf ("\nError - Response word = %x", cm_bk->resp_word);
          exit(1);
        }
    }

    if (!i){
    printf ("\nError -  LTV never returned a response word.");
        exit(1);
    }

/*The command block has been sent and the LTV is now */
/*doing continuous copies if the enable bit is set and it is in run mode*/

}
```

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Overview of Global Memory

Global memory is used for data transfers with the host CPU. You select any one of three global memories.

Internal Slave 0
Internal Slave 1
External Memory Module

Slave 0

This required short-address global memory of 1K bytes has several uses:

- points to the command block in VMEbus global memory each time the host CPU initiates a command
- can serve as an optional internal VMEbus global memory for data transfers with the host CPU if you do not use Slave 1 or an external VMEbus global memory module
- initializes the LTV's internal Slave 1 global memory if you decide to use it
- stores LTV ID information, and VMEbus mastership options for accessing command blocks
- reports error codes

You select the starting address of Slave 0 memory on 400H increments within the 16-bit address range of 0000-FFFF with switches on the module. You also select one or two address modifiers with switches. The area of this memory for data transfers begins at byte 52H through 3FFH, offset from the starting (base) address. Refer to **Starting Address of Slave 0 Memory** later in this chapter.

The LTV defaults to slave operation using Slave 0 global memory.

Slave 1

Use this internal global memory for data transfer, if needed, as a larger alternative to Slave 0. This is a programmable RAM that you must initialize at power up and whenever power is cycled to the LTV.

You enable this memory by entering its starting address and address modifiers into memory access bytes of Slave 0. You may select a starting address that makes Slave 1 contiguous to Slave 0 memory so the pair appear as a single 5K byte internal global memory.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

You program this memory starting on 100H increments within the 16-bit short-address range of 0000-FFFF or the 24-bit standard-address range of 000000-FFFFFF. All short and standard VMEbus programmable address modifiers are available for either address range. Refer to **Configuring Slave 1 Memory** later in this chapter.

You can also configure Slave 1 memory as a user-defined memory with 16 or 24 bit addressing and up to eight user-defined address modifiers, with or without block transfer. Extended addressing is not available.

IMPORTANT: Do not set up Slave 1 to the same global memory address space as an external global memory module.

External Memory Module

You may substitute a VMEbus global memory module for the resident global memory if your application requires more global memory than Slave 1. If so, the LTV must master the VMEbus backplane to access the global memory module in response to commands from the host CPU. The LTV is a logical slave. It initiates VMEbus master cycles only when commanded by the host CPU. The host CPU still must use the command block pointer in Slave 0 memory to initiate commands to the LTV.

The manual for an external memory module tells you how to configure its starting address and address modifiers.

If commanded to read or write a block of data whose address does not match the Slave 0 or Slave 1 address range, then the LTV automatically accesses external global memory because it has VMEbus master capability.

IMPORTANT: If using an external global memory module, follow its guidelines for setting up its address space. Configure the LTV for VMEbus mastership options in byte 48H. Do NOT enter a Slave 1 address or address modifier into Slave 0 bytes 49-4FH to initialize Slave 1. (Refer to Bit/Byte Descriptions of Slave 0 Memory, below).

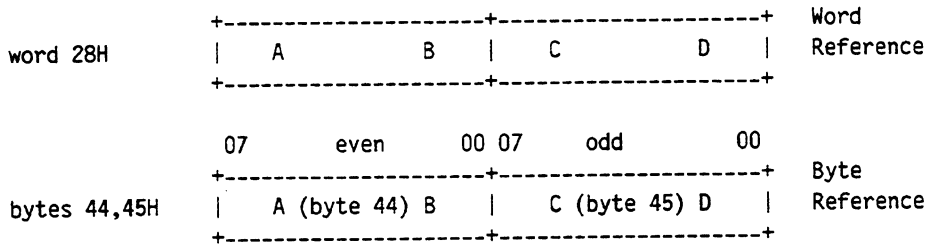
The remainder of this chapter describes Slave 0 memory in detail, and how to access Slave 1 internal global memory.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

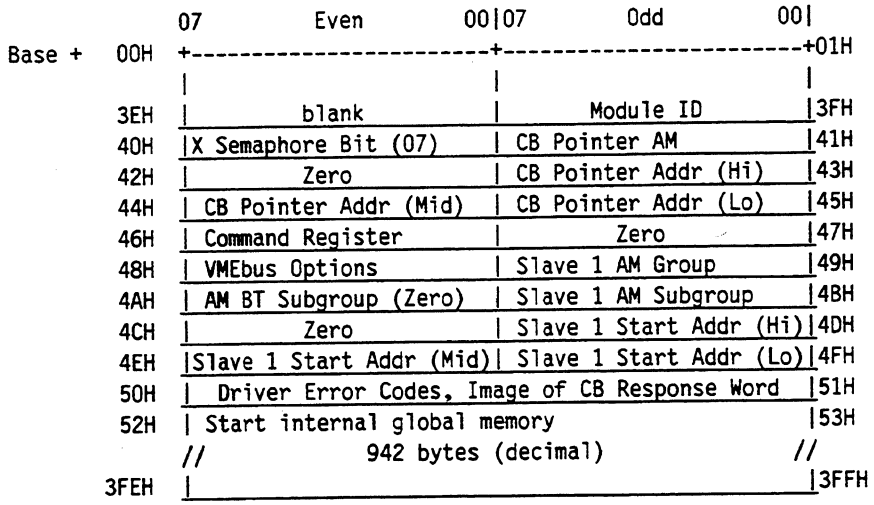
Slave 0 Memory

To compress data, Slave 0 parameters are packed in bytes. The LTV can access VMEbus memory in words, bytes, or only odd-numbered bytes. We present the memory organization in byte pairs as if each pair was a word. Even-numbered bytes starting with zero appear on the left, odd-numbered bytes on the right.

For example, a 4-digit number ABCDH stored in bytes 44H and 45H (word 28H) would have the MSD stored in the even-numbered (left) byte and the LSD stored in the odd-numbered (right) byte.



With this in mind, we now present the Slave 0 memory map. The ASCII hex addresses are offset from the base address that you select with switches on the module. You must use the 16-bit short address for the base address.



IMPORTANT: The host CPU and LTV use Slave 0 for:

Module identification	odd bytes 00-3FH
Semaphore, and pointing to command block	bytes 41-45H
Handshaking and mastership options	bytes 46-48H
Initializing Slave 1 memory	bytes 49-4FH
Reporting error codes	bytes 50-51H
Internal global memory for data transfer	bytes 52-3FFH

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Module ID Data Block

The first 32 odd-numbered bytes (0-3FH) of Slave 0 memory store module ID. These ASCII Hex addresses are offset from the Slave 0 base address that you select with switches on the module.

Offset to Base	Contents	ASCII Code	Description
1H	V	56H	ID PROM identifier,
3H	M	4DH	always "VMEID"
5H	E	45H	(5 characters)
7H	I	49H	
9H	D	44H	
BH	A	41H	Manufacturer's ID
DH	-	2DH	always "A-B" for Allen-Bradley
FH	B	42H	(3 characters)
11H	6	36H	Manufacturer's Model Number
13H	0	30H	(7 characters)
15H	0	30H	
17H	8	38H	
19H	L	4CH	
1BH	T	54H	
1DH	V	56H	
1FH	1	31H	Number of 1K byte blocks of Slave 0 memory (1 character)
21H	blank	20H	Major revision level
23H	A	41H	with leading blank
25H	A	41H	Minor functional revision
27H	blank	20H	with trailing blank
29H	P	50H	PCL followed by
2BH	C	43H	blank and station number*
2DH	L	4CH	(6 characters)
2FH	blank	20H	
31H	X		XX = Value of
33H	X		00H-3FH
35-3FH	Not Available		Reserved for future use

Bytes 31 and 33 indicate the PCL station number that you set using switches on the top of the LTV. Refer to chapter 9 **Switch Settings,** for the section entitled **SW1 PCL Station Number.** The LTV displays the station number in ASCII Hex. The programming terminal's Who function displays it in octal.

When addressing the LTV, the host CPU's software must add 80H (128 decimal) to the PCL station number.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Bit/Byte Descriptions of Slave 0 Memory

We describe the function of Slave 0 bytes in numerical order. Hex byte numbers represent the offset from the Slave 0 base address.

Byte	Bit	Description

LTV ID information in permanent memory

00-3FH (odd)		ID information loaded by the LTV at power-up using ASCII characters. See **Module ID Block**, above.

Semaphore and command block pointer written by host CPU

40H	Bit 07	**Semaphore bit** is set by the host CPU when it sends a command to the LTV. The LTV resets it automatically after writing the command block pointer into its private memory.
41H		**Address modifier** (AM) for the command block address pointer. Enter a single AM in hex. Refer to the **Hex Code** column in table 5.1, below. For example, enter 2DH for short supervisory access.
42H		Zero

For bytes 43-45H below, the 24-bit standard memory address for the command block uses all three address bytes. Place the most significant address bits in byte 43H. If using the 16-bit short memory address for the command block, enter a 16-bit address in bytes 44H and 45H with the most significant address bits in byte 44H.

43H		**CB Pointer** A23-A16 (Hi address byte)
44H		**CB Pointer** A15-A8 (Mid address byte)
45H		**CB Pointer** A7-A0 (Lo address byte)

Handshaking for loading Slave 0

46H	Bit 07	**Handshake** The host CPU sets this bit after loading values into bytes 48-4FH.
	Bit 06	**Handshake** The LTV sets this bit after moving values from bytes 48-4FH into private RAM and initializing Slave 1 and/or setting VMEbus mastership options for accessing command blocks.
47H		Zero

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

VMEbus mastership options for executing selective command blocks if you use an external global memory module
(* = default) (Omit for internal global memory.)

48H Bit 07 **Select ROR or RWD**

Set = Select release on request ROR
Reset* = Select release when done RWD

ROR means that the LTV releases the VMEbus after the transfer is complete and another VMEbus master requests the bus.

RWD means that the LTV releases the VMEbu right after completing the transfer.

Bit 06 **Select ROC**

Set = Select release on clear ROC
Reset* = ignore the bus clear signal

ROC means that the LTV releases the VMEbus immediately after the current bus cycle upon detecting the bus clear BCLR signal.

Ignore means the LTV will not release the VMEbus when the BCLR signal is asserted until its normal release.

Bits 05-04 **Select Data Size** of VMEbus master cycle

05	04	Data Size
0	0	16-bit word transfer
0	1	8-bit odd and even byte transfer starting with the selected byte

8-bit odd-byte transfer (not listed) is for copy commands, only. If used for command blocks, the LTV reports an error.

Parameters that define Slave 1, if you use it

48H Bit 03 **Select Address Size** of Slave 1

Set = standard 24-bit address
Reset* = short 16-bit address

49H Bits 02-00 **Select AM Group Number** of Slave 1

Enter a group number 7, 5, 3, or 2 in hex. Your AMs selected in bytes 4AH and/or 4BH below must be in this group. See **Selecting Slave 1 Address Modifiers** below.

4AH **Select AM BT Subgroup Numbers of Slave 1**
(Set to zero unless user-defined)

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

If using a user-defined AM group number (byte 49H) you must define your own AMs which include block transfer AMs. Use this byte to define block transfer AMs. See **Selecting Slave 1 Address Modifiers** below.

AMs in bytes 4AH and/or 4BH must all have the same group number entered in byte 49H.

4BH **Select AM Subgroup Numbers** of Slave 1

Select up to eight AMs with the same group number entered in byte 49, and include those in byte 4AH if used. See **Selecting Slave 1 Address Modifiers** below.

4C **Zero**

In bytes 4D-4FH below, the 24-bit standard global memory address for Slave 1 uses all three address bytes. Place the most significant address bits in byte 4DH. If using the 16-bit short global memory address for Slave 1, enter the 16-bit address into bytes 4EH and 4FH with the most significant address bits in byte 4EH. Bytes 4D-4FH define Slave 1 starting address.

4D **Slave 1 Global Memory Address** (Hi byte)

4E **Slave 1 Global Memory Address** (Mid byte)

4F **Slave 1 Global Memory Address** (Lo byte)

For example, you would enter a 24-bit global memory address such as A12345H into bytes 4D-4F:

```
                  | A  1 | byte 4D
byte 4E | 2  3    | 4  5 | byte 4F
```

50-51H **Error codes** These codes report errors that the LTV detects while processing command blocks. These codes are also displayed in the response word of the command block. See **Command Block Error Codes** at the end of chapter 6.

52-3FF User-defined memory of 942 bytes (decimal).

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Configuring Slave 0 Global Memory

If using the internal global memory of Slave 0 (bytes 52H and above), omit the next section and go directly to **Starting Address of Slave 0 Memory** and **Address Modifiers of Slave 0 Memory**, below. You configure Slave 0 Memory with switches on the LTV module.

Configuring Slave 1 or VMEbus Options

Once you decide whether you will use Slave 1 or an external global memory module, you must implement your decision by loading appropriate information into Slave 0. You must do this with the host CPU's driver program.

Configuring Slave 1, If Using It

You must configure Slave 1 by loading its starting address information into Slave 0

Address size in byte 48H bit 03

reset (default) = short 16-bit
set = standard 24-bit

Starting Address in bytes 4D-4FH

Address Size	4DH	4EH	4FH
24-bit address	MS	mid	LS
16-bit address	-	MS	LS

where MS, LS = most, least significant digits

If you want to place Slave 1 contiguous to Slave 0, use the 16-bit address for Slave 1 and set the address to Slave 0 base + 400H. Otherwise, start Slave 1 on 100H (256 decimal) byte boundaries at an address above the ending address of Slave 0 using the 16-bit or 24-bit address size. The highest starting address of Slave 1 that lets its entire 4K byte memory be used is FFE000H.

Address modifiers in bytes 49-4BH

group number in 49H bits 02-00
subgroup number in 4BH
BT subgroup number, if used, in 4AH

Refer to **Selecting Slave 1 Address Modifiers**, below for selecting standard, short, or user-defined address modifiers.

IMPORTANT: The host CPU must load or change parameters in Slave 0 at power up and whenever the VMEbus backplane is reset. After writing these parameters, the host CPU must set handshake bit 07 (byte 46H) in Slave 0. Then

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

the LTV sets handshake bit 06 (byte 46H) to acknowledge that it configured Slave 1 or VMEbus mastership options and stored the parameters in the VME Status File.

You may want to monitor these parameters with your programming terminal. If so, refer to **VME Status File** in chapter 9.

Configuring VMEbus Mastership Options If Using External Global Memory for Executing Selective Commands

Select the LTV's VMEbus mastership options by setting specified bits in Slave 0, byte 48H: (default = reset)

VMEbus release with bit 07

set = release on request (ROR)
reset (default) = release when done (RWD)

VMEbus release on BusCLeaR with bit 06

set = release on BCLR signal
reset (default) = ignore BCLR signal

Data size of VMEbus master cycle with bits 05, 04

Data Size	05	04
16-bit word transfer	0	0
8-bit odd and even byte transfer starting with the selected byte	0	1

IMPORTANT: 8-bit odd-byte transfer (not listed) is for copy commands, only. If used for command blocks, the LTV reports an error.

IMPORTANT: The host CPU must load or change parameters in Slave 0 at power up and whenever the VMEbus backplane is reset. After writing these parameters, the host CPU must set handshake bit 07 (byte 46H) in Slave 0. Then the LTV sets handshake bit 06 (byte 46H) to acknowledge that it configured Slave 1 or VMEbus mastership options and stored the parameters in the VME Status File.

You may want to monitor these parameters with your programming terminal. If so, refer to **VME Status File** in chapter 9.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Selecting Slave 1 Address Modifiers

If you use Slave 1 memory, your driver program must initialize its address modifiers (AMs) in bytes 49H, and/or 4AH, 4BH of Slave 0. Address modifier codes are listed in Table 5.1. The host CPU accesses Slave 1 with these address modifiers.

IMPORTANT: For most applications, use address modifiers from groups 7 and 5, and omit the AM block transfer subgroup entry.

You select multiple AMs identified by a 3-digit AM Group Number field (in byte 49H) and an 8-digit Subgroup Number field (in bytes 4AH and/or 4BH).

Select one address modifier group number (7, 5, 3, or 2) by entering this number in byte 49H, bits 02-00. Select group 7 or 5 for standard or short VMEbus address modifiers; or group 3 or 2 for user-defined address modifiers. Do not select groups 6, 4, 1, and 0. They are reserved by the VMEbus Specification, Revision C.1.

Select each AM subgroup by setting the corresponding bit 0-7 (in Bit Column, Table 5.1) and then entering the resulting hex value. For example, select 1F, 1E, 1B, 1A, 19 by selecting AM group number 3 and setting bits 7, 6, 3, 2, 1 (1100 1110) and entering CEH for the AM__SUBGRP.

Use the following table to select your address modifier (AM) subgroups. Enter a 1 for each desired AM, a 0 for all other AMs. Then convert the bit pattern to Hex.

```
__AM Group 7       3F 3E 3D    3B 3A 39
__AM Group 5             2D          29
__AM Group 3       1F 1E 1D 1C 1B 1A 19 18
__AM GROUP 2       17 16 15 14 13 12 11 10
AM_SUBGRP =        __ __ __ __ __ __ __ __  Convert to Hex
AM_BTSUBGRP =      __ __ __ __ __ __ __ __  Convert to Hex
```

Referring back to the example, and assuming that 1FH and 1BH were user-defined block transfer AM subgroups, you would enter:

```
X_AM Group 3       1F 1E 1D 1C 1B 1A 19 18
AM_SUBGRP =        _1 _1 _0 _0 _1 _1 _1 _0  Convert to Hex = CEH
AM_BTSUBGRP =      _1 _0 _0 _0 _1 _0 _0 _0  Convert to Hex = 88H
```

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Table 5.1
Address Modifier Codes

The LTV responds to the following address modifiers. Do not mix AM groups when selecting address modifiers for a specified address.

AM Group	Hex Code	Bit	Function
7)	3FH	7	Standard Supervisory Block Transfer
	3EH	6	Standard Supervisory Program Access
	3DH	5	Standard Supervisory Data Access
	3CH	4	Reserved
	3BH	3	Standard Non-privileged Block Transfer
	3AH	2	Standard Non-privileged Program Access
	39H	1	Standard Non-privileged Data Access
	38H	0	Reserved
5)	2FH	7	Reserved
	2EH	6	Reserved
	2DH	5	Short Supervisory Access
	2CH	4	Reserved
	2BH	3	Reserved
	2AH	2	Reserved
	29H	1	Short Non-privileged Access
	28H	0	Reserved
3)	1FH	7	User-defined
	1EH	6	User-defined
	1DH	5	User-defined
	1CH	4	User-defined
	1BH	3	User-defined
	1AH	2	User-defined
	19H	1	User-defined
	18H	0	User-defined
2)	17H	7	User-defined
	16H	6	User-defined
	15H	5	User-defined
	14H	4	User-defined
	13H	3	User-defined
	12H	2	User-defined
	11H	1	User-defined
	10H	0	User-defined

Address modifier groups 6, 4, 1, and 0 are not available because they are reserved by the VMEbus Specification, revision C.1.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Starting Address of Slave 0 Memory

You must set the starting address of the 1K byte Slave 0 global memory with switch assembly SW4 on the bottom of the module (figure 3.1 below). Select the starting address in 400H increments anywhere within the 0000-FC00H byte limits of VMEbus backplane addressing space. You select the six most significant address lines (A15-A10) for these addresses. Set switches 8-3 of SW4 according to your desired starting address as follows:

IMPORTANT: Low = True Logic 0 = on

	Switches/Address Lines					
Short Base	8	7	6	5	4	3
Address (Hex)	A15	A14	A13	A12	A11	A10
0000	0	0	0	0	0	0
0400	0	0	0	0	0	off
0800	0	0	0	0	off	0
0C00	0	0	0	0	off	off
1000	0	0	0	off	0	0
1400	0	0	0	off	0	off
1800	0	0	0	off	off	0
1C00	0	0	0	off	off	off
2000	0	0	off	0	0	0
2400	0	0	off	0	0	off
2800	0	0	off	0	off	0
2C00	0	0	off	0	off	off
3000	0	0	off	off	0	0
3400	0	0	off	off	0	off
3800	0	0	off	off	off	0
3C00	0	0	off	off	off	off
4000	0	off	0	0	0	0
:			:			
:			:			
7C00	0	off	off	off	off	off
8000	off	0	0	0	0	0
8400	off	0	0	0	0	off
8800	off	0	0	0	off	0
8C00	off	0	0	0	off	off
9000	off	0	0	off	0	0
9400	off	0	0	off	0	off
9800	off	0	0	off	off	0
9C00	off	0	0	off	off	off
A000	off	0	off	0	0	0
A400	off	0	off	0	0	off
A800	off	0	off	0	off	0
AC00	off	0	off	0	off	off
B000	off	0	off	off	0	0
:			:			
:			:			
F000	off	off	off	off	0	0
F400	off	off	off	off	0	off
F800	off	off	off	off	off	0
FC00	off	off	off	off	off	off

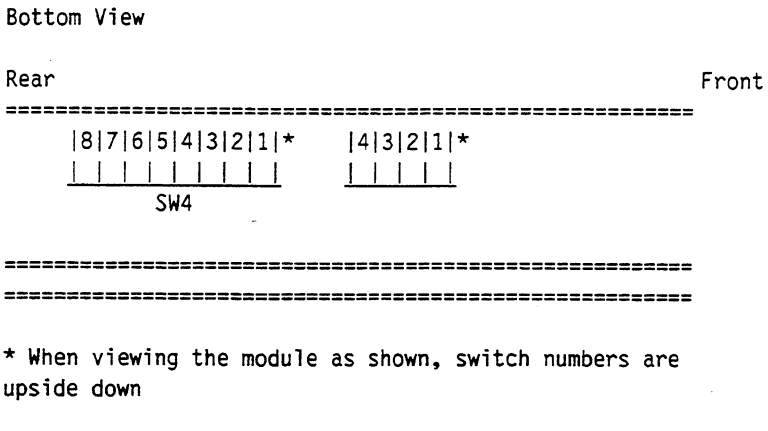
Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Address Modifiers of Slave 0 Memory

The host CPU uses one or both of the following address modifiers defined by the VMEbus Specification when writing to the LTV's Slave 0 global memory to initiate a command. You must select the address modifier(s) of the short global memory using switch assembly SW4 on the bottom of the module (figure 3-1). Set switches 2 and 1 as follows:

Address Modifier	Switch 2	1
Isolates LTV from VMEbus slave access	on	on
29 short non-privileged access	on	off
2D short supervisory access	off	on
29 and 2D	off	off

Figure 3.1
Location of Switch Assembly SW4



* When viewing the module as shown, switch numbers are upside down

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

VMEbus Commands

We give you two types of commands for your host CPU driver program: copy commands to configure the LTV for the continuous transfer of a data file, and selective commands for the one-time transfer of LTV program and/or data files.

NOTE: The LTV can operate solely as a PLC-5/15 processor with no VME interface if required by your application. In that case, ignore this manual and refer to the PLC-5 Family Processor Manual, publication 1785-6.8.2.

Copy Commands These commands configure the LTV to read or write contiguous data files (up to 500 words) each I/O scan. The LTV does not recognize file boundaries so you can transfer more than one type of file. Your host CPU's software driver program issues the command once to configure the transfer. The host CPU transfers the command to the LTV via global memory. The LTV must be in run mode and the enable bit set to execute the transfer. The LTV transfers data until your software driver program issues another command to stop, or the LTV resets the enable bit, or the LTV is switched to program mode.

- Copy-to-VME Command: for continuously copying an LTV data file into VMEbus global memory
- Copy-to-LTV Command: for continuously copying a block of VMEbus global memory into a specified LTV data file

The LTV can initiate, change, or stop copy transfers from its ladder program. Refer to chapter 4 and **VME Status File** in chapter 9 for information on how you do this.

Selective Commands We give you several commands for initiating a single transaction, such as reading or writing data files. Multiple transfers are needed to uploading or downloading LTV memory. These commands require that your host CPU's driver program assemble command packets and interpret reply packets for each command sent to the LTV.

The rest of this chapter describes how the host CPU and LTV implement these commands across the VMEbus backplane. Each command has a **command block** that controls the transfer and a **command sequence** that you must follow when writing the driver program for the host CPU.

We describe copy commands first, then selective commands.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Copy Commands

This section describes the copy command block and the logical sequences required of the host CPU and LTV to execute a copy command.

Copy Command Block

The command block controls handshaking and the transfer of information between the host CPU and LTV. Each time the host CPU initiates a command to the LTV, it writes the command block into VME global memory and writes the address of the command block (CB pointer) into the LTV's Slave 0 memory (bytes 41, 43-45). The command block communicates

the type of command
master/slave handshaking
addresses of source and destination data
number of words to transfer

Word Map of Command Block

```
Word 0     Command word
Word 1     Response word
Word 2     Interrupt level \
Word 3     Status/ID        \_ optional parameters-
                               stored interrupt
Word 4     Reserved, not used
Word 5     Reserved, not used
Word 6     Reserved, not used
Word 7     Control byte and data AM     \
Word 8     VME address: high byte        \
Word 9     VME address: mid and low bytes \_ file address
                                            in global memory
Word 10    Number of words to transfer
Word 11    File number in LTV data file
Word 12    Element number in LTV data file
Word 13    Interrupt level \
Word 14    Status/ID        \_ optional interrupt:
                               LTV completed the transfer
Word 15    Reserved for future use
```

Command Block Bit/Word Descriptions

Word 0 **Command Word**
Specifies type of command

0001H = Copy to VME
0002H = Copy from VME

Word 1 **Response Word**
The host CPU should set this word to zero. The LTV enters the code 00FF in this word to verify acceptance of the command block. If it detects an error while processing the command block, it stops and reports error codes in the high and

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

low bytes of this word and in bytes 50 and 51H of Slave 0. Refer to **Command Block Error Codes** at the end of this chapter.

The host CPU must poll the response word to determine if the LTV accepted the command block and stored all parameters in the VME Status File. (In addition, the LTV can interrupt the host CPU if programmed to do so with words 2 and 3 below.)

Words 2 and 3 below, as an option, let the LTV interrupt the host CPU to acknowledge it has stored all copy command parameters in the VME Status File, or detected an error while processing the command block.

IMPORTANT: Use this interrupt only if you know how your host CPU handles interrupt service routines.

Word 2 **Select Interrupt Level** (parameters stored)
Bits 02-00 Use binary code for levels 1-7.

000 = interrupt not selected
001 = level 1
010 = level 2
: :
111 = level 7

Remaining bits are not used.

Word 3 **Select Status/ID**
FF00-FFFF = 8-bit vector with values 00-FF
XX00-XXFF = 16-bit vector with values XX00-XXFF
where XX = any value except FF

Words 4-6 **Reserved,** not used

Word 7 **Control Bits** in upper byte

These bits control the execution of copy commands and how the LTV accesses the VMEbus.

IMPORTANT: To help retain integrity of data blocks transferred with copy commands, we recommend control bit selections with an (*).

Bit 15 **Enable bit**
Reset = stops the transfer
Set = starts the transfer when the LTV is in run mode

Bit 14 **Select Transfer Point**
Reset = LTV executes transfers after I/O scans
Set = LTV executes transfers before I/O scans

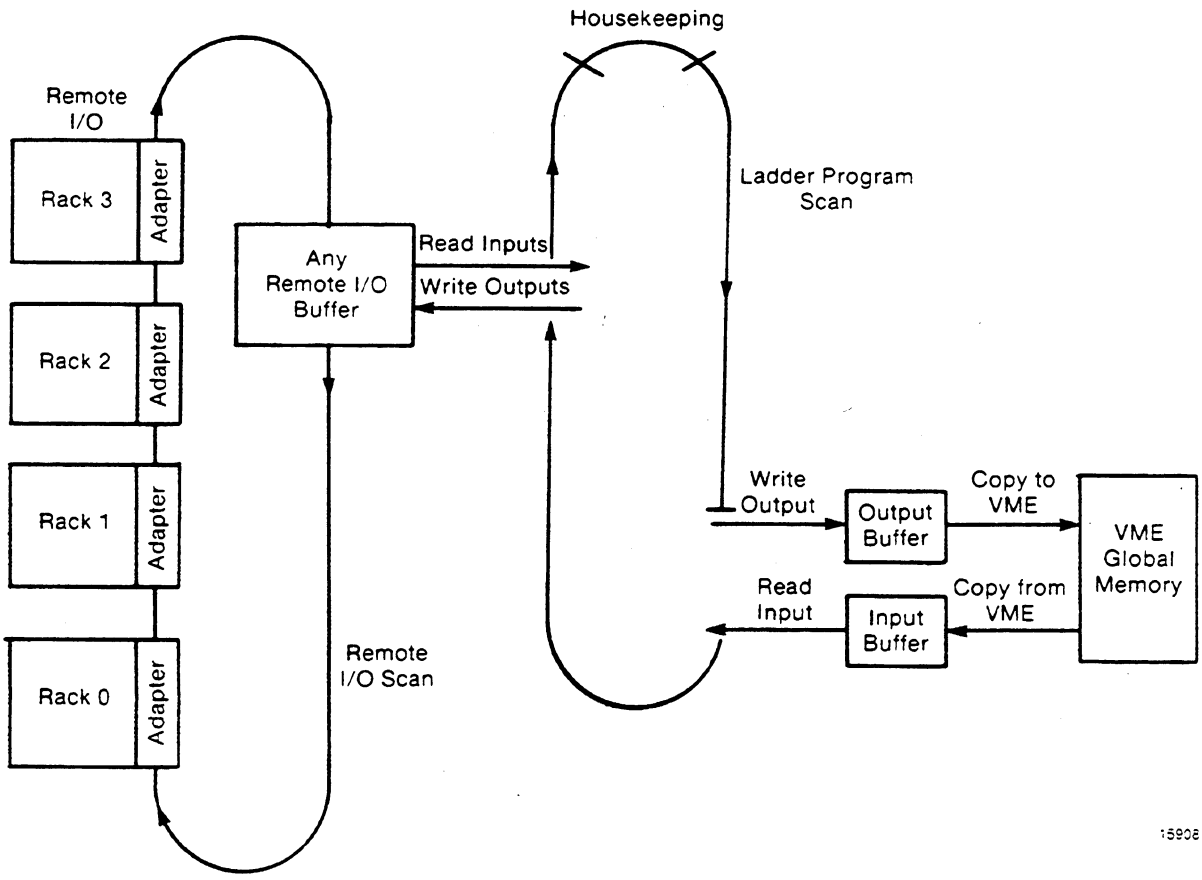
Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

IMPORTANT: The VME Copy Configuration screen presents transfer point incorrectly in 6200 Series Software, Revision 2.1. Correct the display as follows:

Incorrect Revision 2.1		Correct Revision 2.2
before = 0	--->	before = 1
after = 1	--->	after = 0

When you select the same transfer point for both copy commands, the LTV performs the Copy-from-LTV command (read) first, then the Copy-to-LTV command (write) second. Your selection of bit 14 (same for both) determines whether the LTV does copy commands before or after the I/O scan.

When you select a different transfer point for each command, the LTV executes each command according to how you set bit 14 for each.



You can select the VMEbus transfer point before the I/O scan (shown), or after it.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Bit 13 **Select Scan Mode**
When using copy commands and external global memory, you select whether the LTV maintains its ladder program scan time but processes data that may be old (Continuous mode) or delays its ladder program scan and waits for fresh data (Synchronous mode).

Your selection of continuous or synchronous mode takes effect only when the LTV's ladder program scan time is shorter than the time it takes the LTV to gain VMEbus access and complete a copy command when using external global memory.

Normally, the LTV executes copy commands to/from global memory simultaneously with the ladder program scan. Transfer of data between backplane and the LTV's ladder processor occurs only once and is synchronized at the same point of time in the ladder program scan.

When using external global memory, VMEbus access time can be unpredictable if the LTV must wait for another VMEbus master to give up the VMEbus. Your selection governs the operation when VMEbus traffic increases. When using Slave 0 or Slave 1, scan mode selection makes no difference .

Reset* = Continuous: Ladder program scan repeats continuously, independent of VMEbus transfer. The LTV transfers incoming data from global memory to the ladder processor regardless of whether it was updated by the previous ladder program scan. The LTV writes data to global memory regardless of whether the data was bussed to the host CPU during the previous program scan.

Set = Synchronous: Ladder program scan waits for completion of the the last requested VMEbus transfer. The LTV waits to complete the transfer of new data from global memory before executing the ladder program scan, and/or waits for previous data to be transferred to global memory before transferring new data.

IMPORTANT: When debugging your driver program, select continuous mode (bit is reset). Otherwise, the LTV may lock up waiting for a transfer that does not occur. Also let the transfer occur after the I/O scan (set bit 14).

Refer to appendix B for copy command scan timing considerations.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

IMPORTANT: The VME Copy Configuration screen presents scan mode incorrectly in 6200 Series Software, Revision 2.1. Correct the display as follows:

Incorrect Revision 2.1		Correct Revision 2.2
discontin = 0	--->	synch = 1
contin = 1	--->	contin = 0

Bit 12 **Not Used**

Bit 11 **Select Bus Release ROC**
Set = Select release on bus clear signal
Reset* = Ignore bus clear signal

During transfer to an external global memory module, ROC means that the LTV releases the VMEbus immediately after the current bus cycle when the bus clear BCLR signal is asserted.

Ignore means that the LTV will not release the VMEbus while in the middle of reading or writing the data file.

Bit 10 **Select Bus Release RWD or ROR**
Set = Release on request (ROR)
Reset = Release when done (RWD)

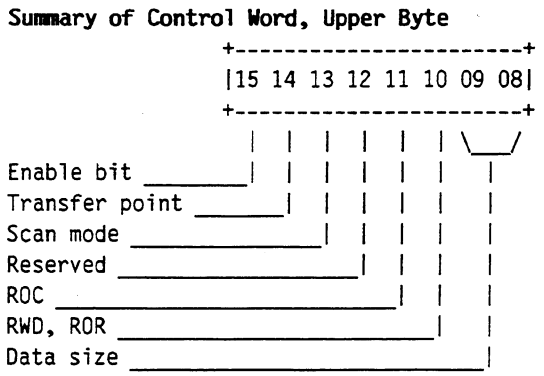
After completion of a transfer from an external global memory module, ROR means that the LTV retains VMEbus access until the host CPU or a VMEbus master requests the VMEbus.

RWD means that the LTV releases the VMEbus and must regain access each transfer. For copy commands, this is slower and less desirable.

Bits 09,08 **Select Data Size** for VMEbus master cycle

09	08	Data size
0	0	16-bit word transfer
0	1	8-bit odd and even byte transfer starting with the selected byte
1	0	8-bit odd byte transfer starting with the (next) odd byte

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com



Use words 7 (low byte), 8 and 9 below to point to the data block in VMEbus global memory used to store source or destination data to be transferred.

Word 7 **Select Data Address Modifier** (low byte)
Bits Enter one address modifier in hex.
07-00 Refer to **Table 5.1 Address Modifier Codes.** For example, enter 2DH for short supervisory access.

Word 8 **Data Address**
Bits High byte of 24-bit address,
07-00

Word 9 **Data Address**
Bits Mid and low bytes or 24-bit address,
15-00 or high and low bytes of 16-bit address

Word 10 **Number of Words**
The number of words to be transferred (integer value of 0-500 decimal or 0-250 if you selected odd-byte-only transfers)

Word 11 **LTV Data File Number** for source or destination (integer value 0-999 decimal)

Word 12 **Element Number** in the LTV data file (integer value 0-999)

Words 13 and 14 below, as an option, let the LTV interrupt the host CPU to acknowledge that the LTV has completed a file transfer after each I/O scan.

IMPORTANT: Use this interrupt only if you know how your host CPU handles interrupt service routines.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Word 13 **Select Interrupt Level**
Bits 02-00 Use binary code for levels 1-7

000 = interrupt not selected
001 = level 1
010 = level 2
: :
111 = level 7

Remaining bits are not used.

Word 14 **Select Status/ID**
FF00-FFFF = 8-bit vector with values 00-FF
XX00-XXFF = 16-bit vector with values XX00-XXFF where XX = any value except FF

To help achieve integrity of transferred data with copy commands, we recommend that you use this interrupt to notify the host CPU of the completion of each transfer. You must assure completion of your VMEbus interrupt service routine in less time than the LTV's program scan (approximately 1ms for a one-word program).

IMPORTANT: VMEbus options for accessing command blocks in external global memory are stored in Slave 0, byte 48H, bits 07-04, and in the VME Status File words 2-5. VMEbus options for accessing external global memory for copy commands are stored in control words 6 and 15 in the VME Status File. You can monitor them on the VME Status screen with your programming terminal. Refer to **Bit/Byte Descriptions in Slave 0 Memory** in chapter 5 and to **VME Status File** in chapter 9.

Copy Command Sequence

A copy command configures the LTV for the continuous transfer of data across the VMEbus backplane. To initiate a copy command, the host CPU must transfer copy command parameters into the LTV's VME Status File. The LTV then responds that it stored the parameters (configuration complete). After configuration, the LTV does not transfer data until in run mode with the enable bit set.

Follow the sequence below when writing the driver program to configure the LTV for copy commands. Your software driver program must instruct the host CPU to perform each step. The LTV performs its steps automatically.

<u>The host CPU (programmed)</u>

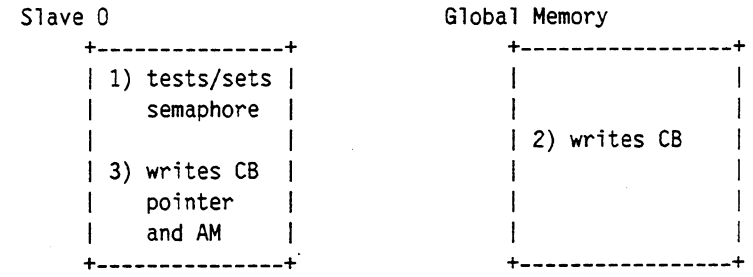
1) tests and sets the semaphore in Slave 0 (byte 40H, bit 07) with a read-Modify-write cycle. If the semaphore is already set, the host CPU must repeat the RMW cycle until the LTV resets the semaphore.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

2) writes the command block (CB) into global memory (address pointer specified in step 3)

3) writes the CB pointer (bytes 43-45H) and the CB pointer's address modifier (byte 41H) into Slave 0 global memory **in that order.**

Graphically, we represent these steps as follows:



The LTV (automatic response)

1) detects a valid address modifier in Slave 0 byte 41

2) places the CB pointer into its internal queue

3) writes an FFH into the address modifier in Slave 0 memory (byte 41H) to prevent copying the same CB pointer twice

4) resets the semaphore in Slave 0 (byte 40H, bit 07)

5) accesses the VMEbus momentarily if the host CPU wrote the command block into external global memory. If written to internal Slave 0 or Slave 1 global memory, the LTV does not take VMEbus mastership but simply moves the command block into internal memory.

6) reads the CB (word 0) to determine the type of copy command

7) writes FFH into the CB response word (word 1) to indicate successful storage of copy command parameters

8) examines CB words 3 and 4 to determine if an interrupt is required to acknowledge successful storage of copy command parameters

The host CPU (programmed)

1) continuously polls the CB response word (word 1) for response code FFH indicating the copy command parameters have been stored successfully, or detects a parameters-stored interrupt

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

IMPORTANT: In order to execute copy commands, the LTV must be in run mode.

In order to re-issue copy commands with the LTV in program mode, the host CPU must repeat the sequence.

In order to re-issue copy commands with the LTV in run mode, the host CPU must repeat the sequence twice: the first time to reset the enable bit, the second time to load new parameters and set the enable bit.

Transferring Data

After configuring the LTV for a copy command and when the LTV is in run mode, the host CPU and LTV transfer the data file across the VMEbus backplane via VMEbus global memory each LTV I/O scan according to the addresses you enter into CB words 7, 8, 9 and 11, 12.

As an option, the LTV can interrupt the host CPU at the completion of each transfer (each I/O scan) if you program this interrupt with words 13 and 14 of the command block.

To determine that a copy command is transferring data correctly each program scan, the LTV's ladder program can poll the error code in the VME Status File, or the host CPU can do this with a selective command. The LTV displays the error code in the VME Status File control word 6 (odd byte) for the Copy-to-VME command and in word 15 (odd byte) for the Copy-from-VME command as follows:

