

**VME Based Ethernet Access
Functional Specification
ER02-104018-001**

1 Introduction

1.1 Purpose

This document provides a functional overview of the hardware and software interface to provide VME Based 802.3 Ethernet Access to Avtron PDC-6 drive systems. The intent of the proposed design is to provide a modern 802.3 Ethernet interface, as well as an interface that will allow migration from legacy 802.4 LAN systems with minimal PLC ladder logic or data structure changes. The system will also provide a communications gateway for an optional Performance View system.

1.2 Scope

This specification presents the functionality of the VME-SBC (VME based Single Board Computer) hardware platform and its associated interfaces to the Avtron Manufacturing Inc. (A.M.I.) 802.3 Ethernet and PLC command structure. This documents intended audience includes Engineers, Technical Writers, Marketing, Sales, and Customers (at the discretion of Marketing and Sales) who are already familiar with the operation of the 802.3 Ethernet and/or the 802.4 LAN Communications product(s).

1.3 Associated Documents

Eng. Report 98-83208-001, MVME126 Enhanced Data Structures,

Avtron Mfg., Inc., 4/16/98

Eng. Report 97-72754-001, MVME126 Enhancement,

Avtron Mfg., Inc., 6/26/97

1.4 Definitions

A16	--	16 bit VME address space.
A24	--	24 bit VME address space.
Real-time Data	--	Drive Setpoint and Feedback data.
UINT8	--	Unsigned 8 Bit Integer.
UINT16	--	Unsigned 16 Bit Integer.
UINT32	--	Unsigned 32 Bit Integer.
FLOAT	--	IEEE 754 Single Precision Floating Point Number (32 bit)
Endian	--	Refers to the organization and addressing of RAM.
Offset	--	Lower 16 bits of A24 VME Address space.
Base Address	--	Upper 8 bits of A24 VME Address space. See Section 8 for further information regarding VME-SBC Base Address determination.

1.5 Document Revision History

27-Dec-2002,	Steve Kovacik,	Original Release
19-May-2003,	Steve Kovacik,	Post-Development Release

2 Overview

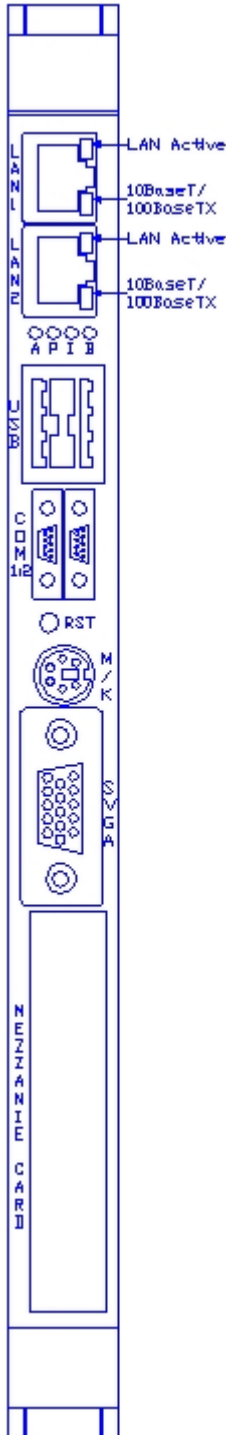


Figure 1

The VME Based Ethernet Access Specification describes a mechanism by which Avtron VME Based PDC-6 systems with VME PLC control are provided with an 802.3 Ethernet communications medium. Utilizing the VME base features in conjunction with the available 802.3 communications medium, the following objectives are recognized through this specification:

- Provide an 802.3 communications path from VME PLCs to Avtron ADDvantage32 drives (Real-time Data Buffer).
- Provide simple integration or upgrade of legacy 802.4 LAN based PDC-6 systems to 802.3 (MVME-162 Interface).
- Provide data access to a Performance View system from the PLC and ADDvantage32 drives on 802.4 LAN (Performance View Interface).

The intended hardware platform is an Intel Pentium® processor-based single board computer (SBC) with an integrated PCI-to-VME interface, **Figure-1**. The VME-SBC will employ a custom Linux operating system. The VME-SBC will have, at a minimum, the following features:

- Ethernet Controller port(s)
- Serial Communication port(s)
- SVGA and Keyboard connection
- Compact Flash RAM support
- IDE Connectivity

The software programs will reside on the VME-SBC's non-volatile Compact Flash RAM. The Linux O.S. and application specific software will auto-boot on VME-SBC power-up. The application software will accept user specific configuration and control data through the VME PLC interface as described within the "Functional Details" section of this specification.

3 Functional Details

3.1 PLC interface

The interface scheme shown in **Tables 1 to 5** in the following section defines the interfaces to the services provided by the VME-SBC to the PLC. The services to the PLC include a Peer Primitive buffer for discrete data read and/or writes to the ADDvantage-32 drives and a Real-Time Data buffer for continuous data read and writes to/from the ADDvantage-32 drives.

NOTE: All buffer layouts provided in the sections which follow are organized in BIG ENDIAN memory organization (see **0x200a – Endian** below for memory organization details).

3.1.1 VME-SBC System Control Buffer

This section contains values used to control the operation of the VME-SBC and its communications to the drive.

Offset	Description	Data Size	Notes
0x2000	Diagnostic Status	UINT8	VME-SBC Board Status
0x2001	Board Command	UINT8	
0x2002	Mode Status	UINT8	
0x2003	802.3 Status	UINT8	
0x2004	MAC Address (optional MVME-162 Interface)	6 * UINT8	Read Only. Refer to Avtron MVME-162 Functional Specification for specific information.
0x200a	Endian	UINT16	Default: little endian.
0x200c	Fatal Code	UINT16	Indicates cause of FATAL Error
0x200e	Non-Fatal Code	UINT16	Indicates cause of NON-FATAL Error
0x2010	Restart Attempt	UINT16	
0x2012	Restart Failures	UINT16	
0x2014	Reserved	UINT16	
0x2016	eth0 I.P. Address	4 * UINT8	Ethernet I.P Address of interface eth0
0x201a	eth0 Subnet Mask	4 * UINT8	Ethernet I.P Subnet Mask of interface eth0
0x201e	Reserved	2 * UINT16	
0x2022	eth1 I.P. Address	4 * UINT8	Ethernet I.P Address of interface eth1
0x2026	eth1 Subnet Mask	4 * UINT8	Ethernet I.P Subnet Mask of interface eth1
0x202a 0x23FF	Reserved	491 * UINT16	

Table 1 – VME-SBC / PLC Interface – System Control

0x2000 - Diagnostic status – read only

0 = Board Power Up / Reset in Progress

1 = Board Power Up / Reset Complete

0x2001 - Board Command – read / write

0 = Do Nothing

1 = Perform System Reset

2 = Stop, Halt Ethernet Communication

3 = Start, Begin Ethernet Communication

4 = Auto Initialize (see section **VME-SBC AUTO INITIALIZE SEQUENCE** later in this document)

5 = Perform Buffer Reset (see section **DATA TABLE RESET COMMAND** later in this document)

0x2002 - Mode Status – read only

0 = Not Ready for Transactions

1 = Ready for Transactions

2 = Fatal Error – Board Inoperable

3 = Non-Fatal Error – Board Operable

4 = MVME-162 Board / Interface Ready

0x2003 – 802.4 Status – read only

0x00 = Network Interface Available

0x01 = Network Interface Not Ready

0x02 = Network Stop Commanded

0x03 = Network Interface Error

0x10 = LAN1 (eth0) Has Valid I.P. Address

0x20 = LAN2 (eth1) Has Valid I.P. Address

0x200a – Endian – read / write

0x0000 – The PLC's memory organization is Little Endian

0x1111 – The PLC's memory organization is Big Endian

Where Big Endian / Little Endian memory organization is defined as follows:

Little Endian -- Also known as "Intel Memory Format". Memory is organized on a byte-addressable basis where lower addresses correspond to "little" order bytes. The address, N, of a long word data item (UINT32, FLOAT) corresponds to the address of the **least** significant byte of the lowest-order 16 bit word. The lower-order 16 bit word is located at address N leaving the least-significant byte at address N+1.