```
07H  Data block in copy command does not exist or
     offset beyond end of data file
09H  Past end of data file or bus error on word write
11H  Bad address modifier
21H  Address error in Slave 0 or Slave 1
41H  Selected more than one arbitration level
FDH  Length greater than 500
FEH  last command-complete interrupt not acknowledged
FFH  Interrupt level greater than 7
```

All other codes indicate a VMEbus transfer error.

For an example driver program to initiate and change copy commands, refer to Chapter 4 **Initiating Copy Commands.** That chapter also shows you how to initiate and change copy commands with ladder program logic.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Selective Commands

This section describes selective commands, the selective command block, and the logical sequences required of the host CPU and LTV to execute a selective command.

Selective commands lets the host CPU

- o read and write data to and from LTV data files
- o read processor status from the LTV
- o change processor mode
- o upload and download memory to and from LTV program files

Selective commands transfer a block of memory over the VMEbus backplane, one transfer per command. Typically, three blocks of VME global memory are associated with each command:

command block
command packet with or without data
reply packet with or without data and/or status

A selective command for writing to the LTV contains the command block, command packet, and data. Typically, the LTV responds with a smaller reply packet that contains status. A command for reading from the LTV contains a command block and command packet. Typically, the LTV responds with a larger reply packet with data and status.

Your host CPU's driver program must allocate global memory space for each command block, command packet, and its corresponding reply packet. We recommend that these three data blocks be placed next to each other in global memory contiguously in the order given.

Selective Command Block

The command block controls handshaking and the transfer of information between the host CPU and LTV. Each time the host CPU initiates a command to the LTV, it writes the command block into VME global memory and writes the address and address modifiers of the command block into the LTV's Slave 0 memory (bytes 41, 43-45). The command block communicates

the type of command
master/slave handshaking
address of command and reply packets
number of bytes in each packet

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Word Map of Selective Command Block

```
Word 0      Command word
Word 1      Response word
Word 2      Interrupt level \
Word 3      Status/ID        \_ optional command-
                                done interrupt
Word 4      Reserved, not used
Word 5      Reserved, not used
Word 6      Reserved, not used
Word 7      Data address modifier        \
Word 8      VME address: high byte        \
Word 9      VME address: mid and low bytes\_ packet address
                                             in global memory
Word 10     Packet length in bytes
Word 11-15  Reserved, Not used
```

Command Block Bit/Word Descriptions

Word 0 **Command word**
Specifies type of command

FFFFH = Selective Command

Word 1 **Response Word**
The host CPU should set this word to zero. The LTV enters the code 00FFH in this word to verify transfer completion. If it detects an error while processing the command block, it stops and reports error codes in the high and low bytes of this word and in bytes 50 and 51H of Slave 0. Refer to **Command Block Error Codes**, below.

Words 2 and 3 below, as an option, let the LTV interrupt the host CPU at the completion of a command. Entering these words enables the option. Otherwise, the host CPU must poll the response word to detect completion.

IMPORTANT: Use this interrupt only if you know how your host CPU handles interrupt service routines.

Word 2 **Select Interrupt level**
Bits 02-00 Use binary code for levels 1-7.

000 = interrupt not selected
001 = level 1
010 = level 2
: :
111 = level 7

Remaining bits are not used.

Word 3 **Select Status/ID**
FF00-FFFF = 8-bit vector with values 00-FF
XX00-XXFF = 16-bit vector with values XX00-XXFF
where XX = any value except FF

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Words
4-6 **Reserved,** not used

Use words 7 (low byte), 8 and 9 below to point to the packet address in VMEbus global memory used to store the source or destination data block.

Word 7 **Select Address Modifier of Packet Address**
Bits Enter one address modifier in hex. Refer to
07-00 table 5.1 in chapter 5. For example, enter 2DH.

Word 8 **Packet Address**
Bits Most significant digits of 24-bit address
07-00

Word 9 **Packet Address**
Mid and low digits of 24-bit address,
or entire 16-bit address

Word 10 **Number of Bytes in Packet**
Includes header and data bytes (depends on the command). Refer to **Packet Format** in chapter 7.

Words 11-15
Reserved, not used

IMPORTANT: VMEbus options for accessing external global memory are stored in Slave 0, byte 48H, bits 07-04; and in the VME Status File. You can monitor them on the VME Status screen with your programming terminal. Refer to **Bit/Byte Descriptions in Slave 0 Memory** in chapter 5 and to **VME Status File** in chapter 9.

Selective Command Sequence

A selective command initiates the one-time transfer of data or LTV memory. In the following sequence, your host CPU's software driver program must instruct the host CPU to perform each step. The LTV performs each step automatically.

The host CPU (programmed)

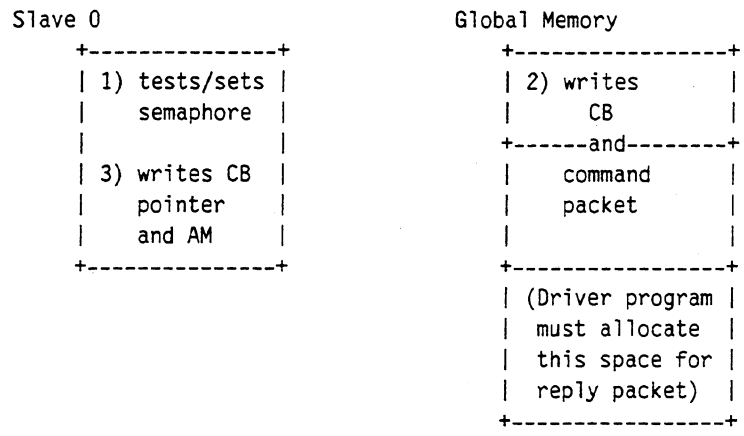
1) tests and sets the semaphore in Slave 0 (byte 40H, bit 07) with a read-Modify-write cycle. If the semaphore is already set, the host CPU must repeat the RMW cycle until the LTV resets the semaphore

2) writes the command block (address pointer specified in step 3) and command packet (address specified in command block) into global memory

3) writes the CB pointer (bytes 43-45H) and the CB pointer's address modifier (byte 41H) into Slave 0 global memory **in that order.**

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Graphically, we show these steps as follows:



The LTV (automatic response)

1) detects a valid address modifier in Slave 0 byte 41

2) places the CB pointer into its internal queue

3) writes an FFH into the address modifier in Slave 0 memory (byte 41) to prevent copying the same CB pointer twice

4) resets the semaphore in Slave 0 (byte 40H, bit 07)

5) accesses the VMEbus momentarily if the host CPU wrote the command block and/or command packet into external global memory. If written to Slave 0 or Slave 1 internal memory, the LTV does not take VMEbus mastership but simply moves the command block and/or command packet into internal memory.

6) reads the CB (word 0) to determine the type of selective command

7) reads the packet pointer address and address modifier (words 7-9) to fetch the command packet

8) executes the command and processes the data

9) writes the reply packet into global memory at a location immediately following the command packet. The LTV gains access to the VMEbus momentarily if the command and reply packets were stored in external memory.

10) writes 00FFH into the CB response word (word 1) to indicate successful completion of the command or writes an error code if it detected an error while processing the command block. Refer to **Command Block Error Codes**, below.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

11) examines CB words 2 and 3 to determine if an interrupt is required to acknowledge successful completion of the command

The host CPU (programmed)

1 polls the CB response word (word 1) for response code FFH indicating a completed command or some other non-zero code indicating an error, or detects the command-complete interrupt (words 2 and 3)

2) upon detecting successful completion by polling or interrupt, accesses the VMEbus to read the reply packet stored in internal or external global memory

If an error is reported in the reply packet, the host CPU should examine the status byte and extended status byte to determine the nature of the error. Chapters 7 and 8 describe selective commands and their error codes.

Refer to chapters 7 and 8 for the method of formatting command and reply packets, and to appendix A for an example driver program.

Command Block Error Codes

When the LTV detects an error while processing the command block, it reports the error code in the even and odd bytes of the response word that it writes into the command block. It also reports these error codes in Slave 0 bytes 50H and 51H.

If the response word (word 1 in the command block) returns the value 00FFH, the LTV detected no errors and the LTV completed the command.

If the LTV detects an error, it stops processing the command and reports two error catagories: parameter errors and logic errors.

You must determine where these errors occurred, correct them, and issue the command again.

Parameter Errors (Reported in even byte of CB word 1, and in Slave 0, byte 50)

```
01H  Bad address in Slave 0 or Slave 1 global memory
02H  Bad AM in external global memory
04H  Bad command word (word 0) in command block
05H  Bad packet length (word 10)
06H  Bad LSAP or SRC value
07H  Bad response packet address
08H  Bad VME interrupt level (word 2)
09H  Bad file number (word 11) for copy command block
0AH  Copy command block did not write to file
```

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

0BH More than one arbitration level
80H VMEbus error

Logic Errors (Reported in odd byte of CB word 1, and in Slave 0, byte 51H) The error occurred at the point in your driver program logic when it commanded the LTV to do the following function.

00H perform an interrupt command
01H wait for completion of interrupt command
02H get command block
03H wait for completion of get command block
04H get command packet
05H wait for completion of get command packet
06H input buffer busy, try again (driver OK)
07H input buffer full, wait for reply (driver OK)
0AH perform new data interrupt
0BH wait for completion of new data interrupt
0CH report bad-result word for interrupt command
0DH wait for completion of report bad-result word, and reset code to 10H
0EH write reply packet into VMEbus global memory
0FH wait for completion of write reply packet
10H write result word
11H wait for completion of write result word
12H perform command complete interrupt
13H wait for completion of command complete interrupt
14H report error in result word (CB word 1 and Slave 0 bytes 50H and 51H)
15H wait for completion of report error
16H wait for completion of report error interrupt

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Introduction

This chapter describes selective commands and the corresponding replies used to transfer data files across the VMEbus backplane. We recommend that you review chapter 6 before starting this chapter.

The selective commands for transferring data across the VMEbus backplane that we cover in this chapter are:

Echo
Identify LTV, Report Status
Write Bit
Read Block
Write Block

The software driver program for your host CPU must assemble byte strings into command packets, and interpret the byte strings of reply packets returned from the LTV.

The software driver program of your host CPU sends the LTV a command by writing a command block and command packet into global memory, and telling the LTV where to find them. The LTV acts on commands in the order that the host CPU writes them. The LTV interprets the command packet and executes the command. It issues a reply packet for each command it receives by writing the reply packet and a command-done code into global memory. It generates an interrupt if programmed to do so. If it cannot execute the command, it generates an error code in its reply packet. The host CPU handles replies from the LTV by processing interrupts or by polling the response word in the command block for the command-done code.

The host CPU assigns a command code and transaction number to each command packet when it writes the command packet into VME global memory. The LTV returns the transaction number and a status code in its reply packet. It writes the reply packet into global memory adjacent to and immediately following the command packet. The reply packet is similar to the command packet except that the LTV adds packet length so the host CPU's driver program knows how long a reply packet to read.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Packet Format

Selective command and reply packets are written into global memory contiguously in byte format. Packets typically contain the following bytes which we describe later in this chapter:

Command Packet

Byte Addr	Byte Abbr	Description
00H	DST	Destination number
01H	PSN	Position (value = 05)
02H	SRC	Source number
03H	PSN	Position (value = 00)
04H	CMD	Command Code
05H	STS	Status code (value = 00 for command)
06H	TNS	Transaction Number (two bytes)
08H	FNC	Function Code
09H	DATA	First byte of up to 244 data bytes If used, the information in this data field varies from one command to another.

Reply Packet

The LTV adds the length of the reply packet in two bytes labeled LNH at the head of the packet. Note that the order of high byte followed by low byte for LNH is not typical.

Byte Addr	Byte Abbr	Description
00H	LNH	Length of packet (hi byte)
01H	LNH	Length of packet (lo byte)
02H	DST	Destination number
03H	PSN	Position (value = 05)
04H	SRC	Source number
05H	PSN	Position (value = 00)
06H	CMD	Command code
07H	STS	Status
08H	TNS	Transaction number (two bytes)
0AH	EXT STS	Extended Status code, if used
0AH	DATA	First byte of up to 244 data bytes if not reporting EXT STS. If used, data in this data field varies from one reply to the other.

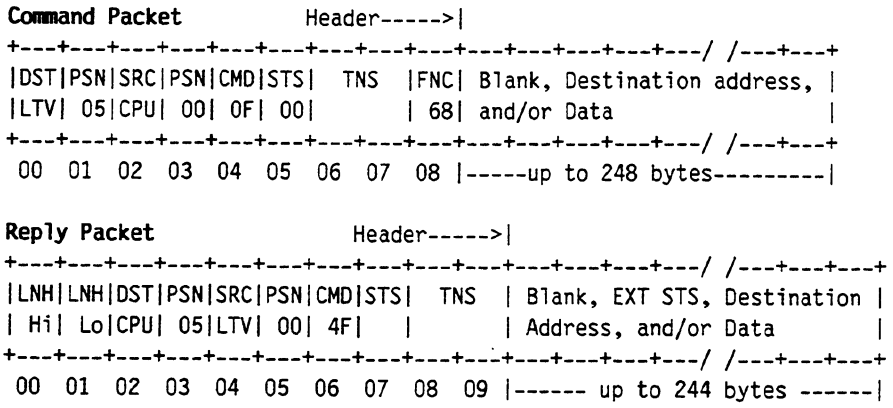
The information above the line is the header. The format is the same for all command and response packets transferred to and from the LTV across the VMEbus backplane. The driver program of the host CPU and the LTV may append data, status, and/or a destination address to the header depending on the type of command.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

We describe header bytes only once in the section **Header Bit/Byte Descriptions**, and appended bytes with the description of each command.

We present packets in horizontal format for space-saving convenience. The general format of command and reply packets are shown below. Bytes are arranged from left to right in the order that they are addressed.

Figure 5.1
Example Command and Reply Packets



Header Bit/Byte Descriptions

Bytes that compose the headers of command and reply packets are described below. Their descriptions will not be repeated in the description of each command that follows. Note that all numbers are decimal except where noted by an H for Hexadecimal.

DST Destination
For a command packet, this byte is the LTV's station number (128-191, 80-BFH).

For a reply packet, this byte is the host CPU's station number (128-191, 80-BFH).

PSN Position
Set first PSN value to 05, second PSN value to 00.

SRC Source
For a command packet, this byte is the host CPU's station number (128-191, 80-BFH).

For a reply packet, this byte is the LTV's station number (128-191, 80-BFH).

CMD Command
CMD and FNC bytes together define the command to be executed. Command codes are included in command descriptions later in this chapter.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

The reply packet also contains the CMD byte. The LTV copies the CMD value from the command packet into the corresponding reply packet.

Bit 07 Always zero

Bit 06 Designates command or response.
The host CPU resets this bit when sending a command. The LTV sets this bit to 1 when sending a reply.
(0 = command, 1 = reply)

Bits 05,04 Not Used (set to zero)

Bits 03-00 Command codes (in Hex)
Use command codes with function codes FNC to specify the type of command.

STS Status
If the LTV detects an error, it reports error codes in the reply packet. Zero means no error. Error codes are described for each command, below.

Set to zero in the command packet.

STS and EXT STS (extended status) are returned in the reply packet in response to some commands. STS bits 07-00 contain the value F0H when reporting extended status. Status and extended status codes that could be returned in the reply packet are described for each command, below.

TNS Transaction code (two bytes)
The host CPU's driver program should generate a unique 16-bit number for each transaction so that it can match replies to corresponding commands. There should not be more than one active packet with the same transaction number from any source.

Whenever the LTV receives a command, it copies the TNS value of the command packet into the same field of the corresponding reply packet without changing the TNS value.

FNC Function code
For a command packet, it combines with the CMD byte to define the command. See CMD, above.

EXT STS Extended status code
If the LTV detects an error, it reports extended status codes in the reply packets of some commands. See STS, above.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Selective Commands

The following table lists selective commands according to how you use them. We also include the command code (CMD) and function code (FNC) that identify each command to the host CPU and LTV.

Command	CMD	FNC
Diagnostic Commands		
Echo	06H	00H
Identify LTV, Report Status	06H	03H
Reads and Writes to a Specified Data File		
Write Bit	0FH	26H
Write Block	0FH	67H
Read Block	0FH	68H

The rest of this chapter describes these commands in detail. Chapter 8 describes commands for uploading and downloading LTV memory across the VMEbus backplane.

IMPORTANT: In the command descriptions that follow, the bytes of the command and reply packets are presented horizontally from left to right in the order they are transferred across the VMEbus backplane. Byte 0 is at the left. CMD and FNC values are expressed in hex. All other values are decimal.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Echo
(06H 00H)

Use this command to check integrity of transmissions across the VMEbus backplane. Also, it is useful when debugging your software driver program because it is the easiest of the selective commands to program. It receives the identical data that it sends.

Command Packet

```
+---+---+---+---+---+---+---+---+---+---+--/ /--+---+
|DST|PSN|SRC|PSN|CMD|STS|  TNS  |FNC|   DATA        |
|   | 05|   | 00| 06| 00|       | 00|up to 244 bytes|
+---+---+---+---+---+---+---+---+---+---+--/ /--+---+
```

Reply Packet

```
+---+---+---+---+---+---+---+---+---+---+---+--/ /--+---+
|LNH|LNH|DST|PSN|SRC|PSN|CMD|STS|  TNS  |   SAME DATA   |
| Hi| Lo|   | 05|   | 00| 46|   |       |up to 244 bytes|
+---+---+---+---+---+---+---+---+---+---+---+--/ /--+---+
```

Refer to **Header Bit/Byte Descriptions**, above for descriptions of all bytes.

Error Codes

This command does not return error codes.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Identify LTV, Report Status (06H 03H)

Use this command to report useful information about the LTV with which the host CPU communicates. You can use it as a diagnostic command when debugging your host CPU's driver program. You can also use it to confirm communication with the specified LTV, identify its operating mode, and report other useful information before initiating an upload or download (described in chapter 8).

Command Packet

```
+---+---+---+---+---+---+---+---+---+
|DST|PSN|SRC|PSN|CMD|STS|  TNS  |FNC|
|   | 05|   | 00| 06| 00|       | 03|
+---+---+---+---+---+---+---+---+---+
```

Reply Packet

```
+---+---+---+---+---+---+---+---+---+---+---+---+--//--+---+
|LNH|LNH|DST|PSN|SRC|PSN|CMD|STS|  TNS  | STATUS (17 bytes)|
| Hi| Lo|   | 05|   | 00| 46|   |       |                  |
+---+---+---+---+---+---+---+---+---+---+---+---+--//--+---+
```

Refer to **Header Bit/Byte Descriptions**, above, for descriptions of all bytes except STATUS.

The STATUS field returned in the reply packet indicates the following information.

Byte	Description
1	Operating status of the LTV
	Bits 2-0 000 = program load 010 = run mode 100 = remote program load 101 = remote test 110 = remote run 001, 011, 111 = not used
	Bit 3 0 = no fault 1 = major fault
	Bit 4 0 = not downloading 1 = download mode
	Bit 5 0 = not uploading 1 = upload mode
	Bit 6 0 = not testing edits 1 = testing edits
	Bit 7 0 = no edits in LTV 1 = edits in LTV
2	Processor type
	Bits 0-7 CBH = LTV
3-6	Processor Memory Size (14K) (low word, low byte first)

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

7	Series and revision of LTV	
	Bits 4-0	00000 = Revision A 00001 = Revision B, etc
	Bits 7-5	000 = Series A 001 = Series B, etc
8	Processor station number	
	Bits 5-0	Station number 0-63
9	FDH	Future development
10	01H	Future development
11,12	Number of data files used (highest assigned file number + 1) (low byte first)	
13,14	Number of program files used (highest assigned file number + 1) (low byte first)	
15	Forcing status	
	Bit 0	0 = no forces active 1 = forces active
	Bit 4	0 = no forces present 1 = forces present
	All other bits = 0	
16	00H	Future development
17	RAM invalid	
	Bits 7-0	0 = RAM valid any bit set = invalid RAM

Error Codes

This command does not return error codes.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Write Bit
(0FH 26H)

Use this command to change bits in an element stored at the specified address in an LTV data file. The command tells the LTV to apply a read-modify-write cycle to:

- o read out the data
- o apply an AND mask
- o apply an OR mask
- o return the results to the specified address

The address/mask field (up to 242 bytes) in the command packet contains multiple blocks, each of which contains an LTV file address, a 2-byte AND mask, and a 2-byte OR mask.

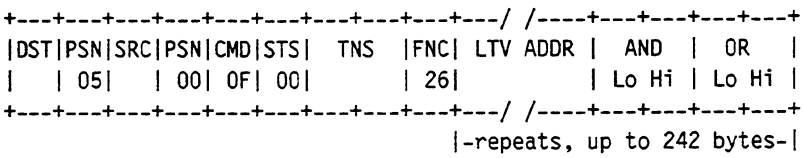
Refer to **Header Bit/Byte Descriptions**, above for descriptions of all bytes except the following.

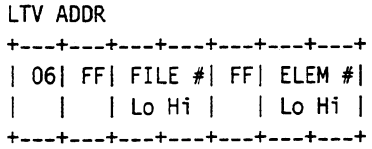
Use the LTV ADDR field to specify the address of the element(s) to be modified. You can use the 242-byte address/mask field to modify selected words in and between data files.

Use the AND mask (2-bytes field) to specify which bits are reset to 0 in the addressed word. A 0 in the AND mask resets the corresponding bit in the addressed word to 0. A 1 in the AND mask leaves the corresponding bit unchanged. Low byte comes first in the AND mask.

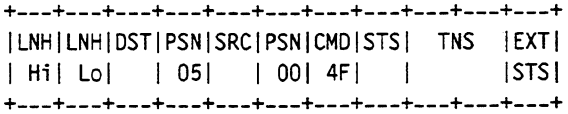
Use the OR mask (2-byte field) to specify which bits to set to 1 in the addressed word. A 1 in the OR mask sets to 1 the corresponding bit the addressed word. A 0 in the OR mask leaves the corresponding bit unchanged. Low byte comes first in the OR mask.

Command Packet





Reply Packet



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Error Codes

Extended status codes are reported in the response packet. The STS byte contains 00H if no error, F0H when the LTV detects an error. If an error, the error code is indicated in the EXT STS byte as follows:

STS	EXT STS	Description
00H	-	no error
F0H	01H	Illegal address - address field has an illegal value
F0H	02H	illegal address - not enough fields specified
F0H	03H	Illegal address - specified too many address levels
F0H	06H	Illegal address - file does not exist
F0H	07H	Beyond end of file
F0H	0BH	Access denied - privilege violation

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Write Block
(0FH 67H)

This command lets the host CPU write file data to the LTV one packet at a time starting at a specified address plus packet offset. Your driver program must re-issue the command for each packet the number of times required to complete the total transaction. It also must manipulate the offset field to place data of each packet in the correct destination location. The LTV automatically checks that the total transaction value does not extend beyond the end of the data file. It does not check for overlap or spaces between packets.

Write-block (and read-block) commands contain a data type ID. The host CPU places the data-type code in the write-block command packet. The LTV places the data-type code in the reply packet of a read-block command. The type of data sent with this write-block command must match the file type written to. The driver program of the host CPU must convert data types when necessary. Refer to **Data Types** below.

IMPORTANT: You may write multiple elements of the same data type in a packet by selecting the data-type ID for the array. You may write one element of a data type in each packet by selecting any of the standard data-type codes such as for integer, timer, counter, control or floating point.

Refer to **Header Bit/Byte Descriptions,** above for descriptions of all bytes except the following.

IMPORTANT: The LTV ADDR, OFFSET, and TOTAL TRANS fields below work together when the total number of words to be written requires multiple packets.

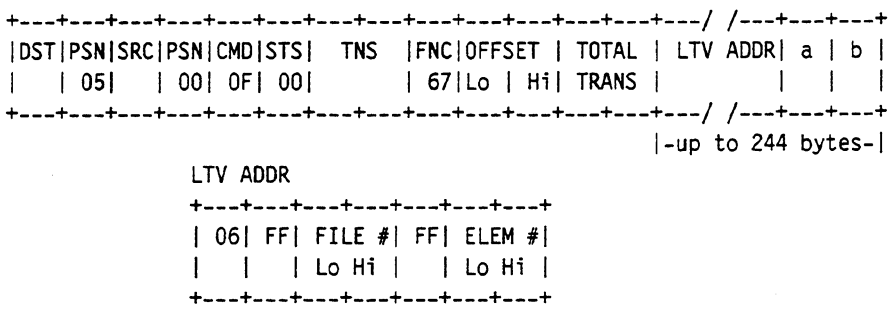
Use the LTV ADDR field to specify the destination file number and first element number. If the total transaction requires more than one packet, keep this address constant and manipulate the OFFSET value.

Use the OFFSET field (2 byte, low byte first) to point to the starting element of each packet when the total transaction requires more than one packet. The offset specifies the number of elements above the base address (LTV ADDR). Set the offset to zero for the first packet and manipulate its value for each successive packet. The LTV does NOT check overlaps or spaces between packets.

Use the TOTAL TRANSaction field (2 bytes, low byte first) to specify the number of DATA elements (excluding ID bytes) of the total transaction. By specifying the total transaction in the first of multiple packets, the LTV can generate an error code if the total transaction value will exceed the end boundary of the destination file.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

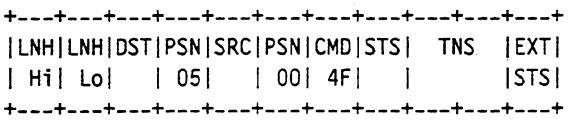
Command Packet



a - data type ID code byte(s) See **Data Types**, below

b - DATA byte field. See **Data Types**, below.

Reply Packet



Error Codes

Extended status codes are reported in the reply packet. The STS byte contains 00H if no error, F0H when the LTV detects an error. If an error, the error code is indicated in the EXT STS byte as follows:

STS	EXT STS	Description
00H	-	no error
F0H	02H	Illegal address - not enough fields specified
F0H	03H	Illegal address - specified too many address levels
F0H	06H	Illegal address - file does not exist
F0H	07H	Illegal address - beyond the end of the file
F0H	0BH	Access denied - privilege violation
F0H	11H	Mismatched data type

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Read Block
(0FH 68H)

This comma
LTV one pa
plus offset.
command fo
complete th
the offset fi
LTV autom
transaction
the data file
as an array.

This read-bl
type ID. Th
write-block
code in the
to **Data Typ**
read-block c
data. The d
data types v

Refer to **Hea**
descriptions

IMPORTAN
fields below
to be read re

Use the LTV
file data to
more than o
manipulate

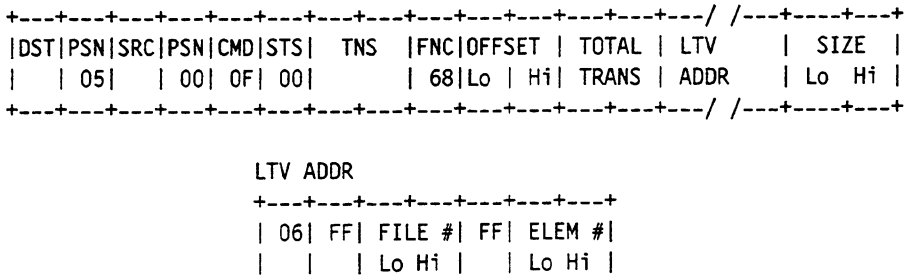
Use the OFF
the starting
transaction
specifies the
(LTV ADDR
and manipu
LTV does N
packets.

Use the TOT
to specify th
bytes) of the
transaction
generate an
will exceed t

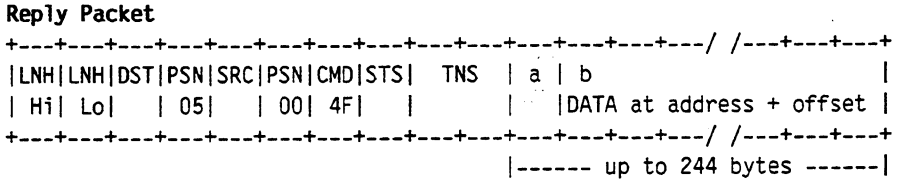
Use the SIZE
the number
reply packet
data in resp

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Command Packet



Reply Packet



a - data type ID code byte(s) See **Data Types**, below.
If there is an error, this field indicates EXT STS extended status and no data is returned in field b.

b - DATA is returned starting at the LTV ADDR plus OFFSET, low byte then high byte for each word. The LTV returns an array of the specified data type containing the number of elements specified by the SIZE byte field. See **Data Types**, below.

Error Codes

Extended status codes are reported in the reply packet. The STS byte contains 00H if no error, F0H when the LTV detects an error. If an error, the error code is indicated in the EXT STS byte as follows:

STS	EXT STS	Description
00H	-	no error
F0H	03H	Illegal address - specified too many address levels
F0H	06H	Illegal address - file does not exist
F0H	07H	Illegal address - beyond the end of the file
F0H	0AH	Transaction size is too large for a packet
F0H	0BH	Access denied - privilege violation

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Data Types

Data types are those resident in the LTV. In the write-block and read-block commands described above, each data type has a code representing its ID. The data-type code is stored in byte field "a" of the command or reply. Some data types have a corresponding size. The data-type size is the number of bytes required to store one element of the data type.

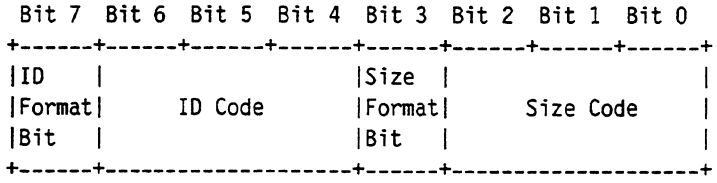
The field that stores the data-type ID and size codes has a default length of one byte for ID and size codes 3-7. When the code exceeds 7, additional bytes are appended to the default byte to specify ID and size. We describe this in **Data-Type Field,** below.

ID Code	--Data-Type-- Abbr	Size	Description
	Specified in Default Byte		
3	A	1	ASCII
4	N,S,I,O	2	Integer (signed, two's complement) includes status and I/O data
5	T	6	A-B timer
6	C	6	A-B counter
7	R	6	A-B control
	Specified in Appended Bytes		
8	F	4	Floating point (IEEE single precision)
9	-	-	Array (specifies data type and size)
10-15	-	-	Reserved
16	D	2	BCD

IMPORTANT: If you want to write one element of a data type per packet, select any of the standard data-type codes such as for integer, timer, counter, control or floating point. If you want to write multiple elements of the same data type per packet, select the data-type code for the array. You specify the data-type and size codes of any standard data type in the array.

Data-Type Field

The data-type field specifies the ID (type of data) and size (number of bytes per element) of the data type used in these write-block and read-block commands. The default data-type field (1-byte) contains a format bit and value field for defining ID and size.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

The data-type field can vary in length if more descriptor bytes are required. Either of two format bits (bit 7 and/or 3) distinguish between a 1-byte or multi-byte field. If the format bit is zero, then the adjacent 3-bit field contains a binary code (0-7) that specifies the data type ID or size. If the format bit is 1, then the adjacent 3-bit field defines the number of descriptor bytes appended to the default byte. The appended descriptor bytes specify the ID or size. The order of descriptor bytes is least to most significant. The most significant (MS) bytes of zero value are permitted but overlooked.

When both the ID and size codes are appended, the ID bytes precede the size bytes.

For example, the following data type descriptor fields have identical value. They describe an ID code of 4 (integer) and a size code of 2 (bytes per element).

```
Bit 76543210        Bit 76543210        Bit 76543210
   +--------+          +--------+          +--------+
   |01000010|          |01001001|          |01001010|
   +--------+          +--------+          +--------+
                       |00000010|          |00000010|
                       +--------+          +--------+
                                           |00000000|
                                           +--------+
```

Example Data Types

We now present examples of several data-type IDs and corresponding data field (fields a and b in the command or reply packets). Note that the packet for a write-block command is limited to one element of a specified data type except for the array and character string.

Integer Example The first byte is the data-type field (field a), the 2-byte element contains the data (field b).

```
Bit  76543210
    +--------+ ID = 4 for integer
a   |01000010| Size = 2 bytes per element
--  +--------+
b   |11111110|  LS
    +--------+
    |00000000|  MS
    +--------+
  value = 254
```

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Floating-Point Example The first two bytes are the data-type field (field a), the 4-byte element contains the data (field b).

```
Bit  76543210
    +--------+ ID in next one byte
a   |10010100| Size = 4 bytes per element
    +--------+
    |00001000| ID = 8 for floating point
--  +--------+
b   |11111110| LS
    +--------+
    |11111110|
    +--------+
    |11111110|
    +--------+
    |11111110| MS
    +--------+
value not computed
```

Control Structure Example The first byte is the data-type field (field a), the 6-byte element contains the data (field b).

```
Bit  76543210
    +--------+
a   |01110110| ID = 7 for control, Size = 6 bytes
--  +--------+
b   |00000000| word 0 (LS)
    +--------+
    |00000000| (MS)
    +--------+
    |00000000| word 1 (LS)
    +--------+
    |00000000| (MS)
    +--------+
    |00000000|  word 2 (LS)
    +--------+
    |00000000| (MS)
    +--------+
value = 0
```

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Counter Example The first byte is the data-type field (field a), the 6-byte element contains the data (field b). Bits in the control word are:
15 = up counter enabled, 14 = down counter enabled,
13 = counter done, 12 = overflow, 11 = underflow

```
Bit  76543210
    +--------+
a   |01100110| ID = 6 for counter, Size = 6 bytes
--  +--------+
    |00000000| Control byte (LS)
    +--------+
    |10000000| Control byte (MS)
    +--------+
    |00000000| Preset (LS)
    +--------+
    |00000001| (MS)
    +--------+
    |00000111|  Accumulated (LS)
    +--------+
    |00000000| (MS)
    +--------+
value: up counter enabled, not done, no
overflow/underflow, preset = 256, accumulated = 7
```

Timer Example The first two bytes are the data-type field (field a), the 10-byte element contains the data (field b). Bits in the control word are:
15 = timer enabled, 14 = timer timing, 13 = timer done
9 & 8 = time base (1 0 for 1 second).

```
Bit  76543210
    +--------+ ID = 5 for timer
a   |01011001| Size in next one byte
--  +--------+
b   |00000110| Size = 6 bytes per element
    +--------+
    |00000000| Control byte (LS)
    +--------+
    |11000010| Control byte (MS)
    +--------+
    |00001010| Preset (LS)
    +--------+
    |00000000| Preset (MS)
    +--------+
    |00001001| Accumulated (LS)
    +--------+
    |00000000| Accumulated (MS)
    +--------+
value: Timer enabled, timing, not done, preset = 10 sec
accumulated = 9 sec.
```

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Array Example The array includes two ID descriptors, the first specifies the structure as an array and its total length, the second specifies the type of data in the array and the number of bytes per element. You must count the second descriptor as part of the data field.

IMPORTANT: Select the array structure when transferring multiple elements of the same data type.

In this example, the first byte is the data-type field and specifies size (total number of data bytes including second descriptor), the second byte is the ID descriptor for the array (both bytes in field a), the third byte is the ID descriptor for the data type followed by data bytes (field b).

```
Bit 76543210
     +--------+ ID in next one byte
a    |10010111| Size = 7 bytes including second descriptor
     +--------+
     |00001001| ID = 9 for array
---  +--------+
b a  |01000010| ID = integer, size = 2 bytes per element
  -  +--------+
  b  |00000000| Integer 0 (LS)
     +--------+
     |00000000| (MS)
     +--------+
     |11111110| Integer 1 (LS)
     +--------+
     |11111111| (MS)
     +--------+
     |11111111|  Integer 2 (LS)
     +--------+
     |00000000| (MS)
     +--------+
value: 0, -2, 255

This array could include enough bytes to fill a packet.
```

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Example of Character String The first byte(s) are the descriptor (field a), followed by the character string (field b). The string is not NULL determined.

```
Bit 76543210
   +--------+ ID = 3
a  |00110011| Size = 3
 --+--------+
b  |01000011| ASCII C
   +--------+
   |01100001| ASCII a
   +--------+
   |01110100| ASCII t
   +--------+
value = Cat
```

```
Bit 76543210
   +--------+ ID = 3 for CS
a  |00111001| Size in next byte
   +--------+
   |00010111| Size = 23 bytes
 --+--------+
b  |01010100| ASCII T
   +--------+
   |01101000| ASCII h
   +--------+
    |01101001| ASCII i
    +--------+
   |01110011| ASCII s
   +--------+
   //      //
   +--------+
   |01101100| ASCII l
   +--------+
   |01100101| ASCII e
   +--------+
   |00101110| ASCII .
   +--------+
value = This is a fine example.
```

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Chapter 8 Selective Commands (Part II)

Introduction

This chapter describes selective commands and their corresponding replies used to upload/download LTV memory across the VMEbus backplane. These commands include:

Command	CMD	FNC
Identify LTV, Report Status	06H	03H
Set Processor Mode	0FH	3AH
Set Upload Privilege	0FH	53H
Physical Read	0FH	17H
Restart after Upload	0FH	55H
Set Download Privilege	0FH	50H
Physical Write	0FH	18H
Restart after Download	0FH	52H

The software driver program for your host CPU must assemble byte strings into command packets, and interpret byte strings of reply packets returned from the LTV.

We recommend that you review **Header Bit/Byte Descriptions** in chapter 7 before reading this chapter because we do not repeat those descriptions for commands described in this chapter.

Criteria for Upload/Download Commands

The following criteria govern what the host CPU should do and how the LTV responds to upload and download commands.

The LTV inhibits

- write commands from a PCL* station to any of its PC program files (ladder or sequential function chart) during an upload or download (The LTV lets a PCL station read from or write to PC data files with a message instruction during an upload.)
- communication from a PCL* station during a download from the host CPU
- execution of an incomplete ladder program resulting from an interrupted download from the host CPU

* PCL is the abbreviation for Allen-Bradley's Peer Communications Link, also known as Data Highway Plus.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

The host CPU should

- upload memory while the LTV is in upload/program or upload/remote program mode in order to verify the transfer. The host CPU cannot verify the transfer if data values change which is likely in upload/run mode, upload/remote run mode, or if a PCL station writes a message instruction to a LTV data file during an upload. (The LTV can be executing a ladder program during an upload if in upload/run or upload/remote run mode.)

- verify the integrity of uploaded or downloaded data, except when uploading while the LTV is executing a ladder program or receiving a message from a PCL station

- if applicable, generate an archive file so the host CPU can later verify that no one has changed the LTV's program.

Verification

The host CPU should verify uploads with a second set of physical read commands, and verify downloads by following physical write commands with physical read commands. The host CPU can always verify a download because the LTV must be in program mode where it freezes data files and inhibits communication from a PCL station. The host CPU can verify only those memory segments that don't change in an upload when the LTV is in upload/run or upload/remote run mode and executing a ladder program, or when a PCL station is communicating with the LTV. You identify verifiable memory segments with "compare" memory segment pointers. Refer to **Set Upload Privilege,** below.

The LTV freezes real-time clock values, the PCL active station list, and remote I/O status in the PC status file during a download or when in upload/program mode,.

Completion

The host CPU issues a restart-after-upload or restart-after-download command at completion. However, the operation can terminate prematurely due to a delay or interruption of power. In either case when the LTV's internal timer times out (at approximately one minute), the LTV's response depends on whether responding to an upload or download.

Upload The LTV automatically reverts to its pre-upload operating mode.

Download The LTV lights the Processor Fault LED on its front panel, and inhibits execution of the incomplete

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

ladder program until it detects the download of a faultless program from the host CPU or programming terminal.

The following commands reset the LTV's internal timer during an upload or download. If the LTV sees none of these commands for one minute, its timer times out.

Identify LTV, Report Status
Physical Read
Physical Write

Command Discrimination

When uploading or downloading, the LTV either inhibits (no) or allows (yes) other commands from the host CPU or from a PCL station.

Command	Upload	Download
Set Processor Mode	no	no
Identify LTV, Report Status	yes	yes
Echo	yes	yes
Physical Read	yes	yes
Physical Write	no	yes*
Write Bit	yes	no
Read Block	yes	no
Write Block	yes	no

* only from the station doing the download

Upload/Download Compatibility

The host CPU can transfer memory (data table and/or ladder program files) between multiple LTVs in the same VMEbus subrack by uploading to itself and then downloading to the target LTV. You must load LTV memory initially with your 1784-T50 programming terminal.

Programs uploaded from the LTV using physical upload commands are NOT compatible with any PLC-5 family processor type except for other 6008-LTV processors. You must use a programming terminal to transfer memory between other types of processors.

The rest of this chapter describes selective commands for up/downloading LTV memory across the VMEbus backplane.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Identify LTV, Report Status (06H 03H)

Use this command to confirm communication with the specified LTV, identify its operating mode, and report other useful upload/download information before initiating an upload or download.

Command Packet

DST	PSN	SRC	PSN	CMD	STS	TNS	FNC
	05		00	06	00		03

Reply Packet

LNH Hi	LNH Lo	DST	PSN	SRC	PSN	CMD	STS	TNS	STATUS (17 bytes)
			05		00	46			

Refer to **Header Bit/Byte Descriptions,** in chapter 7 for descriptions of all bytes except STATUS.

The STATUS field returned in the reply packet indicates the following information.

Byte	Description
1	Operating status of the LTV
	Bits 2-0 000 = program load 010 = run mode 100 = remote program load 101 = remote test 110 = remote run 001, 011, 111 = not used
	Bit 3 0 = no fault 1 = major fault
	Bit 4 0 = not downloading 1 = download mode
	Bit 5 0 = not uploading 1 = upload mode
	Bit 6 0 = not testing edits 1 = testing edits
	Bit 7 0 = no edits in LTV 1 = edits in LTV
2	Processor type
	Bits 0-7 CBH = 6008-LTV
3-6	Processor Memory Size (28K bytes) (low word, low byte first)

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

7	Series and revision of 6008-LTV	
	Bits 4-0	000 = Revision A 001 = Revision B, etc
	Bits 7-5	000 = Series A 001 = Series B, etc
8	Processor station number	
	Bits 5-0	Station number 0-63
9	FDH	Future development
10	01H	Future development
11,12	Number of data files used (highest assigned file number + 1) (low byte first)	
13,14	Number of program files used (highest assigned file number + 1) (low byte first)	
15	Forcing status	
	Bit 0	0 = no forces active 1 = forces active
	Bit 4	0 = no forces present 1 = forces present
	All other bits = 0	
16	00H	Future development
17	RAM invalid	
	Bits 7-0	0 = RAM valid any bit set = invalid RAM

Error Codes

This command does not return error codes.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Set Processor Mode (0FH 3AH)

Use this command to set the LTV's operating mode. Set it to program mode before initiating a download, or return it to run mode after a download. Setting the LTV to program mode before initiating an upload is optional (see Verification, above). Set bits 1 and 0 in flag byte "a" to select the operating mode.

Command Packet

DST	PSN	SRC	PSN	CMD	STS	TNS	FNC	a
	05		00	0F	00		3A	

a - flag byte (explained below)

Reply Packet

DST	PSN	SRC	PSN	CMD	STS	TNS	EXT STS
	05		00	4F			

Bits 0 and 1 of the flag byte determine the operating mode of the LTV.

Mode	Bits	01	00
Program Load program scan idle, I/O scan disabled		0	0
Remote Test program scan enabled, I/O scan disabled		0	1
Remote Run program scan enabled, I/O scan enabled		1	0
No Change to Operating Mode		1	1

Bits 02-07 are not used (set to zero).

Refer to **Header Bit/Byte Descriptions** in chapter 7 for all byte descriptions.

Error Codes

The STS byte contains 00H if no error. When detected, the LTV reports errors in its reply packet as follows:

STS	EXT STS	Description
00H	-	no error
F0H	0BH	Access denied - LTV in upload or download mode

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Set Upload Privilege
(0FH 53H)

Use this command to place the LTV in an uplo
before uploading LTV memory. Follow the suc
completion of this command with successive ph
commands.

During upload, the LTV is in upload/program,
or upload/remote run mode. The host CPU can
static memory segments if the LTV is in uploac
upload/remote run mode, or if LTV memory is a
message commands from a PCL station during
this using compare segments of memory segmer

IMPORTANT: This command returns informat
host CPU to upload or download LTV memory.
pointers to segments of transferrable memory.
you intend to upload to or download from the h
have been originally downloaded to the LTV fr
programming terminal. You cannot download a
program to the LTV over the VMEbus backpla

Command Packet

DST	PSN	SRC	PSN	CMD	STS	TNS	FNC
	05		00	0F	00		53

Reply Packet

DST	PSN	SRC	PSN	CMD	STS	TNS	EXT STS	Memory Segme Pointers (42 t
	05		00	4F				

Memory Segment Pointers

Upload/download Segments

03	Segment 1		Segment 2		Repeats
	Start Pointer	End Pointer	Start Pointer	End Pointer	Segmer

Compare Segments

02	Segment 1		Segment 2	
	Start Pointer	End Pointer	Start Pointer	End Pointer

The three upload/download segments completel
memory in the LTV and include private RAM f
user files (segment 1), force tables (segment 2),
ladder program and data files (segment 3).

The two compare segments define only static ar
memory that do NOT change (unless you make
programming terminal). As options, use these s
implement a program-verify utility (to determi
the LTV program has been modified) or to veri

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

when LTV memory could be changing. All pointers are byte pointers. End pointers point to last byte of a segment.

Refer to **Header Bit/Byte Descriptions** in chapter 5 for descriptions of remaining bytes.

Error Codes

The STS byte contains 00H if no error. When detected, the LTV reports errors in its reply packet as follows:

STS	EXT STS	Description
00H	-	no error
F0H	0BH	Access denied - LTV in upload or download mode

Driver Program Steps

Your driver program must perform the following steps to successfully perform an upload from the LTV.

1) Identify the LTV and its operating mode.

2) Set its operating mode to program or remote program (for verification) or to run or remote run mode.

3) Issue the set-upload-privilege command.

4) Read the data returned in the reply packet to identify the memory segments.

5) Determine upload parameters for each segment.