Bits	31	23	15	7	0
Long Word 0x00000000					
Word 0x00000002			Word 0x00000000		
Byte 0x00000003		Byte 0x00000002		Byte 0x00000001	
Long Word 0x00000001					
Word 0x00000006			Word 0x00000004		
Byte 0x00000007		Byte 0x00000006		Byte 0x00000005	

Big Endian -- Also known as “Motorola Memory Format”. Memory is organized on a byte-addressable basis where lower addresses correspond to “big” order bytes. The address, N, of a long word data item (UINT32, FLOAT) corresponds to the address of the **most** significant byte of the highest-order 16 bit word. The lower-order 16 bit word is located at address N+2 leaving the least-significant byte at address N+3.

Bits	31	23	15	7	0
Long Word 0x00000000					
Word 0x00000000			Word 0x00000002		
Byte 0x00000000		Byte 0x00000001		Byte 0x00000002	
Long Word 0x00000001					
Word 0x00000004			Word 0x00000006		
Byte 0x00000004		Byte 0x00000005		Byte 0x00000006	

0x200C – Fatal Code – read only

Indicates cause of FATAL ERROR (Mode Status, 0x2002) as follows:

Low Byte:

- 0x01 = VME Buss Interface Initialization Failure
- 0x02 = VME Window MAP Failure
- 0x04 = VME Window Endian Failure
- 0x08 = VME Window MAP Failure

High Byte:

- # = Window # for which last failure occurred.
- 0x01 = PLC Interface
- 0x02 = MVME-162 Interface

0x200E – Non-Fatal Code – read only

Indicates cause of NON-FATAL ERROR (Mode Status, 0x2002) as follows:

Low Byte:

- 0x01 = VME Window Data Transfer Error
- 0x02 = LAN1 (eth0) Configuration Error
- 0x04 = LAN2 (eth1) Configuration Error
- 0x08 = Real-Time Data Ethernet Socket Creation Error
- 0x10 = Peer-Primitive Ethernet Socket Creation Error
- 0x20 = Performance View Ethernet Socket Creation Error

High Byte:
= Window # for which last Data Transfer Error occurred.

0x2010 – Restart Attempts – read only

0x2012 – Restart Failures – read only

0x2016 – eth0 I.P. Address – read/write

Ethernet I.P. Address assigned to the VME-SBC's on-board Ethernet port labeled LAN1. The I.P. Address is reflected into the four contiguous registers, starting at 0x2016. The Ethernet I.P. Address is Network Byte Ordered which has the following format:

eth0 I.P. Address = 0x2016. 0x2017. 0x2018. 0x2019

0x201a – eth0 I.P. Subnet Mask – read/write

Ethernet I.P. Subnet Mask assigned to the VME-SBC's on-board Ethernet port labeled LAN1. The I.P. Subnet Mask is reflected into the four contiguous registers, starting at 0x200a. The Ethernet I.P. Subnet Mask is Network Byte Ordered which has the following format:

eth0 I.P. Subnet Mask = 0x201a . 0x201b . 0x201c . 0x201d

0x2016 – eth1 I.P. Address – read/write

Ethernet I.P. Address assigned to the VME-SBC's on-board Ethernet port labeled LAN2. The I.P. Address is reflected into the four contiguous registers, starting at 0x2022. The Ethernet I.P. Address is Network Byte Ordered which has the following format:

Eth1 I.P. Address = 0x2022 . 0x2023 . 0x2024 . 0x2025

0x201a – eth1 I.P. Subnet Mask – read/write

Ethernet I.P. Subnet Mask assigned to the VME-SBC's on-board Ethernet port labeled LAN2. The I.P. Subnet Mask is reflected into the four contiguous registers, starting at 0x2026. The Ethernet I.P. Subnet Mask is Network Byte Ordered which has the following format:

eth1 I.P. Subnet Mask = 0x2026 . 0x2027 . 0x2028 . 0x2029

Note: The Ethernet I.P. Address and Subnet Masks for eth0 and eth1 will be changed from their power-up values to the addresses configured by the registers above following an Auto Initialize command. The Ethernet I.P. Address and Subnet Masks for eth0 and eth1 are stored on the VME-SBC's flash memory. Setting the Ethernet I.P. Address and Subnet Masks using the registers above **will not modify** the Ethernet I.P. Address and Subnet Masks stored on the flash memory. Therefore the VME-SBC will use the Ethernet I.P. Address and Subnet Masks from the flash memory as default from power-up. Refer the Section 7, "Network Configuration Using the Command Line Interface" of this document for specific instruction on setting the power-up default (flash stored) Ethernet Address parameters of the VME-SBC.

3.1.2 PEER-PRIMITIVE BUFFER

This section contains the command structure that allows the PLC to send Peer Primitive commands to transmit and receive the associated responses from the drive(s). Peer Primitive commands are to be used for all communications to and from the drive except for the Real-Time Data.

Offset	Description	Data Size	Notes
	COMM BUF 1:		
0x2400	BUFOWN	UINT8	Buffer Owner
0x2401	BUFCMD	UINT8	Buffer Command
0x2402	BUFSTATUS	UINT8	Buffer Status
0x2403	BUFCCHG	UINT8	Buffer Command Change
0x2404	RESPTIME	UINT16	Message Response Time-Out Value
0x2406	Reserved	UINT16	
0x2408	REQLEN	UINT16	Actual Request Length in Bytes
0x240A	RSPLN	UINT16	Actual Response Length in Bytes
0x240C	BUFREQDP	UINT32	Buffer Request Data Pointer
0x2410	BUFRSPDP	UINT32	Buffer Response Data Pointer
0x2414	BUFTYPE	UINT8	Buffer Contents Format
0x2415	COMMERR	UINT8	Communications Error Report
0x2416	DADDR	4 * UINT8	Destination I.P. Address
0x241A	Reserved	2 * UINT8	
0x241C	DLSAP	UINT8	Destination LSAP
0x241D	Reserved	3 * UINT8	
0x2420	DTASK	UINT8	Destination Task ID
0x2421	STASK	UINT8	Source Task ID
0x2422	PPRIM	990 * UINT8	Equivalent 802.3 Peer Primitive Contents
0x2800	COMM BUFF 2	1024 * UINT8	Structure in 0x2400 – 0x27ff repeated
	.		
0x600 to 0x63ff	COMM BUF 16	1024 * UINT8	Structure in 0x2400 – 0x27ff repeated

Table 2 – VME-SBC / PLC Interface – Peer Primitives

0x2400 – BUFOWN

0 = No Owner
Non-Zero = Owned

The client places a unique code into this field to indicate that this buffer is reserved and owned. A value of zero in this field indicates no owner – commands will not be executed from this buffer if BUFOWN is zero.

0x2401 – BUFCMD

0 = Do Nothing

The VME-SBC communication card will take no action.

1 = Send Acknowledged Message

The VME-SBC communication card will send the message pointed to by the request data pointer using LLC Type 3 service. Successful transmission results in a Transaction in Progress status being reported until a response is returned.

2 = Send Unacknowledged Message

This service is currently not supported by the VME-SBC communications card.

3 = Get Reply Data

This service is currently not supported by the VME-SBC communications card.

4 = Abort Message

The communication card will stop all processing on a message.

0x2402 – BUFSTATUS

0 = Do Nothing

The communication card is taking no action on this buffer.

1 = Transaction in Progress

The communication card is in the process of completing a message transaction.