```
segment_size = segment_end_ptr - segment_start_ptr + 1

number of physical reads @ 240 bytes = segment_size DIV 240

size of last physical read = segment_size MOD 240
```

You can use any packet size up to 244 bytes (122 words). We used 240 bytes in the example.

6) Upload each segment with successive physical-read commands. Store the starting address and size of each segment to use when downloading.

7) Verify the upload with a second set of physical-read commands if the LTV is in upload/program or upload/remote-program mode. Check for identical data.

8) Issue an restart-after-upload command to terminate the upload and return the LTV to its pre-upload mode. If this command returns an error, the host CPU must repeat the entire procedure.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Physical Read
(0FH 17H)

Use this command to upload segments of LTV memory after a successful set-upload privilege command. You can upload up to 244 bytes (122 words) per packet. Words are loaded low byte first. The first byte and the number of bytes read must be an even number.

You upload LTV memory using successive physical-read commands for each of three memory segments. The memory segments are defined by start and end pointers returned by the set-upload-privilege command. The first command starts at the physical address defined by a memory segment pointer. You must increment the physical address in successive commands. You increment the current physical address over the previous physical address by the same number of bytes (equal to the SIZE value) each command until the segment is complete. The packet size of the last command may be less.

Command Packet

DST	PSN	SRC	PSN	CMD	STS	TNS	FNC	a	SIZE	
	05		00	0F	00		17		Lo	Hi

a - The Physical Address is a four-byte field (order of bytes is lowest to highest) where the current packet starts to read (for example, 00 0A 00 00 for physical address A00).

Refer to **Header Bit/Byte Descriptions** in chapter 5 for a description of all bytes except the following.

SIZE - This a 2-byte field (low byte first) that contains the number of bytes to read (up to 244, even number, only) with each physical-read command.

Reply Packet

DST	PSN	SRC	PSN	CMD	STS	TNS	a	DATA (UP TO 244 BYTES)
	05		00	4F				

a - This byte will be the EXT STS extended status byte if there is an error. Otherwise, the LTV omits this byte.

Error Codes

The STS byte contains 00H if no error. When detected, the LTV reports errors in its reply packet as follows:

STS	EXT STS	Description
00H	-	no error
10H	-	Incorrect command format
40H	-	Internal error such as a parity error
F0H	03H	Incorrect address
F0H	07H	Segment exceeds the end of user memory

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

F0H	0AH	Transaction size too large for a packet
F0H	0BH	Access denied
F0H	12H	Invalid packet format

Restart after Upload (0FH 55H)

Use this command at the completion of an upload to return the LTV to its pre-upload operating mode. If the upload was initiated with the LTV in program mode, now your driver program can change the operating mode to run or run/program to resume processor operation.

Command Packet

DST	PSN	SRC	PSN	CMD	STS	TNS	FNC
				0F			55

Reply Packet

DST	PSN	SRC	PSN	CMD	STS	TNS	EXT STS
				4F			

Refer to **Header Bit/Byte Descriptions** in chapter 5 for a description of each byte.

Error Codes

The STS byte contains 00H if no error. When detected, the LTV reports errors in its reply packet as follows:

STS	EXT STS	Description
00H	-	no error
F0H	0BH	Access denied
F0H	0DH	LTV already available

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Set Download Privilege (0FH 50H)

Use this command to place the LTV in download mode before downloading memory. This command clears LTV memory and loads default program files 0 and 1 (ladder), and data files 0, 1, and 2 (I/O and status). Follow this command with successive physical-write commands using memory segment parameters from uploading LTV memory.

Command Packet

DST	PSN	SRC	PSN	CMD	STS	TNS	FNC
	05		00	0F	00		50

Reply Packet

DST	PSN	SRC	PSN	CMD	STS	TNS	EXT STS
	05		00	4F			

Refer to **Header Bit/Byte Descriptions** in chapter 5 for a description of each byte.

Error Codes

The LTV reports errors, if detected, in its reply packet as follows:

STS	EXT STS	Description
00H	-	no error
F0H	0BH	Access denied - LTV is in run mode, memory protected, or being programmed from a programming terminal
F0H	0DH	LTV already available

Driver Program Steps

Your driver program must perform the following steps to successfully perform a download to the LTV.

1) Identify LTV and its operating mode.

2) Set the LTV's operating mode to remote program.

3) Issue the set-download-privilege command.

4) Write data and upload parameters into physical-write command packets for each memory segment.

5) Verify the download by issuing physical-read commands for corresponding physical-write commands and checking for identical data.

6) Issue a reset-after-download command to terminate the download and return the LTV to program mode. If an error is returned, repeat the entire procedure.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Physical Write
(0FH 18H)

Use this command to download LTV memory after a successful set-download-privilege command. You can download up to 120 words (240 bytes) per packet. Words are loaded low byte first. The first byte and the number of bytes written must be an even number.

You download LTV memory using successive physical-write commands for each of three memory segments. The memory segments are defined by start and end pointers returned by the set-upload-privilege command. The first command starts at the physical address defined by a memory segment pointer. You must increment the physical address of successive commands. You increment the current physical address over the previous physical address by the same number of bytes (equal to the SIZE value) each command until the segment is complete. The packet size of the last command may be less.

Command Packet

DST	PSN	SRC	PSN	CMD	STS	TNS	FNC	a	b
	0		0	0F			18		

a - The Physical Address is a four-byte field (order of bytes is lowest to highest) where the current packet starts to write. For example, 00 0A 00 00.

b - You can write up to 120 data words (two bytes per word) per command packet (enter low byte first).

Reply Packet

DST	PSN	SRC	PSN	CMD	STS	TNS	EXT STS
				4F			

Refer to **Header Bit/Byte Descriptions** in chapter 5 for a description of each byte.

Error Codes

The STS byte contains 00H if no error. When detected, the LTV LTV reports errors in its reply packet as follows:

STS	EXT STS	Description
00H	-	no error
10H	-	Incorrect command format
40H	-	Internal error such as a parity error
60H	-	Write operation disallowed
F0H	03H	Incorrect address
F0H	07H	Segment exceeds the end of user memory
F0H	12H	Invalid packet format
F0H	0BH	Access denied

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Restart after Download (0FH 52H)

Use this command to return the LTV from download/program to program mode after downloading memory. Now your driver program can change the LTV's operating mode to run or run/program to resume processor operation.

Command Packet

DST	PSN	SRC	PSN	CMD	STS	TNS	FNC
	05		00	0F	00		52

Reply Packet

DST	PSN	SRC	PSN	CMD	STS	TNS	EXT STS
	05		00	4F			

Refer to **Header Bit/Byte Descriptions** in chapter 5 for a description of each byte.

Error Codes

The LTV reports errors, if detected, in its reply packet as follows:

STS	EXT STS	Description
00H	-	no error
F0H	0BH	Access denied
F0H	0DH	LTV already available

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Introduction

The LTV maintains a status file for programmable controller (PC) operation. The PC status file is Data File 2 which is reserved exclusively for PC status.

The LTV maintains an additional status file for VMEbus operation. This file also lets your LTV's ladder program initiate copy-from-VME and copy-to-VME commands.

We present screen displays, memory maps, and bit/word descriptions of these status files.

PC Status File

The LTV maintains a PC status file where it identifies major, minor, and I/O rack faults, and where it stores information concerning its operation. Your ladder program should examine bits in the PC status file as conditions for taking appropriate action if the LTV detects a fault. The PC status file is described in detail in publication 1785-6.8.2 **PLC-5 Family Processor Manual, Appendix B.**

Screen Display of PC Status File

The processor status screen lets you enter and monitor data in the PC status file (LTV Data File 2). You enter data to enable various processor functions such as fault routines and selectable timed interrupts. You monitor status such as major and minor fault flags and corresponding fault codes. Values are displayed on lines 0-17. The data entry line (20) displays the address corresponding to the cursor's position. Soft function keys F4, F5, F7, and F8 let you:

```
[F4]   display the VME Options Configuration Screen after placing
       the cursor on the VME status file number (line 20).
[F5]   enter the address of another file to display its status
[F7]   display the data monitor screen of PC Data File 3
[F8]   display the data monitor screen of PC Data File 1
```

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

PC Status File

```
   +-------------------------------------------------------------------------+
 0 |I/O STATUS         btx full  00000000   rack fault    00000101            |  0
 1 |I/O CONTROL        reset:    00000000   inhibit:      00000000            |  1
 2 |                                                                         |  2
 3 |ARITHMETIC FLAGS   s:  0      Z:  0     V:  0         C:  0               |  3
 4 |PROCESSOR STATUS   00000000 00000000    USER CTL BITS 00000000 00000000   |  4
 5 |                                                                         |  5
 6 |MINOR FAULT        00000000 00000000                                     |  6
 7 |MAJOR FAULT        00000000 00000000    FAULT CODE         0            |  7
 8 |WHERE FAULTED      prog file:      0    rung:              0            |  8
 9 |FAULT ROUTINE      prog file:      4    watchdog:         70            |  9
10 |SEL TIMED INT      prog file:      6    setpoint:         20            | 10
11 |PROGRAM SCAN [ms]  last:           0    max:               0            | 11
12 |                                                                         | 12
13 |DATE/TIME          1988-10-31 12:34:56  FILE INDEX         0            | 13
14 |ACTIVE NODE LIST                                                         | 14
15 | 0        10       20       30       40       50       60       70      | 15
16 |  01100000 00000000 00000000 00000000 00000000 00000000 00000000 00000000| 16
17 |VME Status file: 54                          Adapter Image File:  1      | 17
18 |                                                                         | 18
19 |Press a key or enter value.                                              | 19
20 |S:7/15 =                                                                 | 20
21 |remote test   forces disabled  decimal data  decimal addr  PLC-5/15      | 21
22 |                      EXTEND  SPECIFY          NEXT  PREV                | 22
23 |                      STATUS  ADDRESS          FILE  FILE                | 23
24 |                        F4      F5              F7    F8                | 24
   +-------------------------------------------------------------------------+
```

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Memory Map of PC Status File (Data File 2)

Values in words 0-14 are for display only and should never be over written. When you enter data into the status file to enable various processor functions, it is stored in words 15-31 marked by an *. A value of zero inhibits the function.

word	
0	Arithmetic Flags
1	General PC Status
2	Configuration Switch Image
3-6	PCL Active Station Table
7	I/O Status Information
8	Current Program Scan (ms)
9	Maximum Program Scan (ms)
10	Minor Fault Bits
11	Major Fault Bits
12	Fault Codes
13	Program File # of Last Fault
14	Rung # of Last Fault
15	File Number for VME Status*
16	Reserved
17	Reserved
18-23	Real Time Clock & Calendar*
24	Index Value for Indexed Addressing*
25	Reserved
26	Function Chart Restart/Continue*
27	I/O Inhibit and Reset*
28	Program Watchdog Setpoint*
29	Fault Routine File #*
30	Selectable Timed Interrupt Setpoint*
31	Selectable Timed Interrupt File #*

IMPORTANT: Do not enter values in words 0-14. They are read, only. Entering values in these words will override their purpose of informing you of PC status or various faults.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Bit/Word Descriptions of PC Status File

Refer to publication 1785-6.8.2, PLC-5 Family Processor Manual, Appendix B, **Status File.**

The following bit description and fault code applies to the LTV and should be added to Appendix B, **Status File,** referenced above.

Word 11 Bit 09

This bit indicates that the LTV incorrectly powered down, typically caused by incorrect timing of the AC Fail signal from the power supply or power monitor module. This signal also sets fault code 90 in word 12.

You clear this bit only in one of the following ways (not by clearing memory):

- Download a new program from the host CPU.
- Toggle the LTV keyswitch from run to program to run.
- Clear the bit from the status file.

Word 12 Fault code 90

Incorrect power down due to the absence or incorrect timing of the AC Fail signal.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

VME Status File

The VME status file is a file that you select. You specify this integer file by entering its file number (9-255) in word 15 (low byte) of the PC status file. You can display this file on two screens. The VME Status Screen displays words 0-5 which are read, only. The VME Copy Configuration Screen displays words 6-23.

Screen Displays

VME Status

The VME Status Screen displays the values stored in words 0-5 of the VME status file. It lets you monitor read-only data such as command block parameters and the image of switch settings from when you configured the LTV. Values are displayed on lines 0-18. The data entry line (20) displays the address corresponding to the cursor's position. Soft function keys F2 - F5 let you:

```
[F2]   display the PROCESSOR STATUS SCREEN
[F4]   display the VME COPY CONFIGURATION SCREEN
[F5]   enter the address of another file to display its status
```

The screen is a read-only display.

```
   +-------------------------------------------------------------------+
 0 |                              VME STATUS                           |  0
 1 |                                                                   |  1
 2 |        SLAVE 0 1K BYTE                        SYSFAIL             |  2
 3 | Start address               0000H    LTV (assert=0,negate=1)    0 |  3
 4 | Address modifier:          2DH 29H   System (assert=0, negate=1) 0 |  4
 5 | (not selected=0, selected=1) 0   0                                |  5
 6 |                                              BUS OPTIONS          |  6
 7 |        SLAVE 1 4K BYTE                  Arbit (not valid=0,valid=1) 0 |  7
 8 | Size (16-bit=0,24-bit=1)       0        Release type (RWD=0,ROR=1)  0 |  8
 9 | Start address              000000H      Release on BCLR (no=0,yes=1)0 |  9
10 | Address modifier group        00H       Data size for CBs          00 | 10
11 | Address modifier BT subgroup* 00H       (D16=00,D08(E0)=01,D08(0)=10  | 11
12 | Address modifier subgroup     00H                                     | 12
13 | *only needed for user-defined AM          INTERRUPT ON COMMAND BLOCK  | 13
14 |                                         Interrupt level (disable=0) 0 | 14
15 |                                         Status I/D              0000H | 15
16 |                                                                       | 16
17 |                                                                       | 17
18 |                                                                       | 18
19 | Press a function key.                                                 | 19
20 | [                                                                     | 20
21 | remote program   no forces   decimal date   decimal addr   PLC-5/VME| 21
22 |       PROC          VME       SPECIFY                               | 22
23 |      STATUS        CONFIG     ADDRESS                               | 23
24 |        F2           F4         F5                                   | 24
   +-------------------------------------------------------------------+
```

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

VME Copy Configuration

The VME Copy Configuration Screen lets you monitor or configure copy command parameters by entering values and setting bits in words 6-23 of the VME Status File using your programming terminal. Also, your ladder program can configure, start, and stop copy commands by writing to words 6-14 for a copy-from-VME or to words 15-23 for a copy-to-VME command. Values are displayed on lines 0-18. The data entry line (20) displays the address at the cursor's position. Soft function keys F2 - F5 let you:

```
[F2]   display the PROCESSOR STATUS SCREEN
[F4]   display the VME STATUS SCREEN
[F5]   enter the address of another file to display its status
```

```
   +-------------------------------------------------------------------------+
 0 |                          VME COPY CONFIGURATION                         | 0
 1 |                                                                         | 1
 2 |         COPY FROM VME                        COPY TO VME                | 2
 3 |                                                                         | 3
 4 |Enable (off=0,on=1)                0    Enable (off=0,on=1)            0 | 4
 5 |X-fer pt (before=0,after=1)        0    X-fer pt (before=0,after=1)    0 | 5
 6 |Scan mode (discontin=0,contin=1)   0    Scan mode (discontin=0,contin=1) 0 | 6
 7 |Error                             00H   Error                         00H| 7
 8 |Source address modifier           00H   Dest address modifier          00H| 8
 9 |VME address                    000000H   VME address                 000000H| 9
10 |Length (words)                     0    Length (words)                 0 | 10
11 |Data size(D16=0,D08(E0)=1,D08(0)=2 0    Data size(D16=0,D08(E0)=1,D08(0)=2 0| 11
12 |Destination DT file                0    Source DT file                 0 | 12
13 |Element                            0    Element                        0 | 13
14 |Interrupt level (disable=0)        0    Interrupt level                0 | 14
15 |Status/ID                       0000H   Status/ID                     0000H| 15
16 |RWD/ROR (RWD=0, ROR=1)             0    RWD/ROR (RWD=0, ROR=1)         0 | 16
17 |ROC (ignore=0, ROC=1)              0    ROC (ignore=0, ROC=1)          0 | 17
18 |                                                                         | 18
19 |Press a key or lenter value.                                             | 19
20 |N55:23 = [                                                               | 20
21 |remote program  no forces     decimal data  decimal addr  PLC-5/VME Addr 76| 21
22 |         PROC            VME    SPECIFY                                  | 22
23 |        STATUS         STATUS   ADDRESS                                  | 23
24 |          F2             F4       F5                                     | 24
   +-------------------------------------------------------------------------+
```

IMPORTANT: The VME Copy Configuration screen contains these errors:

Line	Caption	Incorrect Revision 2.1		Correct Revision 2.2
5	X-fer pt	before = 0	--->	before = 1
		after = 1	--->	after = 0
6	Scan mode	discontin = 0	--->	synch = 1
		contin = 1	--->	contin = 0
16,17	absent	no heading		as shown above

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Memory Map of VME Status File

Values in words 0-5 marked with an * are for display only and should never be overwritten. They store command block parameters (words 2-5) and the LTV's configuration switch settings (word 1).

When data is entered into the VME status file to initiate a copy command, it is stored in words 6-14 for a copy-from-VME command or in words 15-23 for a copy-to-VME command. Zero values inhibit these commands.

word	
0*	SYSFAIL status
1*	Configuration Switch Image
2*	VMEbus/Master/Slave Options
3*	VMEbus Address Modifier
4-5*	VMEbus Address
6	Copy-from-VME Control Byte/Error Code
7	Address Modifier, VME Source
8	VME Source Address, Hi Byte
9	VME Source Address, Lo Byte
10	Length of Copy
11	LTV Destination File Number
12	LTV Destination Element Number
13	Completion Interrupt Level
14	Status/ID for Completion
15	Copy-to-VME Control Byte/Error Code
16	Address Modifier, VME Destination
17	VME Destination Address, Hi Byte
18	VME Destination Address, Lo Byte
19	Length of Copy
20	LTV Source File Number
21	LTV Source Element Number
22	Completion Interrupt Level
23	Status/ID for Completion
24-31	Reserved for Future Use

* Read, only

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Word/Bit Descriptions

Word and bit descriptions of the VME status file follow. Bits are numbered in decimal, 15-00.

IMPORTANT: Data in words 0-5 are displayed on the VME Status Screen of your programming terminal. Numbers in parentheses in the Title column refer to the line in the screen display where you can find this parameter. L or R refer to the left or right side of the screen.

Category	Word	Bit	Title	Description
LED status (read, only)	0	15-02		Reserved for future use (zero)
		01	Sysfail (4R)	0 = Sysfail asserted from VMEbus 1 = Sysfail negated
		00	Sysfail (3R)	0 = Sysfail asserted by LTV 1 = Sysfail negated
Configuration Switch Image (read, only) Also described in chapter 10	1	15-13 and 11-08		Reserved for future use (zero)
		12	Valid Arbit Level (7R)	0 = Arbitration level not valid 1 = Arbitration level valid
		07-02	Slave 0 Address (3L)	LTV sets bits 07-02 to correspond to switches 8-3 in SW4 that you set for base address of Slave 0 memory. Refer to **SW4, Slave 0 Memory Access** in chapter 10.
		01-00	Slave 0 AM (4,5L)	LTV sets bits 01-00 to correspond to switches 2-1 in SW4 that you set for Slave 0 address modifiers 01 00 Address Modifier 0 0 Isolate LTV from VMEbus 0 1 29 Short non-privileged access 1 0 2D Short supervisory access 1 1 29 and 2D
VMEbus Options (read, only) for Command Block	2	15	ROR/RWD (8R)	0 = Release when done (RWD) 1 = Release on request (ROR) RWD means that the LTV immediately releases the VMEbus and after completing the current VMEbus transfer. ROR means that the LTV releases the VMEbus after the transfer is complete and another VMEbus master requests the bus.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Category	Word	Bit	Title	Description
		14	ROC/Ignore (9R)	0 = Ignore bus clear signal 1 = Release bus clear signal Ignore means that the LTV will not release the VMEbus when the bus clear signal is asserted. ROC means that the LTV releases the VMEbus immediately after the current bus cycle upon detecting the bus clear BCLR signal.
		13-12	Data Size (10,11R)	Data size for command blocks 13 12 Format 0 0 16-bit word transfers 0 1 8-bit odd and even byte transfers starting on selected byte 1 0 8-bit odd-byte transfers starting with the (next) odd byte (invalid for CBs)
		11	Slave 1 Addr Size (8L)	0 = 16-bit address for short global memory 1 = 24-bit address for standard global memory
		10-03		Not defined (zero)
		02-00	AM Group Number (10L)	Specifies standard global memory address modifier group (group 7, 5, 3, or 2) Stored in binary, displayed in Hex*
	3	15-08	AM BT Subgroup Number (11L)	Address modifiers for VMEbus block transfer operation. Selected bits determine the address modifiers (ie 10001000 = 3F, 3B)* Only required for user-defined AMs.
		07-00	AM Subgroup Number (12L)	Address modifiers for global memory address. Selected bits determine the address modifiers (ie 10101000 = 3F,3D,3B)*
				* Refer to **Selecting Slave 1 Address Modifiers** in chapter 5.
	4	15-08		Reserved
		07-00	VME Addr (9L)	VME Global Memory Address (hi byte)
	5	15-08	VME Addr (9L)	VME Global Memory Address (mid byte)
		07-00	VME Addr (9L)	VME Global Memory Address (lo byte)

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

IMPORTANT: Data in words 6-23 below are displayed on the VME Copy Configuration Screen. Bits are numbered in decimal. Numbers in parentheses in the Title column refer to the line in the screen display where you can find this parameter. L or R refer to the left or right side of the screen. **NOTE:** Copy command parameters are described in detail in chapter 6.

Category	Word	Bit	Title	Description
Copy-from-VME Control Word	6	15	Enable (4L)	0 = stop copy command 1 = start copy command
		14	Xfer Pt (5L)	0 = LTV executes transfers before I/O scan 1 = LTV executes transfers after I/O scan
		13	Scan Mode (6L)	0 = continuous program scan runs independent of VME transfer 1 = synchronous program scan waits for (is dependent on) VME transfer
		12	Reserved	
		11	ROC/Ignore (17L)	0 = ignore bus clear signal 1 = release on bus clear signal
		10	RWD/ROR (16L)	0 = RWD (release when done) 1 = ROR (release on request)
		09,08	Data Size (11L)	Size of transferred data 09 08 Format 0 0 16-bit word transfers 0 1 8-bit odd and even byte transfers starting on selected byte 1 0 8-bit odd-byte transfers starting with the (next) odd byte
		07-00	Error Code (7L)	Binary code, displayed in hex Refer to **Error Codes for Copy Commands** at the end of this chapter.
Copy-from-VME	7	15-10	Source AM (8L)	Address modifier of VME source file Stored in binary, displayed in Hex
	8	15-08		Not used (zero)
		07-00	VME Addr (9L)	VME source address (high byte) Zero for short 16-bit address
	9	15-08	VME Addr (9L)	VME source address (mid byte) High byte for short 16-bit address
		07-00	VME Addr (9L)	VME source address (low byte) Low byte for short 16-bit address

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Category	Word	Bit	Title	Description
	10	15-00	Length (10L)	The number of words (integer, 0-500) Stored in binary, displayed in decimal
	11	15-00	LTV File (12L)	Destination file number in the LTV (3-999) Stored in binary, displayed in decimal
	12	15-00	Element (13L)	Destination starting element number (0-999) Stored in binary, displayed in decimal
	13	02-00	Interrupt Level (14L)	Binary code for levels 1-7 (displayed in decimal) 000 = interrupt not selected 001 = level 1 010 = level 2 : : 111 = level 7
	14	15-00	Status/ID (15L)	FF00-FFFF = 8-bit vector with values 00-FF XX00-XXFF = 16-bit vector with values XX00-XXFF where XX = any value except FF
Copy-to-VME Control Word	15	15	Enable (4R)	0 = stop copy command 1 = start copy command
		14	Xfer Pt (5R)	0 = LTV executes transfer before I/O scan 1 = LTV executes transfer after I/O scan
		13	Scan Mode (6L)	0 = continuous program scan runs independent of VME transfer 1 = synchronous program scan waits for (is dependent on) VME transfer
		12	Reserved	
		11	ROC/Ignore (17R)	0 = ignore bus clear signal 1 = release on bus clear signal
		10	RWD/ROR (16R)	0 = RWD (release when done) 1 = ROR (release on request)
		09,08	Data Size (11L)	Size of transferred data 09 08 Format 0 0 16-bit word transfers 0 1 8-bit odd and even byte transfers starting on selected byte 1 0 8-bit odd-byte transfers starting with the (next) odd byte
		07-00	Error Code (7L)	Binary code, displayed in hex Refer to **Error Codes for Copy Commands** at the end of this chapter.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Category	Word	Bit	Title	Description
Copy-to-VME	16	05-00	VME AM (12R)	Address modifier of VME destination file Stored in binary, displayed in Hex
	17	15-08		Not used (zero)
		07-00	VME Addr (13R)	VME destination address (high byte) Zero for short 16-bit address
	18	15-08	VME Addr (13R)	VME destination address (mid byte) High byte for short 16-bit address
		07-00	VME Addr (13R)	VME destination address (low byte) Low byte for short 16-bit address
	19	15-00	Length (10R)	The number of words (integer, 0-500) Stored in binary, displayed in decimal
	20	15-00	LTV File (8R)	Source data file number in the LTV (3-999) Stored in binary, displayed in decimal
	21	15-00	Element (9R)	Source starting element number (0-999) Stored in binary, displayed in decimal
	22	02-00	Interrupt Level (14R)	Binary code for levels 1-7 (displayed in decimal) 000 = interrupt not selected 001 = level 1 010 = level 2 : : 111 = level 7
	23	15-00	Status/ID (16R)	FF00-FFFF = 8-bit vector with values 00-FF XX00-XXFF = 16-bit vector with values XX00-XXFF where XX = any value except FF
	24-31			Reserved

Transfer Error Codes for Copy Commands

When the LTV detects an error while attempting to transfer data across the VMEbus backplane, it displays the error code in the VME status file, control word 6, (low byte) for the copy-to-VME command and in word 15 (low byte) for the copy-from-VME command as follows:

07H Invalid copy command data block
09H Past end of data file or bus error on word write
11H Bad address modifier
21H Address error in Slave 0 or Slave 1
41H Selected more than one arbitration level
FDH Length greater than 500
FEH last command-complete interrupt not acknowledged
FFH Interrupt level greater than 7

All other codes indicate a VMEbus transfer error.

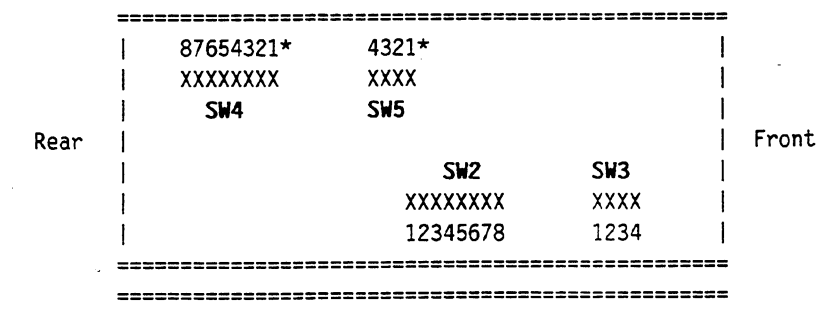
Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Switch Assembly Locations

The LTV has five switch assemblies that you set prior to installation in the VMEbus subrack. Switch assembly SW1 is located on the top of the module, assemblies SW2-SW5 are located on the bottom. (figure 10.1)

Switch Assy & Location		Number of Switches	Function
SW1	Top	8	PCL # and Memory Protect
SW2	Bottom	8	Not Used
SW3	Bottom	4	Terminating Resistor
SW4	Bottom	8	Slave 0 Memory Access
SW5	Bottom	4	Arbitration Level

Figure 10.1
Switch Assemblies on Bottom of Module



* When viewing the module as shown, switch numbers are upside down.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

SW1
8-Switch Assembly
(Top of Module)

PCL Station Number and Memory Protection

This switch assembly determines the station (node) number of the LTV, and whether or not you can download from a programming terminal.

PCL Station Number Select a PCL station number (0-63 decimal, 0-77 octal) by entering the number using switches 1-6. Stations on the PCL use this ID number when communicating with the LTV.

The LTV adds 128 decimal (80H) to this number for the VMEbus station number. You must use 128 + the PCL station number in your host CPU software driver program when communicating with the LTV.

IMPORTANT: 1 = on (closed) 0 = off (open)

Station Number		Switches
Decimal	Octal	1 2 3 4 5 6
0	0	1 1 1 1 1 1
1	1	0 1 1 1 1 1
2	2	1 0 1 1 1 1
3	3	0 0 1 1 1 1
4	4	1 1 0 1 1 1
5	5	0 1 0 1 1 1
6	6	1 0 0 1 1 1
7	7	0 0 0 1 1 1
8	10	1 1 1 0 1 1
9	11	0 1 1 0 1 1
10	12	1 1 0 0 1 1
11	13	0 0 1 0 1 1
12	14	0 1 0 0 1 1
13	15	0 1 0 0 1 1
14	16	1 0 0 0 1 1
15	17	0 0 0 0 1 1
16	20	1 1 1 1 0 1
17	21	0 1 1 1 0 1
18	22	1 0 1 1 0 1
19	23	0 0 1 1 0 1
20	24	1 1 0 1 0 1
21	25	0 1 0 1 0 1
22	26	1 0 0 1 0 1
23	27	0 0 0 1 0 1

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Station Number		Switches
Decimal	Octal	1 2 3 4 5 6
24	30	1 1 1 0 0 1
25	31	0 1 1 0 0 1
26	32	1 0 1 0 0 1
27	33	0 0 1 0 0 1
28	34	1 1 0 0 0 1
29	35	0 1 0 0 0 1
30	36	1 0 0 0 0 1
31	37	0 0 0 0 0 1
32	40	1 1 1 1 1 0
33	41	0 1 1 1 1 0
34	42	1 0 1 1 1 0
35	43	0 0 1 1 1 0
36	44	1 1 0 1 1 0
37	45	0 1 0 1 1 0
38	46	1 0 0 1 1 0
39	47	0 0 0 1 1 0
40	50	1 1 1 0 1 0
41	51	0 1 1 0 1 0
42	52	1 0 1 0 1 0
43	53	0 0 1 0 1 0
44	54	1 1 0 0 1 0
45	55	0 1 0 0 1 0
46	56	1 0 0 0 1 0
47	57	0 0 0 0 1 0
48	60	1 1 1 1 0 0
49	61	0 1 1 1 0 0
50	62	1 0 1 1 0 0
51	63	0 0 1 1 0 0
52	64	1 1 0 1 0 0
53	65	0 1 0 1 0 0
54	66	1 0 0 1 0 0
55	67	0 0 0 1 0 0
56	70	1 1 1 0 0 0
57	71	0 1 1 0 0 0
58	72	1 0 1 0 0 0
59	73	0 0 1 0 0 0
60	74	1 1 0 0 0 0
61	75	0 1 0 0 0 0
62	76	1 0 0 0 0 0
63	77	0 0 0 0 0 0

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Memory Protection Select memory protection to prevent the LTV from responding to a Clear Memory command or a download from a programming terminal.

Function	Switch
Memory Protected	7 on (closed) for protected
Not Used	8 Always off (open)*

* If this switch is on, the LTV cannot communicate with its remote I/O.

SW2
8-Switch Assembly

Not Used

SW3
4-Switch Assembly

Terminating Resistors

You select a terminating resistor only when the LTV terminates either end of a remote I/O or PCL transmission cable.

PCL Terminating Resistor Select this resistor only if the LTV is an end device on the PCL. This is the case when you connect only one PCL cable to the PEER COMM INTF connector on the front panel of the LTV. Do not select this resistor if the LTV is not connected to the PCL or if the LTV is spliced into the PCL cable (two cable connections to this connector).

Remote I/O Terminating Resistor Select this resistor whenever the LTV operates as a programmable controller with one or more remote I/O chassies (normal LTV operation).

Terminating Resistor for	Switch Number
PCL	1 on (closed) for terminated
Remote I/O	2 on (closed) for terminated
Not Used	3 and 4 (set to off)

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

SW4
8-Switch Assembly

Slave 0 Memory Access

This switch assembly determines the base (starting) address and address modifiers of Slave 0 global memory.

Base Address of Slave 0 Memory Select the starting address in 400H increments anywhere within the 0000-FC00H byte limits of VMEbus backplane addressing space. You select the six most significant address lines (A15-A10) for this address. Set switches 8-3 as follows:

IMPORTANT: 0 = on (closed), off = open

	Switches/Address Lines					
Short Base	8	7	6	5	4	3
Address (Hex)	A15	A14	A13	A12	A11	A10
0000	0	0	0	0	0	0
0400	0	0	0	0	0	off
0800	0	0	0	0	off	0
0C00	0	0	0	0	off	off
1000	0	0	0	off	0	0
1400	0	0	0	off	0	off
1800	0	0	0	off	off	0
1C00	0	0	0	off	off	off
2000	0	0	off	0	0	0
2400	0	0	off	0	0	off
2800	0	0	off	0	off	0
2C00	0	0	off	0	off	off
3000	0	0	off	off	0	0
3400	0	0	off	off	0	off
3800	0	0	off	off	off	0
3C00	0	0	off	off	off	off
4000	0	off	0	0	0	0
:			:			
:			:			
7C00	0	off	off	off	off	off
8000	off	0	0	0	0	0
8400	off	0	0	0	0	off
8800	off	0	0	0	off	0
8C00	off	0	0	0	off	off
9000	off	0	0	off	0	0
9400	off	0	0	off	0	off
9800	off	0	0	off	off	0
9C00	off	0	0	off	off	off
A000	off	0	off	0	0	0
A400	off	0	off	0	0	off
A800	off	0	off	0	off	0
AC00	off	0	off	0	off	off
B000	off	0	off	off	0	0
:			:			
:			:			
F000	off	off	off	off	0	0
F400	off	off	off	off	0	off
F800	off	off	off	off	off	0
FC00	off	off	off	off	off	off

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Address Modifiers of Slave 0 Memory The host CPU uses one or both of the following address modifiers when writing to the LTV's Slave 0 global memory to initiate a command. You select the address modifier(s) with switches 2 and 1 as follows:

IMPORTANT: on = closed, off = open

Address Modifier	Switch 2	1
Slave 0 not accessible	on	on
29 short non-privileged access	on	off
2D short supervisory access	off	on
29 and 2D	off	off

Setting both switches on isolates the LTV from any communication on the VMEbus backplane. This is helpful when troubleshooting or debugging your VMEbus system.

SW5
4-Switch Assembly

VMEbus Arbitration Level

VMEbus arbitration prevents simultaneous use of the bus by two or more VMEbus masters. The LTV has four arbitration levels from which you select only one.

IMPORTANT: 0 = off (open), on = closed

Arbitration Level	Switches 4	3	2	1
0	0	0	0	on
1	0	0	on	0
2	0	on	0	0
3	on	0	0	0
inhibit	0	0	0	0

You disable the LTV's VMEbus mastership by turning off all four switches. Then, the LTV cannot process commands to an external VMEbus global memory. This is helpful when troubleshooting your VMEbus system.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Chapter 11 Installation Considerations

Introduction

This chapter has two parts:

- information for installing your LTV, and connecting your remote I/O subsystem with reference to the PLC-5 Family Assembly and Installation Manual
- tips for setting up your VMEbus subsystem

PLC-5 Family Assembly and Installation Manual

The PLC-5 Family Programmable Controller Assembly and Installation Manual, publication 1785-6.6.1, explains how you connect your LTV to remote I/O chassis and prepare for programmable controller operation. Although written for PLC-5 family programmable controllers, most of this publication applies to the LTV. We list the EXCEPTIONS by chapter and section, below.

Chapt	Section	Disregard
2	Processor Module	a) Front panel "ADPT" LED b) Switch Assemblies section
	Memory Modules	Entire section
4	Adapter Mode	Entire section
5	Setting the I/O Chassis Switches	Figure 5.12
	Setting the Processor Module Switches	Entire section Figures 5.15 - 5.21
6	How to Install the Memory Modules	Entire section
	How to Remove the Memory Modules	Entire section
7	Chapter Objectives	Section of the Trouble-shooting Chart labeled When Processor Is in Adapter Mode

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Power Requirement

The LTV draws 4.5 amps at 5 volts DC from the VMEbus backplane. Compute the total current requirements of all cards in the VMEbus subrack to be sure this value does not exceed the backplane or power supply rating.

Position in the VMEbus Subrack

Place the LTV in any slot position in the VMEbus subrack except for the left-most slot reserved for the system controller.

Grounding

Proper grounding is important to proper operation of a programmable controller system. Ground loops and RFI can introduce errors and/or cause faults.

We recommend that you make a good electrical connection from the VMEbus subrack to earth ground, and from the LTV's front panel to the frame of the VMEbus subrack. Follow the grounding instructions in the Assembly and Installation Manual for PLC-5 Family Programmable Controllers, publication 1785-6.6.1, for grounding the I/O subsystem.

Power Monitor Module

The power monitor module detects power failures, and signals the VMEbus system to shut down in an orderly manner before power is lost. Proper initialization when power is re-applied depends on proper shut down. For more information, refer to **AC FAIL Signal**, below.

The LTV requires advanced warning of power down, as described in Rule 5.1 of the VMEbus Specification. In the event of a power down without proper warning, the LTV responds as follows when power is restored:

- detects a bad RAM
- turns on its red PROC light
- indicates the nature of the fault by latching Fault Code 90 - Improper Shut Down in word 12 of the PC status file
- latches a major fault flag (configuration fault) in word 11 bit 09 of the PC status file

When the LTV detects a RAM fault, you must clear LTV memory before you can display the PC Status File screen. The LTV does NOT erase this fault upon receiving a Clear Memory (RAM) command. It remains latched so you can observe the fault from your programming terminal's display of the PC status file, or until the host CPU can

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

read the fault bit (word 11 bit 09) and take appropriate action. Erase the fault by any one of the following:

- re-downloading the ladder program to the LTV in program mode
- toggling the LTV keyswitch from program to run
- clearing the bit from the status file (word 11 bit 09)

Refer to The VMEbus Specification, section 5, for additional information on the power monitor.

Refer to the manufacturer's instructions for electrically and physically connecting the power monitor to the VMEbus subrack and power source.

Setup Tips

We recommend the following tips when setting up your VMEbus subsystem:

- **Module Insertion**
  Be sure that module connectors are fully inserted in the backplane connectors of the VMEbus subrack so that electrical connections are secure. Some subrack backplanes bow when you attempt to insert a module, giving the appearance of proper insertion when in fact electrical connections are faulty.

- **Position of System Controller**
  Place it in slot 1 of the VMEbus subrack. If located elsewhere, any VMEbus modules to its left will not receive bus grant or IACKIN signals.

- **IACKIN to IACKOUT jumpers**
  Jumper all unoccupied slots to pass this signal. Do not jumper any occupied slots including those of the LTV. If neither a module nor a jumper is installed, then slaves further down the VMEbus subrack will not see the IACK cycle from the interrupt handler, and will not have their status/ID read by the interrupt handler.

- **BGXIN to BGXOUT jumpers**
  Jumper all unoccupied slots with all four jumpers (BG0IN to BG0OUT up to BG3IN to BG3OUT). Do not jumper any occupied slots. These four jumpers pass bus arbitration grants on all four levels to the host CPU or VMEbus masters which have requested the bus.

- **Interrupt Handler**
  Use only one interrupt handler per level per VMEbus subrack or unpredictable system operation may result.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

- **Bus Arbitration Level**
  Select one bus arbitration level for each VMEbus master. Be sure that the bus arbiter can support the level(s) that you select. Some system controllers and debuggers support only one arbitration level (usually level 3). Refer to chapter 10, **SW5 VMEbus Arbitration Level.**

- **Bus Release, RWD or ROR**
  The type of VMEbus release that you choose determines when the LTV acting as a VMEbus master gives up the bus after completing a command. If you select Release When Done (RWD), the LTV relinquishes control of the VMEbus immediately after completing the last transfer of a command. If you select Release on Request (ROR), the LTV retains control of the VMEbus after completing the command until another VMEbus master requests the VMEbus.

- **Timing for Master Read-Modify-Write (RMW) Cycle**
  The LTV could lock up if a host CPU requires more than 16us for a RMW cycle. If a longer time period occurs, the LTV could clear the semaphore between the host CPU read and write portions of the RMW cycle rather than after the completed cycle.

- **Bus Timer**
  The absence of an active bus timer can cause the system to lock up if a VMEbus master attempts to access an empty backplane address slot at power up. Some VMEbus masters scan the VMEbus subrack to determine the location(s) of valid memory modules. If this is done without an active bus timer, the bus will be locked up indefinitely while the VMEbus master waits for an acknowledgement from the non-existent memory module.

- **Bus Timer and Block Transfer**
  A bus timer may not accommodate block transfers across the VMEbus backplane. Typically, the bus timer resides in the system controller and monitors the time period of a VMEbus backplane cycle. If the timeout function is triggered from the address strobe instead of the data strobe, then a timeout could occur because the address strobe could be held low for up to 256 successive transfer cycles, a time period that could exceed the bus timer's time out period.

- **Block Transfer**
  If the VMEbus master executes block transfer cycles, be sure that the target slave module also supports them. Some slave modules do not support block transfer cycles.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

- **AC FAIL Signal**
  The LTV requires an AC FAIL signal from the power supply or power monitor for proper shutdown and restoration of operation at subsequent power up. You may have to physically wire this connection from the power monitor or external power supply to the main P1 connector on the VMEbus backplane.

  IMPORTANT: If you do not have a power monitor module or your VMEbus power supply does not provide an AC fail signal, you can still power down the LTV without causing a fault by using the VMEbus reset. Press the VMEbus reset button and keep it pressed while you turn off power to the LTV.

  If you use a power supply with an AC FAIL signal, be sure that the AC FAIL signal can drive the AC FAIL circuit on the VMEbus backplane. Refer to The VMEbus Specification Revision C.1, section 5.1 (Timing), 6.21 (Driving), and 6.22 (Loading).

- **Processor Fault at Power Up**
  The power supply or power monitor must provide an AC FAIL signal during power down for an orderly LTV shutdown. If not, the LTV faults when power is restored. It sets a major fault bit in its status file (data file 2, word 11, bit 09) and sets fault code 90 in word 12.

  90 Absence of AC FAIL signal from power supply or power monitor during power down

- **SYS FAIL Signal**
  The LTV requires 2 seconds to power up. It asserts the SYS FAIL signal until power up is complete, or continues asserting this signal if it detects an internal hardware fault.

  Failure of other VMEbus modules to monitor this signal may cause then to fail if they attempt to access the VMEbus backplane during this time.

- **LTV Station Number**
  In its Module ID Data Block (Slave 0 memory, bytes 0-3FH), the LTV reports the PCL station number (located in bytes 31 and 33H) in ASCII Hex data format. This is in contrast with your programming terminal's Who function that reports the ID station number in octal.

- **Mode Selection with Keyswitch**
  When you change operating modes with the keyswitch, we suggest that you pause in program mode for a second if you have a long ladder diagram program (long program scan).

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

- **Troubleshooting Copy Commands**
  Until your driver program is completely debugged, select the following VMEbus options in the copy command block, word 7:

  Transfer point, bit 14, set to 0 for transfer after remote I/O scan

  Scan mode, bit 13, reset to 0 for a continuous program scan (program scan does not wait for VME transfer)

- **Master LED**
  The Master LED comes on momentarily when the LTV requests the VMEbus. It stays on only if some other device in the VMEbus subrack has tied up the bus probably due to a fault.

- **Address Overlap**
  Ensure that address access spaces of multiple VMEbus modules do not overlap. Overlapping can cause faulty operation including bus timeouts.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Chapter 12 Troubleshooting

Introduction

This chapter has two parts:

- LED troubleshooting tables
- Transfer Error codes and Command Block Error Codes

LED Status

The LTV's status indicators tell you its operating condition for the following situations:

LED Indicator	During Powerup	Normal Operation
POWER	green	green
MASTER	off	off or flashing green [1]
SLAVE	off	off or flashing green [1]
SYS FAIL	red	off
COMM	red	off, green, or flashing green [1]
REM I/O	red	green
ADPT	green	off
BATT	off	off
PROC	red	green
FORCE	yellow	off

[1] may flicker depending on the amount of activity

Easily identified faults are indicated by flashing LEDs as follows:

If LED	Flashes	Then the LTV has detected
COMM	red	a fault on the PCL
REM I/O	red	a fault on the remote I/O link
ADPT	green	an internal hardware fault
PROC	red	an internal processor fault

Refer to the LED troubleshooting chart on the following pages for detailed troubleshooting information.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Troubleshooting with LED Indicators

Use the following table to identify probable causes and take corrective action.

Indicator	Color (status)	Description	Probable Cause	Recommended Action
POWER	green	Power is on		None
MASTER	off	Infrequent or no master cycles	Programming error	Find and correct it.
			Using internal memory	None
	green	LTV requesting mastership	No arbiter in system	Install arbiter and ve levels serviced (0-3).
			VME Arbiter level does not match LTV's level	Select correct LTV arb level to match VME arb
			Faulty arbiter	Replace
			A faulted master is holding the bus	Replace faulted master
			No bus timer in system and a master accessed a wrong memory location.	Install a system bus timer to provide a bus error to master cycle.
				Access valid memory.
SLAVE	off	Infrequent or no access by any master		None
SYS FAIL	red	System Fault	Faulted module in VMEbus subrack	Determine which module is asserting SYS FAIL replace or repair.
COMM	green (blinking)	LTV is receiving or transmitting on peer link		None
	red	Watchdog timeout	Hardware error	Cycle power. Replace LT
	red (dull)		Duplicate station address selected	Correct station address
	off	Peer link quiet	None	
REM I/O	green	Active remote I/O link		None
	red	Fault on remote I/O link	Wiring, adapter module(s)	Check all connections, check adapter module(s)

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Indicator	Color (status)	Description	Probable Cause	Recommended Action
	green/red (blinking)	Partial remote I/O link fault	One or more remote I/O chassis faulted	Check status bits in PC Status File 2 (word 7) to identify faulted chassis number; check wiring, adapter module(s), PS.
	off	No remote I/O selected		None
ADPT	green (blinking)	Hardware fault		Replace LTV Return for service
	off	LTV OK or off		None
BATT	red	Battery low		Replace battery within 24 hours (typical)
	off	Battery is good		None
PROC	green	LTV is in run mode and fully operational		None
	red (blinking)	Major fault	Run-time error	Check major fault bit in PC Status File 2 (word 11) for error definition. Clear fault bit, correct problem, and return to run mode.
	red	Memory fault	User RAM has checksum error	Clear memory and reload program
			Memory-module error	Check backplane switch settings and/or insert correct memory module
	off	LTV is in program or test mode, or is not powered		Check power supply and connections
FORCE	amber	Forces enabled		None
	amber (blinking)	Forces present but not enabled		None
	off	No forces present		None
PROC REM I/O COMM	all red		Internal diagnostics have failed	Power down, reseat LTV in rack, power up. Clear memory, reload program. Replace LTV.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Error Codes

Transfer Error Codes For Copy Commands

When the LTV detects an error while attempting to transfer data of a copy command across the VMEbus backplane, it displays the error code in the VME Status File control word 6 (odd byte) for the Copy-to-VME command and in word 15 (odd byte) for the Copy-from-VME command as follows:

```
07H  Data block in copy command does not exist or offset
     beyond end of data file
09H  Past end of data file or bus error on word write
11H  Bad address modifier
21H  Address error in Slave 0 or Slave 1
41H  Selected more than one arbitration level
FDH  Length greater than 500
FEH  last command-complete interrupt not acknowledged
FFH  Interrupt level greater than 7
```

All other error codes indicate a VMEbus transfer error.

Command Block Error Codes

When the LTV detects an error while processing the command block, it reports the error code in the even and odd bytes of the response word that it writes into the command block. It also reports these error codes in Slave 0 bytes 50H and 51H.

If the response word (word 1 in the command block) returns the value 00FFH, the LTV detected no errors and the LTV completed the command.

If the LTV detects and error, it stops processing the command and reports two error catagories: parameter errors and logic errors.

You must determine where these errors occurred, correct them, and issue the command again. Refer to **Selective Commands** in chapter 6 for a description of the logical sequence your driver program must follow to issue these commands.

Parameter Errors (Reported in CB response word 1 even byte, and Slave 0 byte 50H)

```
01H  Bad address (in Slave 0 or Slave 1 global memory)
02H  Bad A/M in external global memory
03H  Bad address in external global memory
04H  Bad command word (word 0) in command block
05H  Bad packet length (word 10)
06H  Bad LSAP or SRC value
07H  Bad response packet address
08H  Bad VME interrupt level (word 2)
```

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

09H Bad file number (word 11) for copy command block
0AH Copy command block did not write to file
0BH Selected more than one arbitration level
80H VMEbus error

Logic Errors (Reported in CB response word 1 odd byte, and Slave 0 byte 51H) The error occurred at the point in your driver program logic when it commanded the LTV to do the following function.

00H perform an interrupt command
01H wait for completion of interrupt command
02H get command block
03H wait for completion of get command block
04H get command packet
05H wait for completion of get command packet
06H input buffer busy, try again (driver OK)
07H input buffer full, wait for reply (driver OK)
0AH perform new data interrupt
0BH wait for completion of new data interrupt
0CH report bad-result word for interrupt command
0DH wait for completion of report bad-result word, and reset code to 10H
0EH write reply packet into VMEbus global memory
0FH wait for completion of write reply packet
10H write result word
11H wait for completion of write result word
12H perform command complete interrupt
13H wait for completion of command complete interrupt
14H report error in result word (CB word 1 and Slave 0 bytes 50H and 51H)
15H wait for completion of report error
16H wait for completion of report error interrupt

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Appendix A Example Driver Program

Introduction

This appendix presents an example driver program written in C language for setting the processor mode of the LTV. It is representative of the types of host CPU driver programs that you must create for transferring data across the VMEbus backplane.

Example Driver Program to Set Processor Mode

```
/*********************************************************************
 * Synopsis
 *        This function will set the LTV processor mode (Provided the
 *        keyswitch is in REMote position).
 *
 * Calling Sequence
 *        n = modeset (slv0_addr,src_node,vme_addr,vme_am,mode,st_stat)
 *
 * Parameters
 *
 *        UTINY    *slv0_addr        Short VME memory address of LTV
 *        UTINY    src_node          SRC CMD node number
 *        UTINY    *vme_addr         Address of block of VME memory XX bytes long
 *        UTINY    vme_am            Address modifier for block of VME memory
 *        UTINY    mode              Processor mode 0 = PROGRAM  1 = TEST  2= RUN
 *        USHORT   *st_stat          Address to store the CMD response status
 *
 *        Returns  int
 *********************************************************************/


/*  Define variables and structures */

typedef unsigned char       UTINY;  /* 8 bits no sign */
typedef char        BOOL;   /* 8 bits with sign */
typedef unsigned int        USHORT;         /* 16 bits no sign */
typedef int         SSHORT;         /* 16 bits with sign */


/* Timeouts    */

#define SEMA4_TIMEOUT    5000    /* Reset semaphore about 5 sec */
#define CB_TIMEOUT       5000    /* Command block received */
#define RESPONSE_TIMEOUT         20000     /* Command response */


/* Byte Offsets   */

#define SRC_OFFSET          49
#define SEMA4_OFFSET        64
#define EXTSTAT_OFFSET      10
#define STATUS_OFFSET       7
#define COMEM_RESPONSE      2
```

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

```
/* Return Values  */

#define DONE 0
#define CB_NOT_RECEIVED    0x1000
#define SEMA4_NOT_RESET    0x1001
#define CMD_NOT_DONE       0x1002
#define BAD_SOURCE         0x1100
#define BAD_AMOUNT         0x1102
#define BAD_NODE           0x1105
#define CHECK_STSXSTS      0x1106
#define BAD_MODE           0x1107
#define NOT_HEX            0x1108


/* Structure for command packet to set mode  */

struct cmd_pkt {
        UTINY dst;          /* Destination number                    */
        UTINY psn1;         /* Position (value = 05)                 */
        UTINY src;          /* Source number                         */
        UTINY psn2;         /* Position (value = 00)                 */
        UTINY cmd;          /* Command code                          */
        UTINY sts;          /* Status code (value = 00 for command   */
        UTINY trns_h;       /* Transaction Number (high byte)        */
        UTINY trns_l;       /* Transaction Number (low byte)         */
        UTINY fnc;          /* Function Code                         */

        UTINY mode_type;    /* Data - Mode select value              */
);

#define PKT_LEN 10          /* Length in bytes of set mode cmd packet */


/* Structure for Selective Command Block  */

struct cmd_b {
        USHORT   command_word;
        USHORT   response_word;
        USHORT   int_level;          /* for int command complete */
        USHORT   int_vector;
        USHORT   res1;
        USHORT   res2;
        USHORT   res3;
        USHORT   am;                 /* Data address modifier   */
        USHORT   addr_msbyte;        /* VME address: high byte   */
        USHORT   addr_mid_low;       /* VME address: mid and low bytes */
        USHORT   pkt_length;         /* Packet length in bytes   */
        USHORT   resv1;
        USHORT   resv2;
        USHORT   resv3;
        USHORT   resv4;
        USHORT   resv5;
};

#define CB_SIZE 32                   /* Number of bytes in Command Block */
```

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

```
USHORT    modeset (slv0_addr,src_node,vme_addr,vme_am,mode,st_stat)

UTINY     *slv0_addr;
UTINY     src_node;
UTINY     *vme_addr;
UTINY     vme_am;
BOOL      mode;
USHORT    *st_stat;

{
     struct cmd_pkt cm_pkt;
     struct cmd_b cm_blk;
     long temp_addr;
     UTINY *pkt_addr,*cb_addr,*ptr,*tptr,*pkt_buf,*reply_addr;
     SSHORT i;
     UTINY fsrt, bite, node, semaphore, ext_status;
     USHORT word,response_status;


/* Make sure a proper mode is selected        */

     if (mode < 0 || mode > 2)
          return (BAD_MODE);

     cm_pkt.mode_type = mode;


/* Make sure source node is valid.  Bit 7 must be set. ( >128 decimal).  */

     if (!(src_node & 0x80))
          return (BAD_SOURCE);


/* Get LTV node address */

     fsrt = *(slv0_addr + SRC_OFFSET);          /* Get 1st hex ascii digit */
     if (fsrt >= '0' && fsrt <= '9')            /* Convert to binary */
          bite = fsrt - '0';
     else if (fsrt >= 'A' && fsrt <= 'F')
          bite = fsrt - 'A' + 10;
     else
          return (NOT_HEX);
     bite = bite * 16;                          /* Move to high part of byte */

     fsrt = *(slv0_addr + SRC_OFFSET + 2);      /* Get second hex ascii digit */
     if (fsrt >= '0' && fsrt <= '9')            /* Convert to binary */
          node = fsrt - '0';
     else if (fsrt >= 'A' && fsrt <= 'F')
          node = fsrt - 'A' + 10;
     else
          return (NOT_HEX);
     node = bite + node;                        /* Add upper part (bite) to node */

/* Assemble command packet to set LTV mode */

     cm_pkt.dst          = node;
     cm_pkt.psn1         = 5;
     cm_pkt.src          = src_node;
     cm_pkt.psn2         = 0;
     cm_pkt.cmd          = 0x0F;
     cm_pkt.sts          = 0;
     cm_pkt.trns_h       = 0;
     cm_pkt.trns_l       = 0;
     cm_pkt.fnc          = 0x3A;
     cm_pkt.mode_type    = mode;
```

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

```
/* Prepare to send Selective Command Block to LTV */
/* First zero out the command block structure so unused words are set to 0.*/

    ptr = &cm_blk;
    for (i=0; i< CB_SIZE; i++)
        *ptr++ = 0;


/* Set command block address equal to the address given to the function */

    cb_addr = vme_addr;


/*Have the reply packet address follow beneath the command block*/

    pkt_addr = cb_addr + CB_SIZE;