2 = Response Returned

The communication card has successfully received a response to the transaction request and has placed it at the buffer pointed to by the response data pointer.

3 = Communications Error, No Confirmation

The communication card has received no acknowledgement of message delivery at the destination address.

4 = Communications Error, Problem Reported

The communication card has received an acknowledgment of receipt of the message at the destination address, but had a problem in delivering the message. The problem LLC status code is contained in the RSRVD1 field.

5 = Message Aborted

The communication card has stopped all processing on a message.

6 = Aborted all Transactions due to Illegal Procedure (Handshake Error)

The communication card received another buffer command before completing the previous command on this buffer.

7 = Message Time-out

The communication card did not receive a response to the peer-primitive message within the time specified by the RESPTIME parameter.

8 = Illegal Message Format

The communication card could not send the message due to incompatible information contained within the peer-primitive buffer. The communication card has stopped processing the message.

9 = Hardware Error

The communication card could not send the message due to a hardware error. The communication card could not complete the message.

0x2403 – BUFCCHG

0 = Set upon the MVME162's recognition of the command in BUFCMD.

1 = New Invocation of a Command.

0x2404 – RESPTIME

The default time period the communications card will wait for a response message resulting from the initiation of a peer-primitive acknowledged message is 10mS. A value (in nano-seconds) placed in this register prior to the initiation of the message will increase the communications card's default wait time for an acknowledgement by this amount. Note: The communication card will take no other action (other Peer-Primitives, RTD, EGD, etc) while waiting for receipt of a message.

0x2408 – REQLEN

Length in bytes of Request.

0x240A – RSPLLEN

Length in bytes of Response.

0x240C – BUFREQDP

This field contains an index to the send data. The index defines the starting byte within the PPRIM buffer of the Peer-Primitive Request message. The index must be within the buffer's 990 byte space – 0 to 990 respectively.

0x2410 – BUFRSPDP

This field contains an index to the response (received) data. The index defines the starting byte within the PPRIM buffer of the Peer-Primitive Response message. The index must be within the buffer's 990 byte space – 0 to 990 respectively.

0x2414 – BUFTYPE

Must be set to 2.

0x2415 – COMMERR

Communications error code. See section 5, 802.3 Peer-Primitive Structure for information regarding Error Messages.

0x2416 – DADDR

Ethernet I.P. Address of the node that is to receive the peer primitive request.

Destination I.P. Address = 0x2416 . 0x2417 . 0x20418 . 0x2419

0x241C – DLSAP

For send case: 4 = LSAP for Real-time Data (Not Supported by VME-SBC as Peer Primitive).
8 = LSAP for Messaging

For receive case: Not Used

0x2420 – DTASK

For send case: Not Used

For receive case: Source Buffer (this buffer's) COMM BUFF # .

0x2421 – STASK

Source Buffer (this buffer's) COMM BUFF #.

0x2422 - PPRIM

Peer Primitive request or response data adhering to the encoding rules defined in Section 4, "802.3 Peer Primitive Structure" of this document.

3.1.3 REAL-TIME DATA BUFFER

The Real-time Data Buffer is divided into four sections as follows:

- ◆ Autoscan Setup
- ◆ Autoscan Write
- ◆ Autoscan Read
- ◆ Miscellaneous Status

3.1.3.1 AUTOSCAN SETUP SECTION

Offset	Description	Data Size	Notes
0x6400	Number Setpoints per drive	UINT16	1 – 16. Power Up Default = 8.
0x6402	Autoscan Channel Enable Word	UINT16	Bit 00 = 1 = Channel 0 enabled Bit 01 = 1 = Channel 1 enabled Bit 15 = 1 = Channel 15 enabled Power Up Default = 0xFFFF (all channels enabled)
0x6404	Autoscan Channel 00 Drive ID	UINT16	0 – 254, Power Up Default to drive 1
0x6406	Autoscan Channel 01 Drive ID	UINT16	0 – 254, Power Up Default to drive 2
0x6422	Autoscan Channel 15 Drive ID	UINT16	0 – 254, Power Up Default to drive 16
0x6424	Analog Table Write Address	UINT32	Power Up Default to Avtron Standard (42)
0x6428	Digital Table Write Address	UINT32	Power Up Default to Avtron Standard (96)
0x642c	Response Time-Out Value	UINT16	Power Up Default = 0 uSec.
0x642e	Reserved	2*UINT16	

Table 3 – VME-SBC / PLC Interface – Real-time Data - Autoscan Setup

0x6400 - Number Setpoints per Drive – read / write

Tells the VME-SBC how many Floating Point Setpoint values to sent to each drive. This number effects all drives.

Refer to Table 3 above for Power Up default value.

A value written to this word **will only be recognized** following the issue of an Auto Initialize Command.

0x6402 - Autoscan Channel Enable Word – read / write

The VME-SBC can communicate with 1 to 16 drives. Each drive is assigned to a “Autoscan Channel” by configuring the appropriate Autoscan Channel Drive ID word. A drive is included in the Real-time Data Scan by setting (enabling) the appropriate bit in this word (bit 0 = channel 0, bit 1 = channel 1...bit 15 = channel 15). A drive is excluded from the scan by clearing (disabling) the appropriate bit. This value can be **Dynamically Changed** prior to the initiation of a scan.

0x6404 – Autoscan Channel Drive Ids – read / write

An Autoscan Channel is associated with a drive by setting the word for the desired channel to the last octet of the drive's Ethernet I.P. Address. Drive IDs can have the value of 0 to 254. 255 (0xff) is a reserved value. The VME-SBC formulates the complete 4 byte Ethernet I.P. Address of the drive by using the first three octets of its Ethernet I.P. Address (0x2004. 0x2005.0x2006) and appending the Drive ID as the last octet of the drive's Ethernet I.P. Address. For example, if the VME-SBC's Ethernet I.P. Address is 192.153.168.4, and the Autoscan Channel00 Drive ID is set to 100, then the VME-SBC will associate Autoscan Channel 00 to the drive with Ethernet I.P. Address is 192.153.168.100.

Drive Ethernet I.P. Address = 0x2004 . 0x2005 . 0x2006 . Drive ID

Refer to Table 3 above for Power Up default values.

Values written to these words **will only be recognized** following the issue of an Auto Initialize Command.

0x6424 – Analog Table Write Address – read / write

Refer to Table 3 above for Power Up default values.

A Value written to this word **will only be recognized** following the issue of an Auto Initialize Command.

0x6428 – Digital Table Write Address – read / write

Refer to Table 3 above for Power Up default values.

A Value written to this word **will only be recognized** following the issue of an Auto Initialize Command.

0x642c – RESPONSE TIME-OUT VALUE

The default time period the communications card will wait for the Autoscan Response messages from all drives which have been enabled by the Autoscan Channel Enable Word (0x6402) resulting from the initiation of a Change Buffer Owner (0x6bfc) command is 10mS. A value (in nano-seconds) placed in this register will increase the communications card's default message response wait time by this amount. Note: The communication card will take no other action (other Peer-Primitives, RTD, EGD, etc) while waiting for receipt of a message.

A Value written to this word **will only be recognized** following the issue of an Auto Initialize Command.

3.1.3.2 AUTOSCAN WRITE SECTION

NOTE: the following section ****MUST**** not exceed 999 UINT16's.

Offset	Description	Data Size	Notes
0x6432	Available	103 * UINT16	Available for inter-PLC communications and future expansion
0x6500	Performance View Service Configuration and PLC Data.	300 * UINT16	Refer to Section 3.3 of this document, "Performance View Interface".
0x6758	VME-162 Service Configuration	66 * UINT16	Refer to Section 3.2 of this document, "VME-162 Interface".
0x67dc	Drive 00 Setpoint	16 * FLOAT	Number FP's actually transmitted to drive specified by 0x6400
0x681c	Drive 01 Setpoint	16 * FLOAT	
	.		
0x6b9c	Drive 15 Setpoint	16 * FLOAT	
0x6bdc	Drive 00 Digital I/O	UINT16	
0x6bde	Drive 01 Digital I/O	UINT16	
	.		
0x6bfa	Drive 15 Digital I/O	UINT16	
0x6bfc	Change (Buffer Ownership)	UINT16	Bit 15 = 1 = PLC Initiating Communication Bit 15 = 0 = Communication Complete
0x6bfe	User Serial Number	UINT16	Optional user (PLC) defined transaction serial number.

Table 4 – VME-SBC / PLC Interface – Real-time Data - Autoscan Write

Setpoint Data (0x67dc – Setpoints and 6bdc – Digital I/O) – read / write

One to sixteen Analog Setpoint values will be sent to the drive as set by "Number Setpoints per Drive" (0x6400). Three to Thirty-three data items, with each data item representing 16 bits shall be sent to each ADD-32 drive. The first 2 to 32 data words are 1 to 16 floating point numbers (depending on the setting of "Number Setpoints per Drive" (0x6400). The last data word is digital bits sent as one 16 bit data word. The particular signals or I/O points sent as real time data are determined by the ADD-32 drive program and configuration. PLC ladder logic access to the real time data in VME memory must have a priori knowledge of its contents.

The Setpoint Data is sent to the drive as 1 to 16 floating point numbers (which are each two words long) and 1 digital word of binary setpoints. The ADD-32 receives all data in LITTLE ENDIAN format. If the Endian (0x200a) is set to 0000, the Setpoint Data will be stored in the Setpoint Data buffer in the LITTLE ENDIAN format. Set Endian to 1111 for the Setpoint Data items to be stored in the Setpoint Data buffer in the BIG ENDIAN format. This command allows alternate PLC's that are BIG ENDIAN to easily use the real time data without byte converting data sent to the VME-SBC card. Refer to the section "SYSTEM CONTROL BUFFER, 0x200a – Endian" earlier in this document for discussion of LITTLE / BIG ENDIAN formats.

Refer to "Section 4, Real Time Data Format" later in this document for Setpoint Data format details.

0x6bfc – Change – read / write

Bit 15 of this UINT16 is set by the PLC to 1 to initiate a Real-time Data scan and is cleared to 0 by the VME-SBC upon receipt of the Real-time Data.

NOTE: This location is shared between the Autoscan Read Section and the Autoscan Write Section.

0x6bfe – User Serial Number – read / write

This location is optionally set by the PLC to a user defined transaction serial number. The data at this location is copied from this location to the “**Echo of Serial Number**” (0x7026) by the VME-SBC immediately before clearing the Change flag. The VME-SBC makes no other use of the data contained at this location.

NOTE: This location is shared between the Autoscan Read Section and the Autoscan Write Section.

3.1.3.3 AUTOSCAN READ SECTION

NOTE: the following section ****MUST**** not exceed 999 UINT16's.

Offset	Description	Data Size	Notes
0x6bfc	Change (Buffer Ownership)	UINT16	Bit 15 = 1 = PLC Initiating Communication Bit 15 = 0 = Communication Complete
0x6bfe	User Serial Number	UINT16	Copied from Write Buffer at end of scan
0x6c00	Drive 00 Feedback's	8 * FLOAT	
0x6c20	Drive 01 Feedback's	8 * FLOAT	
	.		
0x6de0	Drive 15 Feedback's	8 * FLOAT	
0x6e00	Drive 00 Digital I/O Response	16 * UINT16	
0x6e20	Drive 01 Digital I/O Response	16 * UINT16	
	.		
0x6fe0	Drive 15 Digital I/O Response	16 * UINT16	
0x7000	Channel Status Word	UINT16	Bit 00 = 0 = Channel 0 Comm OK Bit 01 = 0 = Channel 1 Comm OK Bit 15 = 0 = Channel 15 Comm OK
0x7002	Drive 00 Extended Status Flags	UINT16	Bit 00 = 1 = Error 00 occurred Bit 01 = 1 = Error 01 occurred Bit 15 = 1 = Error 15 occurred
0x7004	Drive 01 Extended Status Flags	UINT16	
	.		
0x7020	Drive 15 Extended Status Flags	UINT16	
0x7022	Copy of Diagnostic Status	UINT8	Copy from System Buffer at end of scan
0x7023	Copy of Board Command	UINT8	Copy from System Buffer at end of scan
0x7024	Copy Mode Status	UINT8	Copy from System Buffer at end of scan
0x7025	Copy 802.3 Status	UINT8	Copy from System Buffer at end of scan
0x7026	Echo of User Serial Number	UINT16	Copied from User Serial Number at end of scan.
0x7028	RTD Interface Status	UINT16	
0x702a	MVME-162 Interface Status	UINT16	
0x702c	Perf. View Interface Status	UINT16	
0x702e	Reserved	UINT16	
0x7030	RTD Invocation Counter	UINT32	
0x7034	MVME-162 Invocation Counter	UINT32	
0x7038	Perf. View Invocation Counter	UINT32	
0x703C	VME-162 System Control Buffer	10 * UINT16	
0x7050	Reserved	472 * UINT16	Reserved for future expansion

Table 5- VME-SBC / PLC Interface – Real-time Data - Autoscan Read

0x6bfc – Change – read / write

Bit 15 of this UINT16 is set by the PLC to 1 to initiate a Real-time Data scan and is cleared to 0 by the VME-SBC upon receipt of the Real-time Data.

NOTE: This location is shared between the Autoscan Read Section and the Autoscan Write Section.

0x6bfe – User Serial Number – read / write

This location is optionally set by the PLC to a user defined transaction serial number. The data at this location is copied from this location to the “**Echo of Serial Number**” (0x7026) by the VME-SBC immediately before clearing the Change flag. The VME-SBC makes no other use of the data contained at this location.

NOTE: This location is shared between the Autoscan Read Section and the Autoscan Write Section.

Feedback Data (0x6c00 – Feedbacks and 0x6e00 – Digital I/O Response) – read only

Thirty-two data items with each data item representing 16 bits shall be retrieved from each ADD-32 drive. The first 16 data words are 8 floating point numbers. The second 16 data words are digital bits read in a word at a time. The particular signals or IO points returned as real time data are determined by the ADD-32 drive program and configuration. PLC ladder logic access to the real time data in VME memory must have a priori knowledge of its contents.

The Feedback Data is returned by the drive as 8 floating point numbers (which are each two words long) and 16 digital words. The ADD-32 returns all data in LITTLE ENDIAN format. If the Endian (0x200a) is set to 0000, the Feedback Data items will be stored in the Feedback Data buffer in the LITTLE ENDIAN format. Set Endian to 1111 for Feedback Data to be stored in the real time data buffer in the BIG ENDIAN format. This command allows alternate PLC's that are BIG ENDIAN to easily use the real time data without byte converting data sent to the VME-SBC card. Refer to the section “SYSTEM CONTROL BUFFER, 0x200a – Endian” earlier in this document for discussion of LITTLE / BIG ENDIAN formats.

Refer to “Section 4, Real Time Data Format” later in this document for Setpoint Data format details.

0x7000 – Channel Status Word – read only

This word is a bit mask which summarizes the communication status of the channels. A value of 0 in the appropriate bit (bit 0 = channel 0, bit 1 = channel 1...bit 15 = channel 15) indicates Communications is functioning correctly. A value of 1 indicates a communications failure. The exact cause of failure is enunciated in the appropriate Extended Status Flag for the channel (0x7002 - 0x7020).