/* Initialize local PCL command block structure */

    cm_blk.command_word   = 0xFFFF;          /* Choose selective command */
    cm_blk.am             = vme_am;          /* Address mod. of cmd block */

    temp_addr             = pkt_addr;
    cm_blk.addr_msbyte    = (temp_addr >> 16);    /* Addr bits 16-23*/
    cm_blk.addr_mid_low   = (temp_addr & 0x0000FFFF); /*Addr bits 0-15*/
    cm_blk.pkt_length     = PKT_LEN;         /* No. of bytes of CMD packet */


/* Move the information in command block structure into command block buffer.
Do this in a byte manner (not word) so any address boundary concerns of the
68000 can be avoided. */

    ptr  = &cm_blk;          /* local pointer to command block structure*/
    tptr = cb_addr;          /* local pointer to buffer to store command block */

    for (i=0; i< CB_SIZE ; i++)
        *tptr++ = *ptr++;


/* Now move packet into VME memory. Also do in a byte manner */

    pkt_buf = &cm_pkt;       /* Local pointer to packet */

    for (i=0; i < PKT_LEN; i++)
        *pkt_addr++ = *pkt_buf++;


/* Before sending command block pointer to LTV, see if the LTV's semaphore is
set. If it is not, then set it to declare ownership of the command block pointer area.
If it never resets for a given amount of time return an error code. Use a true test-
and-set if there are multiple masters in the VME system. */

    for (i=SEMA4_TIMEOUT; i; i--) {
        semaphore = *(slv0_addr+SEMA4_OFFSET);  /* Read semaphore byte */
        semaphore = semaphore & 0x80;           /* Check bit 7        */
        if (semaphore != 0x80)                  /* If it isn't set then */
                break;                          /* can set it         */
    }
    if (!i)
        return (SEMA4_NOT_RESET);

    *(slv0_addr+SEMA4_OFFSET) = 0x80;           /* Set semaphore      */
```

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

```
/* Now place command block buffer address and its address modifier into VME
   memory.  Write the modifier last - when the LTV sees a valid modifier
   it will begin processing. */

      temp_addr = (long) cb_addr;

      bite = ((temp_addr & 0x00FF0000) >> 16);          /* Addr bits 16-23 */
      *(slv0_addr + SEMA4_OFFSET+3) = bite;

      bite = ((temp_addr & 0x0000FF00) >> 8);           /* Addr bits 8-15 */
      *(slv0_addr + SEMA4_OFFSET+4) = bite;

      bite = (temp_addr  & 0x000000FF);                 /* Addr bits 0-7 */
      *(slv0_addr + SEMA4_OFFSET+5) = bite;

      *(slv0_addr + SEMA4_OFFSET+1) = vme_am;           /* AM of command block */



/*  Poll the address modifier until it is set to FF by the LTV */

      for (i=CB_TIMEOUT; i; i--) {
       bite = *(slv0_addr+SEMA4_OFFSET+1);


/* If address modifier set to FF by LTV the command block has been received */

      if (bite == 0xFF)
          break;
   }
   if (!i)
      return (CB_NOT_RECEIVED);


/* Command block has been received by LTV - now wait for the LTV's response
packet or timeout. */


      for (i=RESPONSE_TIMEOUT; i; i--) {
          word = *(cb_addr+COMEM_RESPONSE);          /* Get response word */
          word  <<= 8;                               /* from the command block */
          word  |= *(cb_addr+COMEM_RESPONSE+1);
          if (word != 0x0000)                        /* If its not 0 then  */
                    if(word == 0x00FF)               /* Check if 00FF...Done */
                              break;                 /* If not 00FF return error*/
      else
      return (word);
      }
      if (!i)
          return (CMD_NOT_DONE);
```

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

```
/* Reply packet is placed after the command packet in VME memory by the
   LTV.  Determine the response status */

        ext_status = 0;
        reply_addr  = vme_addr + CB_SIZE + PKT_LEN;
        response_status = (USHORT) *(reply_addr + STATUS_OFFSET);
        if (response_status != 0) {

/* If there is problem, return response status and extended status code */

            ext_status = *(reply_addr + EXTSTAT_OFFSET);
            *st_stat = ((response_status << 8) | ext_status);
            return (CHECK_ST_STAT);
        }

        return(DONE);
)
```

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Scan Timing

This appendix applies to copy-to-VME and copy-from-VME commands. We present this information to help you understand the timing relationships between transfers across the VMEbus backplane and LTV program scans.

The LTV executes three scans simultaneously: remote I/O scan, program scan, and VME transfers (see figure, below).

Remote I/O Scan

The LTV executes the remote I/O scan asychronously to the program scan, but updates remote I/O buffers (LTV I/O image files) synchronously with the program scan. The remote I/O scan time is 7ms per rack number and increases directly with the number of assigned I/O rack numbers in the remote I/O subsystem. A 4-rack system requires 28ms, not including remote I/O buffer update.

Program Scan

The program scan consists of three serial parts: the ladder program scan, buffer update, and housekeeping. The ladder program scan is 8ms per 1K words of program memory. Buffer update is the time required to update remote I/O buffers (up to four) and the VME buffer. Buffer update time is 100us + 5us per word. A 64-word buffer requires 820us (420us for input, 420us for output). Housekeeping takes from 1ms to 3ms.

VME Transfer

The LTV transfers data over the VMEbus backplane typically at the rate of 1us per word in each direction. This is the time the LTV takes to load or empty a VME input or output buffer. A 64-word buffer requires 64us.

When you select synchronous mode as an LTV program scan option and the VMEbus transfer time is LONGER than one LTV program scan, the LTV holds the program scan until the VMEbus transfer is complete. The hold time depends on the amount of VMEbus traffic the LTV must wait for, which may depend on the arbitration level that you selected for the LTV.

If you select continuous mode, the LTV does not hold the program scan, but transfers the entire buffer regardless of the mix of new and old data.

Refer to **Select Scan Mode** (Bit 13) in chapter 6 for more information on synchronous and continuous modes.

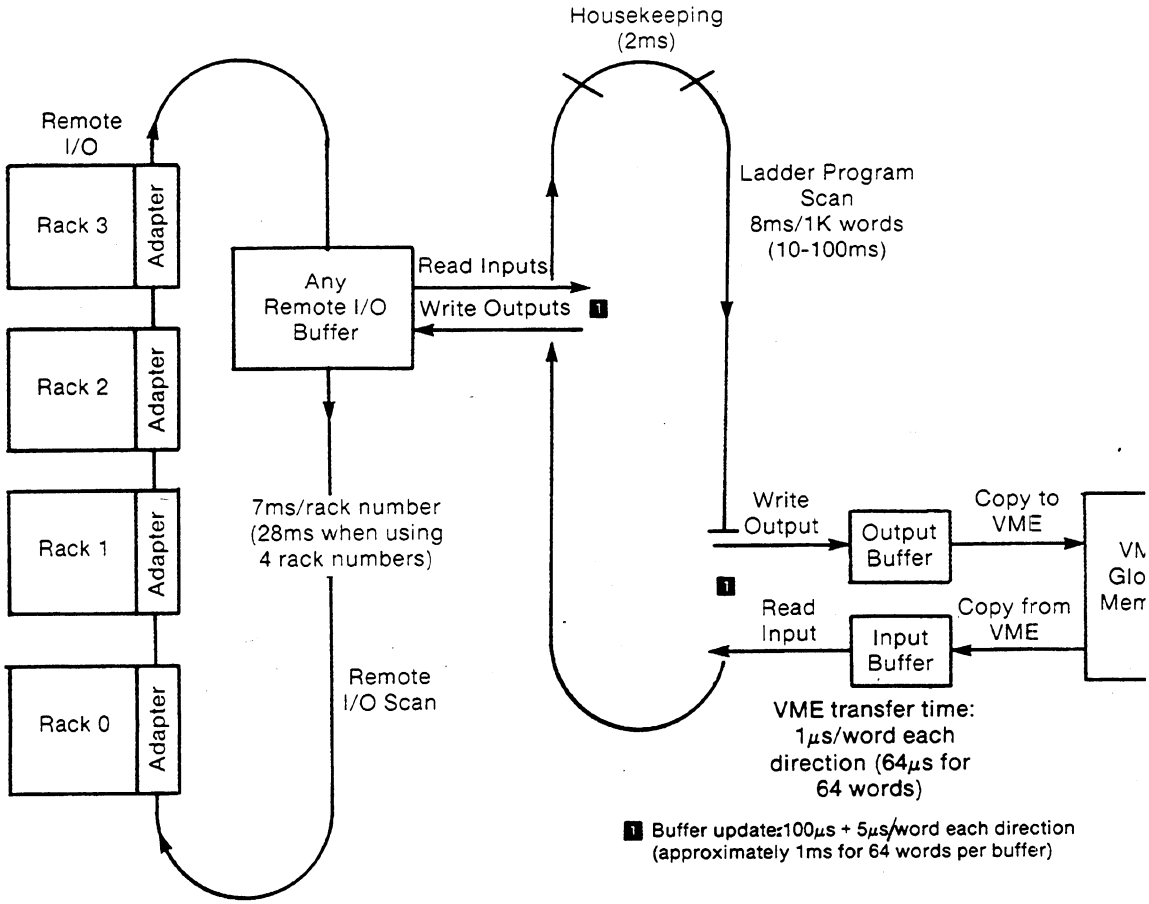
Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Timing Considerations for Copy Commands

The time required to copy data from VME memory into the LTV, process the data in the LTV, and copy it back to VME memory is no less than 2 and no more than 3 LTV program scans. (LTV program scan = ladder program scan + buffer update + housekeeping.) This assumes that delays waiting for VMEbus mastership are small compared with LTV scan times.

The time required to copy data from a remote input to VME memory, or the time required to copy data from VME memory to a remote output is equal to the remote I/O scan time (7ms/rack) plus not less than 1 but not more than 2 LTV program scans.

LTV Scan Times



The diagram shows the VME transfer point before the I/O scan.

The LTV executes remote I/O scans and VME transfers at the same time it executes the LTV program scan.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

A
AC fail signal 11-5
address modifiers for
  command block 5-5
  data address 6-7
  packet address 6-13
  slave 0 5-13, 10-6
  slave 1 5-6, 5-8, 5-10
arbitration level 10-6, 11-4

B
bit/byte (bit/word) descriptions
  copy command block 6-2
  selective command block 6-12
  packet headers 7-3
  PC status file 9-4
  VME status file 9-8
  Slave 0 memory 5-5
block transfer 2-5, 11-4
bus timer 11-4

C
command block for
  copy commands 6-2
    bit/word descriptions 6-2
  selective commands 6-11
    bit/word descriptions 6-12
command block pointer 5-5
command sequence for
  copy commands 4-5, 6-8
  selective commands 6-13
commands, See
  copy commands, below
  selective commands, below
configuring for
  copy commands 3-9, 9-6
  external memory (VMEbus options) 5-2, 5-6, 5-9
  Slave 0 5-1, 5-8, 5-12
  Slave 1 5-1, 5-6, 5-8
copy commands: copy-to/from-VME 6-1
  command block 6-2
    bit/word descriptions 6-2
  command sequence 6-8
  control bits 6-3
  driver programs to initiate and
    change copy commands 4-6
  error codes 6-15, 9-12, 12-4
  ladder programs to
    initiate copy commands 4-1
    change command parameters 4-3
  tutorial for copy commands 3-3

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

D
data integrity 6-3, 6-8
data size, VMEbus master cycle 5-6, 5-9
data types 7-15
definitions used in manual 3-1
down/upload considerations
  command discrimination 8-3
  compatibility 8-3
  completion 8-2
  criteria 8-1
  verification 8-2

E
electrostatic damage 2-3
error codes for
  command blocks 6-15, 12-4
  copy command transfers 6-10, 12-4
external global memory 5-2
  configuring 5-2, 5-6, 5-9

G
global memory 5-1
  external 5-2
  Slave 0 See Slave 0, below
  Slave 1 See Slave 1, below
grounding 11-2

H
handshaking, host CPU and LTV 5-5

I
ID data block for LTV module 5-4
installation
  reference to PLC-5 A & I Manual 11-1
interface, VMEbus and 6008-LTV 2-7
interrupt handler 11-3
interrupts for
  copy commands 6-2, 6-3, 6-7
  selective commands 6-12
I/O subsystem 2-5

J
jumpers in VMEbus subrack 11-3

L
LED status 2-2, 12-1, 12-2
LTV
  faceplate 2-2
  features 2-1
  operation 2-5, 2-7
  system description 2-3
  VMEbus compliance capabilities 2-8

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

M
mechanical package 2-2
memory maps for
copy command block 6-2
copy command sequence 6-9
selective command block 6-12
selective command sequence 6-14
Slave 0 5-3
PC status file 9-3
VME status file 9-7
memory protection 10-4

O
operation as programmable controller 2-3
block transfer 2-5
I/O subsystem 2-5
parity checking 2-6
peer communications link 2-6

P
packet format 7-2
header bit/byte descriptions 7-3
parity checking 2-6
peer communications link 2-6
PCL station number 5-4, 10-2
position in VMEbus subrack 11-2
power monitor module 11-1
power requirements 11-2
processor fault at power up 11-5
programming examples
ladder logic for copy commands 4-1
software driver for copy commands 4-5
software driver to change LTV's processor mode A-1
publications, related 1-1

R
release options for VMEbus
ROR, RWD, ROC 5-6, 5-9, 6-6, 11-4
response word for
copy commands 6-2
selective commands 6-12

S
scan mode
synchronous, asynchronous 6-5
screen displays for
PC status file 9-2
VME copy configuration 3-10, 9-6
VME status file 9-5

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

selective commands 6-11
  command block 6-11
    Bit/Word descriptions 6-12
  command descriptions
    echo 7-6
    identify LTV, report status 7-7, 8-4
    write bit 7-9
    write block 7-11
    read block 7-13
    set processor mode 8-6
    set upload privilege 8-7
    physical read 8-9
    restart after upload 8-10
    set download privilege 8-11
    physical write 8-12
    restart after download 8-13
  command sequence 6-13
  driver program example A-1
  packet format 7-2
    header bit/byte descriptions 7-3
semaphore bit 5-5
Slave 0 memory 5-1
  address modifiers 5-13
  bit/byte descriptions 5-5
  handshaking 5-5
  memory map 5-3
  module ID data block 5-4
  starting address 5-12
  selecting parameters for Slave 1 5-6, 5-8
  switch settings 10-5, 10-6
Slave 1 memory 5-1
  address modifiers 5-8, 5-10
  selecting parameters 5-6, 5-8
  starting address 5-7
starting address
  command block 5-5
  command packet 6-13
  data for copy commands 6-7
  reply packet 6-14
  Slave 0 5-12, 10-5
  Slave 1 5-7, 5-8
status files
  PC status file 9-1
    bit/word descriptions 9-4
    memory map 9-3
    screen display 9-2
  VMEbus status file 9-5
    bit/word descriptions 9-8
    memory map 9-7
    screen display 9-5
    VME copy configuration screen 9-6

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

switch settings 10-1
arbitration level 10-6
memory protection 10-4
PCL station # 10-2
slave 0 address modifiers 10-6
slave 0 memory access 10-5
terminating resistors 10-4
tutorial 3-5
system description 2-3

T
terminating resistor 10-4
timing considerations for
copy commands B-1
transfer point 6-4
transferring data (copy commands) 6-10
troubleshooting 12-1
tutorial for copy commands 3-1
connecting equipment 3-6
defining the transfer 3-7
entering data 3-8
initiating the commands 3-13
overview of tutorial 3-3
setting switches 3-5

V
VMEbus compliance capabilities 2-8
VMEbus mastership options 5-6
configuring 5-9

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

LTV Configuration Worksheet

Parameter Description	Range	Memory Location	Info Page	Your Entry	Driver Mnemonic
1) LTV Station (Node) Number					
PCL #	0-63	SW1	10-2	_____	_____
VME # (add 128 to PCL#)	128-193	SW1	10-2	_____	_____
2) Slave 0					
Base address	0-FFFFH	SW4(8-3)	10-5	_____	_____
Base addr mod	29H, 2DH	SW4(2,1)	10-6	_____	_____
Arbitration level	0,1,2 or 3	SW5	10-6	_____	_____
3) Slave 1					
Use these parameters if using Slave 1 memory in place of external memory module. Your driver program must download these parameters to Slave 0.					
Base address	[1]	4D-4FH	5-7	_____	_____
Base addr size	16*, 24 bit	48H,B03	5-6	_____	_____
Base addr mod group	7,5,3 or 2	49H,B02-00	5-6	_____	_____
Base addr mod subgroup	bit pattern	4BH	5-7	_____	_____
	* = default		5-12		
4) VMEbus Mastership Options					
Use these parameters if using an external memory module in place of Slave 1. Your driver program must download these parameters to Slave 0.					
LTV bus release type	RWD*, ROR	48H,B07	5-6	_____	_____
LTV release on clear	ignore*,ROC	48H,B06	5-6	_____	_____
LTV data size	16-bit*,E08	48H,B05-04	5-6	_____	_____
	* = default		5-12		

---

[1] The lowest starting (base) address for Slave 1 is at Slave 0 base + 400H when you want to place Slave 1 contiguous to Slave 0. Otherwise, start Slave 1 on 100H (256 decimal) byte boundaries at an address above the ending address of Slave 0.

The highest starting (base) address is FFE000H for this 4K byte global memory.

Review **Definitions** in chapter 3 if necessary.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Selective Command Worksheet

Command Block, Command and Reply Packets (Refer to chapter 6)

Command Block starting addr					
Command Block starting AM					
Command Packet starting addr					
Reply Packet starting addr					

Packet Headers (Refer to chapters 7 and 8)

Command #	DST				
Function #	FNC				
Destination	DST				
Source	SRC				
LTV address	LTV ADDR				
Offset	OFFSET				
Transaction	TOTAL TRANS				
Size	SIZE				
Other					

Data Files (Refer to chapter 7)

Write Buffer address					
File type					
File number					
Elem number					
Length					
Read Buffer address					
File type					
File number					
Elem number					
Length					
Other					

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

ALLEN-BRADLEY
A ROCKWELL INTERNATIONAL COMPANY

As a subsidiary of Rockwell International, one of the world's largest technology companies — Allen-Bradley meets today's challenges of industrial automation with over 85 years of practical plant-floor experience. More than 13,000 employees throughout the world design, manufacture and apply a wide range of control and automation products and supporting services to help our customers continuously improve quality, productivity and time to market. These products and services not only control individual machines but integrate the manufacturing process, while providing access to vital plant floor data that can be used to support decision-making throughout the enterprise.

With offices in major cities worldwide.

WORLD HEADQUARTERS
1201 South Second Street
Milwaukee, WI 53204 USA
Tel: (414)382-2000
Telex: 43 11 016
FAX: (414)382-4444

EUROPE/MIDDLE EAST/AFRICA HEADQUARTERS
Allen-Bradley Europe B.V.
Amsterdamseweg 15
1422 AC Uithoorn
The Netherlands
Tel: (31)2975/43500
Telex: (844) 18042
FAX: (31)2975/60222

ASIA/PACIFIC HEADQUARTERS
Allen-Bradley (Hong Kong) Limited
2901 Great Eagle Center
23 Harbour Road
G.P.O. Box 9797
Wanchai, Hong Kong
Tel: (852)/573-9391
Telex: (780) 64347
FAX: (852)/834-5162

CANADA HEADQUARTERS
Allen-Bradley Canada Limited
135 Dundas Street
Cambridge, Ontario N1R 5X1
Canada
Tel: (519)623-1810
Telex: (069) 59317
FAX: (519)623-8930

LATIN AMERICA HEADQUARTERS
1201 South Second Street
Milwaukee, WI 53204 USA
Tel: (414)382-2000
Telex: 43 11 016
FAX: (414)382-2400

Publication 6008-6.5.10 — March 1988

PN 955102-19
Printed in USA

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Artisan Technology Group is an independent supplier of quality pre-owned equipment

Gold-standard solutions

Extend the life of your critical industrial, commercial, and military systems with our superior service and support.

We buy equipment

Planning to upgrade your current equipment? Have surplus equipment taking up shelf space? We'll give it a new home.

Learn more!

Visit us at **artisantg.com** for more info on price quotes, drivers, technical specifications, manuals, and documentation.

Artisan Scientific Corporation dba Artisan Technology Group is not an affiliate, representative, or authorized distributor for any manufacturer listed herein.

We're here to make your life easier. How can we help you today?

(217) 352-9330 | sales@artisantg.com | artisantg.com

ARTISAN® TECHNOLOGY GROUP