0x7002 – Extended Status Flags – read only

Each word (one per channel) contains a bit mask which indicates the specific communications error which occurred with the drive associated with that channel. The bit mask is defined as follows:

Bit 00 = 1 = Response Time Out (> 20 ms)

One or more of these bits being set will cause the bit for the corresponding channel in the Channel Status Word (0x7000) to be set.

0x7022 – Copy of Diagnostic status – read only

This byte is copied from System Buffer location 0x2000 just prior to clearing of Change (0x6bfc).

- 0 = Board Power Up / Reset in Progress
- 1 = Board Power Up / Reset Complete

0x7023 – Copy of Board Command – read / write

This byte is copied from System Buffer location 0x2001 just prior to clearing of Change (0x6bfc).

- 0 = Do Nothing
- 1 = Perform System Reset
- 2 = Stop, Halt Ethernet Communication
- 3 = Start, Begin Ethernet Communication
- 4 = Auto Initialize (see section **VME-SBC AUTO INITIALIZE SEQUENCE** later in this document)
- 5 = Perform Buffer Reset (see section **DATA TABLE RESET COMMAND** later in this document)

0x7024 – Copy of Mode Status – read only

This byte is copied from System Buffer location 0x2002 just prior to clearing of Change (0x6bfc).

- 0 = Not Ready for Transactions
- 1 = Ready for Transactions
- 2 = Fatal Error – Board Inoperable
- 3 = Non-Fatal Error – Board Operable

0x7025 – Copy of 802.3 Status – read only

This byte is copied from System Buffer location 0x2003 just prior to clearing of Change (0x6bfc).

0x7026– Echo of User Serial Number – read only

This word is copied from User Serial Number Buffer (0x6bfe) by the VME-SBC just prior to clearing of Change (0x6bfc). . The VME-SBC makes no other use of the data contained at this location.

0x7028 – RTD Interface Status – read only

The VME-SBC updates this status word to report the success of or any errors encountered while processing the RTD Interface. The RTD Interface is processed on each invocation initiated by the PLC setting the Change Buffer Ownership (0x6bfc) word.

- 0x0001 = Ethernet Socket Error
- 0x0010 = RTD Interface Success

0x702a – MVME-162 Interface Status – read only

The VME-SBC updates this status word to report the success of or any errors encountered while processing the MVME-162 Interface. The MVME-162 Interface is processed, when enabled (0x6758), upon on each invocation initiated by the PLC setting the Change Buffer Ownership (0x6bfc) word.

0x0001 = Time-out waiting on MVME-162 to clear Buffer Ownership Flag
0x0002 = Auto Initialize of MVME-162 in progress
0x0004 = Failure trying to Auto Initialize the MVME-162
0x0010 = RTD Interface Success

0x702c – Performance View Interface Status – read only

The VME-SBC updates this status word to report the success of or any errors encountered while processing the MVME-162 Interface. The MVME-162 Interface is processed, when enabled (0x6758), upon on each invocation initiated by the PLC setting the Change Buffer Ownership (0x6bfc) word.

0x0001 = Ethernet Socket Error
0x0010 = Success processing PLC Data Buffer EGD message
0x0020 = Success processing MVME-162 EGD message

0x7030 – RTD Invocation Counter – read only

This 32-Bit integer is incremented on each successful attempt by the VME-SBC to complete the Autoscan service requested by the PLC. The Autoscan service is initiated by the Change Buffer Ownership byte (0x6bfc).

0x7034 – MVME-162 Invocation Counter – read only

This 32-Bit integer is incremented on each successful attempt by the VME-SBC to complete the MVME-162 Interface service requested by the PLC. The MVME-162 Interface service is initiated, when enabled, by the Change Buffer Ownership byte (0x6bfc).

0x7038 – Performance View Invocation Counter – read only

This 32-Bit integer is incremented on each successful attempt by the VME-SBC to complete the Performance View Interface service requested by the PLC. The Performance View Interface service is initiated, once if any EGD Channel is enabled (0x6500) and again if the PLC Data buffer is enabled (0x6502), for each initiation of the Change Buffer Ownership byte (0x6bfc).

0x703c – MVME System Control Buffer – read Only

The VME-SBC echo's the contents of the VME-162 Control Buffer (0x2000 – 0x2015) for optionally monitoring the status of the MVME-162. The registers are marked as 'Read Only' to prevent user interaction of the automatic VME-SBC control the MVME-162.

3.2 MVME-162 INTERFACE

The interface scheme shown in **Table 6** of the following section defines the interface to the service provided by the VME-SBC to an optional MVME-162 card. The interface to an optional MVME-162 card allows the Real-Time data buffer of a MVME-162 card to be encapsulated (on a drive by drive basis) within the VME-SBC's real time data buffer. The MVME-162 service allows for the migration from legacy 802.3 system to 802.4 Ethernet systems on a drive by drive basis, with minimal PLC ladder logic changes.

NOTE: All buffer layouts provided in the sections which follow are organized in BIG ENDIAN memory organization (see **0x200a – Endian** below for memory organization details).

3.2.1 MVME-162 INTERFACE CONTROL BUFFER

This section contains values used to control the operation of the VME-SBC and its interface with an optional MVME-162 card. The VME-SBC will assume control of and manipulate the MVME-162 "System Control Buffer" and "Real-time Data Setup Buffer" as well as monitor the status of the MVME-162 card to insure that it is functional and ready for Real-time Data transactions. The VME-SBC will manipulate the MVME-162 card's Real-time Data buffer as required to initiate Real-time Data transactions and encapsulate the setpoint and feedback data within the VME-SBC's Real-time Data buffer.

Offset	Description	Data Size	Notes
0x6758	MVME-162 Interface Enable	UINT8	Power Up Default to Disabled, 0x00
0x6759	MVME-162 Buffer Structure	UINT8	Power Up Default to 682105, 0x00
0x675a	MVME-162 Base Address	UINT8	Power Up Default to 0x60
0x675b	Reserved	UINT8	
0x675c	MVME Channel Enable	UINT16	Power Up Default to All Disabled, 0x0000
0x675e	Reserved	6 * UINT8	
0x6764	Rack Number	UINT16	Power Up Default to 0
0x6766	Reserved	59 * UINT16	

Table 6 – VME-SBC / MVME-162 Interface Control Buffer

0x6758 – MVME-162 Interface Enable – Read / Write

The VME-SBC will not provide the MVME-162 interface service and control when "The MVME Interface is Disabled". Data within the remaining MVME-162 Interface Control Buffer is ignored. When "The MVME Interface is Enabled", the VME-SBC assumes control of and encapsulates the MVME-162's real time date with that of the VME-SBC's real time date for each section defined by the MVME Channel Enable word. A Value written to this word **will only be recognized** following the issue of an Auto Initialize Command.

- 0x00 – The MVME Interface is Disabled (default)
- 0x11 – The MVME Interface is Enabled

0x6759 – MVME-162 Buffer Structure – Read / Write

The VME-SBC supports Real-time Data Scan control of Legacy 802.4 MVME-162 cards containing either of following software part numbers. A Value written to this word **will only be recognized** following the issue of an Auto Initialize Command.

- 0x00 – The MVME-162’s software is 682105 (default)
- 0x01 – The MVME-162’s software is 681932

0x675a – MVME-162 Base Address – Read / Write

Base Address if the MVME-162 card from which Real-time Data is to be obtained. The Base Address if the MVME-162 card is set by the jumper configuration on board A22614 of the MVME-162 card as illustrated in Table 7. A Value written to this word **will only be recognized** following the issue of an Auto Initialize Command.

VME-162 Base Address	Jumper X4	Jumper X2	Jumper X1
0xE8	X	X	X
0xE0	X	X	
0xD8	X		X
0xD0	X		
0x78		X	X
0x70		X	
0x68			X
0x60			

Table 7 – VME-162 Base Address board, PN A22614

0x675c – MVME Channel Enable Word – Read / Write

The VME-SBC can interface with the Real-time Data channels of 0 to 16 drives within a MVME-162 module. The VME-SBC controls and monitors the MVME-162 such that it is functional and ready for Real-time Data transactions. A drive’s Real-time Data from a VME-162 card is encapsulated into the VME-SBC Real-time Data Scan by setting (enabling) the appropriate bit in this word (bit 0 = channel 0, bit 1 = channel 1...bit 15 = channel 15). If a corresponding channel is also enabled within the VME-SBC’s Autoscan Channel Enable Word (0x6402), this MVME Channel Enable bit is ignored. A drive’s Real-time Data from a MVME-162 card excluded from the VME-SBC Real-time Data Scan by clearing (disabling) the appropriate bit. This value can be **Dynamically Changed** prior to the initiation of a VME-SBC Real-time Data Scan.

0x6764 – Rack Number – Read / Write

The MVME-162 MAC address is set to a default value during the Auto-Initialize sequence as follows:

$$\text{MAC Address} = 0x1200\ 0000\ 0000 + [(\text{Rack Number} * 256) + \text{Board ID}) * 2]$$

Where 'Board ID' is determined from the from the X1, X2 and X4 jumper configuration on board A22614. See 0x675a for additional information.

Note: Settings from the System Control Buffer and the Autoscan Setup buffer of the MVME-162 are populated (where necessary) by the VME-SBC, from the VME-SBC's System Control and Autoscan Setup buffers:

VME-SBC		MVME-162
System Control Buffer:		
MAC Address	→	MAC Address
Rack Number*	→	Rack Number
Endian	→	Endian
Autoscan Setup Buffer:		
Number of Setpoints per Drive	→	Number of Setpoints per Drive
Channel Drive IDs	→	Channel Drive IDs
Analog Table Write Address	→	Analog Table Write Address
Digital Table Write Address	→	Digital Table Write Address

* The MVME-16e Interface Control Buffer is the source location for this parameter within the VME-SBC

3.3 PERFORMANCE VIEW INTERFACE

The interface scheme shown in **Table 8** in the following section define the interface to the service provided by the VME-SBC to an optional Performance View System. The Performance View service provides 802.3 UDP Ethernet Global Data (EGD) messages containing selected drive(s) Real-Time Data buffer and a PLC data buffer to a Performance View system. Refer to Section 6 “EGD Message Format” for further information.

NOTE: All buffer layouts provided in the sections which follow are organized in BIG ENDIAN memory organization (see **0x200a – Endian** below for memory organization details).

3.3.1 PERFORMANCE VIEW INTERFACE CONTROL BUFFER

This section contains values used to control the operation of the VME-SBC and its communications to an optional Performance View System.

Offset	Description	Data Size	Notes
0x6500	Performance View EGD Channel Enable	UINT16	
0x6502	PLC EGD Data Buffer Enable	UINT16	
0x6504	Performance View I.P. Address	4 * UINT8	
0x6508	Reserved	4 * UINT16	
0x6510	Channel 00 Drive EGD Msg. ID	UINT16	0 – 65535, Power Up Default to drive 1
0x6512	Channel 01 Drive EGD Msg. ID	UINT16	0 – 65535, Power Up Default to drive 2
	.		
0x652e	Channel 15 Drive EGD Msg. ID	UINT16	0 – 65535, Power Up Default to drive 3
0x6530	PLC Data Buffer EGD Msg. ID	UINT16	0 – 65535, Power Up Default to drive 17
0x6532	PLC Data Buffer Length	UINT16	0 – 256, Power Up Default to drive 64
0x6534	PLC Data Buffer Endian	UINT16	Default: little endian.
0x6536	UNIT32 Endian Convert Length	UINT16	0 – 256, Power Up Default to 16 Floats
0x6538	PLC Data Buffer	256 * UINT16	
0x6738	Reserved	16 * UINT16	

Table 8 – VME-SBC / Performance View Interface Control Buffer

0x6500 – Performance View EGD Channel Enable Word – read / write

The VME-SBC can transmit the Real-time Data Channels of 0 to 16 drives to an optional Performance View system. The VME-SBC will transmit a drive’s Real-time Data from the Real-time Data Buffer to the Performance View I.P. Address by setting (enabling) the appropriate bit in this word (bit 0 = channel 0, bit 1 = channel 1...bit 15 = channel 15). Transmission of a drive’s Real-time Data buffer is excluded by clearing (disabling) the appropriate bit. This value can be **Dynamically Changed** prior to the initiation of a VME-SBC Real-time Data Scan.

0x6502 – PLC Data Buffer Enable – read / write

The VME-SBC supports transmission of a data buffer, filled with user defined data, originating from the PLC to a Performance View system. This value can be **Dynamically Changed** prior to the initiation of a VME-SBC Real-time Data Scan.

0x0000 – Disable Transmission of PLC Data buffer to Performance View

0x1111 – Enable Transmission of PLC Data buffer to Performance View

0x6504 – Performance View I.P. Address – read / write

Ethernet I.P. Address of the target Performance View collector. The I.P. Address is reflected into the four contiguous registers, starting at 0x6504, with the following format:

I.P. Address = 0x6504 . 0x6505 . 0x6506 . 0x6507

A Value written to this word **will only be recognized** following the issue of an Auto Initialize Command.

0x6510 – Channel Drive EGD Message IDs – read / write

An EGD message is produced for each Real-time Data Channel enabled by the Drive EGD Channel Enable Word (0x6500). Each EGD message produced by the VME-SBC includes the Channel Drive EGD Message ID for the specific Real-time Data buffer being transmitted. The Performance View collector uses the Channel Drive EGD Message ID within the EGD message to uniquely identify the contents as originating to a particular Real-time Data Channel. Values written to these words **will only be recognized** following the issue of an Auto Initialize Command.

0x6530 – PLC Data Buffer EGD Message ID – read / write

An EGD message is produced containing the contents of the PLC Data Buffer (0x6534), when the PLC Data Buffer Enable (0x6502) is set. This EGD message includes in the PLC Data Buffer EGD Message ID such that the Performance View collector can uniquely identify the contents as originating from the PLC Data Buffer. A Value written to this word **will only be recognized** following the issue of an Auto Initialize Command.

0x6532 – PLC Data Buffer Length – read / write

The length of the PLC Data Buffer size to be transmitted. The value measured in Words (16 bits, UINT16). A value of 0 will send an EGD message with no data, alternately a value greater than 256 will be truncated to a value of 256. A Value written to this word **will only be recognized** following the issue of an Auto Initialize Command.

0x6534 – PLC Data Buffer Endian – read / write

0x0000 – The PLC Data Buffer memory organization is Little Endian

0x1111 – The PLC Data Buffer memory organization is Big Endian

A Value written to this word **will only be recognized** following the issue of an Auto Initialize Command.

0x6536 – UNIT32 Endian Convert Length – read / write

If the PLC Data Buffer Endian (0x6534) is set to 'Little Endian', this register is unused by the VME-SBC. If the PLC Data Buffer Endian (0x6534) is set to 'Big Endian', this register defines the number of UINT32 (or floating point) values to interpret as Big Endian. This value of UINT32 registers, starting from the beginning of the PLC Data buffer, will be converted to Little Endian as UINT32. The remaining length of the PLC Data buffer will be converted as to Little Endian as UINT16. A Value written to this word **will only be recognized** following the issue of an Auto Initialize Command.

0x6538 – PLC Data Buffer – read / write

The contents of the data written to this register will be transmitted to the Performance View, when the PLC Data Buffer Enable (0x6502) is set. The Performance View expects all UINT16 and UINT32 data to be in LITTLE Endian format. Refer to refer to register 0x6534 and 0x6536 above for information regarding the Endian Conversion service provided by the VME-SBC to the PLC Data Buffer data.

3.4 VME-SBC AUTO INITIALIZE SEQUENCE

The current initialization sequence for the VME-SBC consists of many steps and is overly complex. Through the use of default values, the initialization sequence is greatly simplified to the following steps:

1. Optionally set "Endian" (0x200a).
2. Optionally "eth0 I.P. Address" (0x2016).
3. Optionally "eth0 I.P. Subnet Mask" (0x201a).
4. Optionally set "Number Setpoints per Drive" (0x6400).
5. Optionally set "Channel Enable Word" (0x6402).
6. Optionally set "Channel Drive IDs" (0x6404 – 0x6422).
7. Optionally set "Analog Table Write Address" (0x6424).
8. Optionally set "Digital Table Write Address" (0x6428).
9. Optionally set Performance View Control Buffer parameters (0x6570 – 0x65a4).
10. Optionally set VME-162 Interface Control Buffer parameters (0x67cc – 0x67cf).
11. Issue "Auto Initialize" command – write 0x04 to "VME-SBC System Control Board Command" (0x2001).
12. Wait for "Board Command" to return to zero.

All optional values will remain at their respective Power Up Default Values unless explicitly set prior to the issue of the Auto Initialize Command.

The order in which items 1 to 6 above are written (if desired) is not critical; however, any values to be written must be done prior to the issue of the "Auto Initialize" Command.

Upon successful completion of the Auto Initialize Sequence, the Mode Status should reflect a 'Ready for Transactions' code.

3.5 DATA TABLE RESET COMMAND

Issuing this command -- write 0x05 to "VME-SBC System Control Buffer Command" (0x2001) – causes all data tables (VME-SBC System Control, Peer Primitive, Real-time Data) values to be reset to their power up values (see Table 1).

NOTE: It is the responsibility of the PLC program to insure the drives are in a safe condition prior to issuing of this reset command.

3.6 VME-SBC POWER UP SEQUENCE

Upon a Power Up reset, the VME-SBC executes the following initialization sequence:

- I. Boot Linux operating system.
- II. Run Linux boot scripts.
- III. Run Application:
 1. Initialize VME bus access
 2. Initialize various registers to power-up default values
 3. Clear VME-SBC System Control buffer to zero
 4. Set Diagnostic Status to one
 5. Clear Board Command to zero
 6. Clear Real-time Data Buffer to zero
 7. Clear Communications tracking buffers to zero
 8. Reset miscellaneous flags and counters
 9. Set Mode to Ready for Transactions

4 REAL TIME DATA FORMAT

Data as read in from / stored to drive:

First Byte	Byte 1	First Floating Point Number
Second Byte	Byte 2	“
Third Byte	Byte 3	“
Fourth Byte	Byte 4	“
.	.	
.	.	
Thirty-third Byte	Byte 33	First Digital Word
Thirty-fourth Byte	Byte 34	“
Thirty-fifth Byte	Byte 35	Second Digital Word
Thirty-sixth Byte	Byte 36	“
.	.	
.	.	

How data is stored in Real time Data buffers:

LITTLE ENDIAN

	Address Offset +	0	1	2	3
Addr offset 0	FLT PT 1	BYTE 1	BYTE 2	BYTE 3	BYTE 4
Addr offset 4	FLT PT 2	BYTE 5	BYTE 6	BTYE 7	BTYE 8
	
	
Addr offset 32	DIG WORD 1&2	BYTE 33	BYTE 34	BYTE 35	BYTE 36
Addr offset 36	DIG WORD 3&4	BYTE 37	BYTE 38	BYTE 39	BYTE 40
	
	

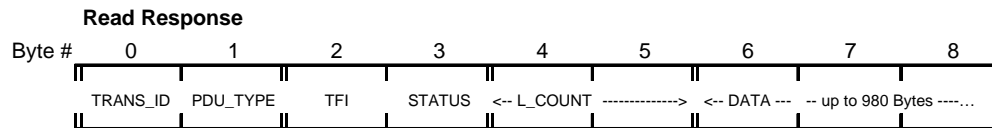
BIG ENDIAN

	Address Offset +	0	1	2	3
Addr offset 0	FLT PT 1	BYTE 4	BYTE 3	BYTE 2	BYTE 1
Addr offset 4	FLT PT 2	BYTE 8	BYTE 7	BTYE 6	BTYE 5
	
	
Addr offset 32	DIG WORD 1&2	BYTE 34	BYTE 33	BYTE 36	BYTE 35
Addr offset 36	DIG WORD 3&4	BYTE 38	BYTE 37	BYTE 40	BYTE 39
	
	

5 802.3 PEER-PRIMITIVE STRUCTURE

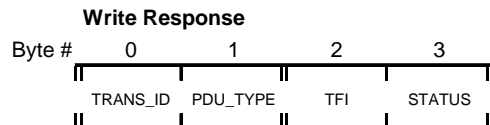
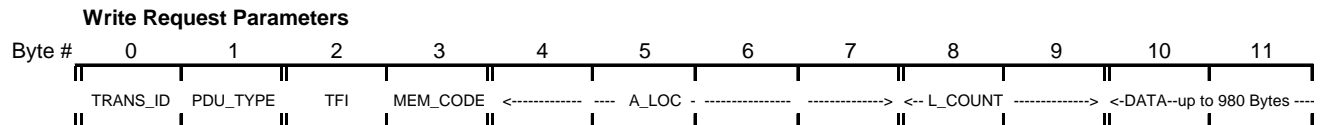
5.1 Block Read Request/Response

The Block Read Request is used to obtain a contiguous block of data from a drive. Upon successful execution of a Block Read Request, the drive will return the requested data, available in the format specified by the Block Read Response.



5.2 Block Write Request/Response

The Block Write Request is used to write a contiguous block of data to a drive. Upon successful execution of a Block Write Request, the drive accept the data contained therein and return a acknowledgment in the form specified by the Block Write Response.



5.3 Peer-Primitive Buffer Mnemonic Definitions

- TRANS_ID - Transaction ID number. Value specified within a 'Request' message that is returned by the target node's 'Response' message. The value is used to uniquely identify "response" messages when multiple 'Request' messages can / are pending.
- PDU_TYPE - Protocol Data UINT Type. Specified to inform the target device of the format of the data contained within the message.

PDU Type	Description	Notes
PDU_TYPE = 0x01	Error Response	
PDU_TYPE = 0x04	Status Request	Not Supported by VME-SBC
PDU_TYPE = 0x05	Status Response	Not Supported by VME-SBC
PDU_TYPE = 0x06	Config Request	Not Supported by VME-SBC
PDU_TYPE = 0x07	Config Response	Not Supported by VME-SBC
PDU_TYPE = 0x08	Capabilities Request	Not Supported by VME-SBC
PDU_TYPE = 0x09	Capabilities Response	Not Supported by VME-SBC
PDU_TYPE = 0x0A	Reset Request	Not Supported by VME-SBC
PDU_TYPE = 0x0B	Reset Response	Not Supported by VME-SBC
PDU_TYPE = 0x0E	Read Request	
PDU_TYPE = 0x0F	Read Response	
PDU_TYPE = 0x10	Write Request	
PDU_TYPE = 0x11	Write Response	
PDU_TYPE = 0x12	Read-Mod-Write Request	Not Supported by VME-SBC
PDU_TYPE = 0x13	Read-Mod-Write Response	Not Supported by VME-SBC

Table 9 – PDU Types

- TFI - Transport Format Identification. Always set to 0x02 for communication to / from ADDvantage32 drives.
- MEM_CODE - Memory Code, in the case of an ADDvantage32 drive, the DATA FILE of the requested source or written destination data. Refer to the Table 10 below for a abbreviated list of available ADDvantage32 DATA FILEs. Refer to the ADDvantage32 Technical Manual for a complete list.

Data Type	Value	Data_type
analog_io	09	Float
table	17	Float
dig_io	28	Unsigned16
Faultfifo	60	Unsigned16

Table 10 – ADDvantage32 Data Files – Mem_code

- A_LOC - Address Location (address offset) from the start to the data file specified by the MEM_CODE value from / to where the data is to be read / written.
- L_COUNT - Length Count. The amount of data (in Bytes) to be read from or written to the source / destination pointed to by the MEM_CODE and A_LOC values.
- STATUS - Stats or Error Code associated as a result of a 'Request' message. Included as part of a 'Response' message, indicates success or failure of a 'Request' message. Possible Error code values and definitions are listed within Table 11.

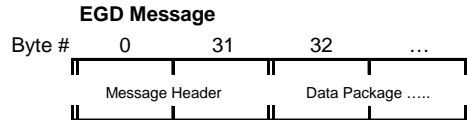
Error Message	Error Code
Service is not supported	0
Memory type [Mem_type] is not supported	1
Address [A_loc] is too large	2
Not enough Parm_detail received for Service	3
Wrong mode for service execution	4
Address [A_loc] is write protected	5
No response from attached device	6
Service aborted due to a fatal error	7
Error encountered while executing the service with [A_loc]	8
Invalid service for the address [A_loc]	9
Data requested exceeds the maximum allowed	10
Record number [Rec_num] is too large	11
Record number [Rec_num] is not defined	12
Invalid encoding of Parm_detail	13
Zero locations requested (N_count=0)	14
Illegal floating point value an IEEE Not-a-Number) in service write request	15
Service request denied, device in load mode	16
Service request denied, load mode needs to be initialized	17
Data count value [N_count] did not match actual data count	18
Hot back up UINT did not respond	19
Error code [Unsigned16] returned from device	20
Device did not respond to request	21
Message Acknowledged as Success	80

Table 11 - Status Response

6 EGD MESSAGE FORMAT

The Ethernet UDP messages produced by the VME-SBC as part of the Performance View interface adheres to the following format. The format is consistent with the EGD message structure as defined by G.E. Fanuc at the time of this publication.

The EGD message contains a fixed 32 Byte Message header and a data package of variable length:



The message header contains the following information:

Parameter Name	Length	Description
PDTypeVersion	UINT16	Type=13 (0Dh) in low byte, version=1 in high byte
RequestID	UINT16	Incremented every time data produced.
ProducerID	UINT32	The TCP/IP address of the VME-SBC .
ExchangeID	UINT32	Set by registers 0x6510 through 0x6530, to identify the message's source Channel ID or PLC Data buffer ID.
TimeStampSec	UINT32	Not Used by VME-SBC
TimeStampNanoSec	UINT32	Not Used by VME-SBC
Status	UINT32	Not Used by VME-SBC
ConfigSignature	UINT32	Not Used by VME-SBC
Reserved	UINT32	Not Used by VME-SBC

The Data Package for each Real-Time Data Channel is of length 64Bytes, formatted into Little Endian and packaged as follows:

Data Contents	0	1	2	3
FLT PT 1	BYTE 1	BYTE 2	BYTE 3	BYTE 4
FLT PT 2	BYTE 5	BYTE 6	BTYE 7	BTYE 8

FLT PT 8	BYTE 29	BYTE 30	BTYE 31	BTYE 32
DIG WORD 1&2	BYTE 33	BYTE 34	BYTE 35	BYTE 36
DIG WORD 3&4	BYTE 37	BYTE 38	BYTE 39	BYTE 40

DIGWORD 15&16	BYTE 61	BYTE 62	BYTE 63	BYTE 64

The Data Package for the PLC Data Buffer is of length defined by register 0x6532, and Endian formatted as defined by registers 0x6534 and 0x6536.

7 NETWORK CONFIG. USING THE COMMAND LINE INTERFACE

The VME-SBC includes a small editor program called 'nano' for modifying text-based files. Before modifying any file, it is a good idea to make a back-up copy of the file. Improper modifications to files can cause undesirable device operation. A directory on the flash disk has been provided '/usr/avtron/backup' for saving backup copies of files. To copy a file using the Linux command line:

```
[root@VME_SBC /root]# cp /etc/sysconfig/network /usr/avtron/backup/network.backup.001
```

The example copies the file 'network', located in the /etc/sysconfig directory to the directory /usr/avtron/backup with filename network.backup.001

Use the 'nano' editor to modify the file /etc/sysconfig/network-scripts/ifcfg-eth0 to contain the desired default values of the VME-SBC's on-board Ethernet port labeled LAN-1 (eth0).

```
[root@VME_SBC /root]# nano /etc/sysconfig/network-scripts/ifcfg-eth0
```

The contents of the file *ifcfg-eth0* should be similar to:

```
DEVICE=eth0  
ONBOOT=yes  
BOOTPROTO=static  
IPADDR=10.1.208.101  
NETMASK=255.255.0.0
```

Note: Modify only the IPADDR and NETMASK values!

Use the 'nano' editor to modify the file /etc/sysconfig/network-scripts/ifcfg-eth1 to contain the desired default values of the VME-SBC's on-board Ethernet port labeled LAN-2 (eth1).

```
[root@VME_SBC /root]# nano /etc/sysconfig/network-scripts/ifcfg-eth1
```

The contents of the file *ifcfg-eth1* should be similar to:

```
DEVICE=eth1
ONBOOT=yes
BOOTPROTO=static
IPADDR=10.1.208.102
NETMASK=255.255.0.0
```

Note: Modify only the IPADDR and NETMASK values!

Use the 'nano' editor to modify the file */etc/sysconfig/network* to contain the desired default values for the VME-SBC's HOSTNAME and GATEWAY Address.

```
[root@VME_SBC /root]# nano /etc/sysconfig/network
```

The contents of the file *ifcfg-eth1* should be similar to:

```
NETWORKING=yes
HOSTANME=VME_PC
GATEWAY=10.1.205.10
```

Note: Modify only the HOSTNAME and GATEWAY parameters!

8 VME-SBC BASE ADDRESS DETERMINATION

The VME-SBC occupies a 24-Bit (A24) address space on the VME bus. The VME-SBC base address is defaulted to 0xE8. The 'offset' addresses outlined within this document refer to the number of bytes offset from this base address. As an example, access to the VME-SBC's Diagnostic Status from the VME bus would be accomplished from VME bus address 0xE82000. The VME-SBC base address is determined at power-up or following a 'System Reset' command.

The default VME-SBC is calculated by the VME-SBC buy asserting the DTR line of COM1 and reading the status of the DCD, DSR and CTS lines (also of COM1). Table 9 gives the resulting VME-SBC base address where 'X' represents a connection of the COM1 DTR line with the indicated DCD, DSR and/or CTS lines of COM1.

VME-SBC Base Address	Jumper CTS	Jumper DSR	Jumper DCD
0xE8			
0xE0*			X
0xD8		X	
0xD0		X	X
0x78	X		
0x70	X		X
0x68	X	X	
0x60	X	X	X

Table 9 – VME-SBC Base Address

Note: VME-SBC Base Address 0xE0 can be obtained by installing connector assembly, Avtron PN A29315 to COM1 of the VME-SBC.

Note: COM2 of the VME-SBC is used by the operating system as a serial console port. No modification or alternate use of this port should be attempted by the user.

9 RESULTS

This document was written in accordance with current Avtron Manufacturing, Inc. ISO requirements.

10 CONCLUSION

This document describes the required functional specification for the VME-SBC software.

A handwritten signature in black ink that reads "Steve Kovacic". The signature is written in a cursive, flowing style.

Steve Kovacic
Project Engineer
March 14, 2003