Model V243

96-channel, Low Level Multiplexer

INSTRUCTION MANUAL

February 26, 2001

*****Special Option*****

Model V243-S001

96-channel, Low-level Signal Conditioner

August 21, 1997

*****Special Option*****

Model V243-S001

The Model V243-S001 is the same as the V243-VA41 except it has been modified to have filter settings to 2 Hz, 10 Hz, and 50 Hz.

August 21, 1997

*****Special Option*****

Model V243-S002

96-channel, Low-level Signal Conditioner

March 31, 2000

*****Special Option*****

Model V243-S002

The Model V243-S002 is the same as the V243-VA91 except it has been modified as follows:

Added front panel connectors (BNC/LEMO/SMB) to bring a differential voltage from an external calibration source into the module.

March 31, 2000

*****Special Option*****

Model V243-S003

96-channel, Low-level Signal Conditioner

June 26, 2000

Special Option

Model V243-S003

The Model V243-S003 is the same as the V243-VA91 except that the filter cutoffs are 2Hz, 10Hz and 50Hz rather than 10Hz, 50Hz and 500Hz.

June 26, 2000

CONTENTS

FIGURES

TABLES

96-channel, Low-level Signal Conditioner

Provides exceptionally high accuracy at low input levels

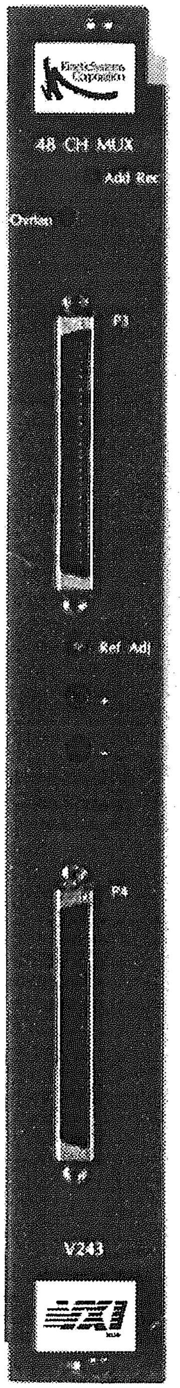
V243

Features

- Use with V208 host ADC
- 16, 32, 48, 64 or 96-channel input options
- Differential inputs and programmable gain per channel
- 2-pole, active Butterworth, low-pass filters with programmable cutoff: 10, 50, 500 Hz, or 5 kHz bypass
- Open-thermocouple detection option available
- End-to-end channel calibration

Typical Applications

- Automotive body engineering tests
- Automotive powertrain testing
- Automotive safety tests
- Aircraft engine testing
- Rocket engine testing
- Satellite testing



General Description *(Product specifications and descriptions subject to change without notice.)*

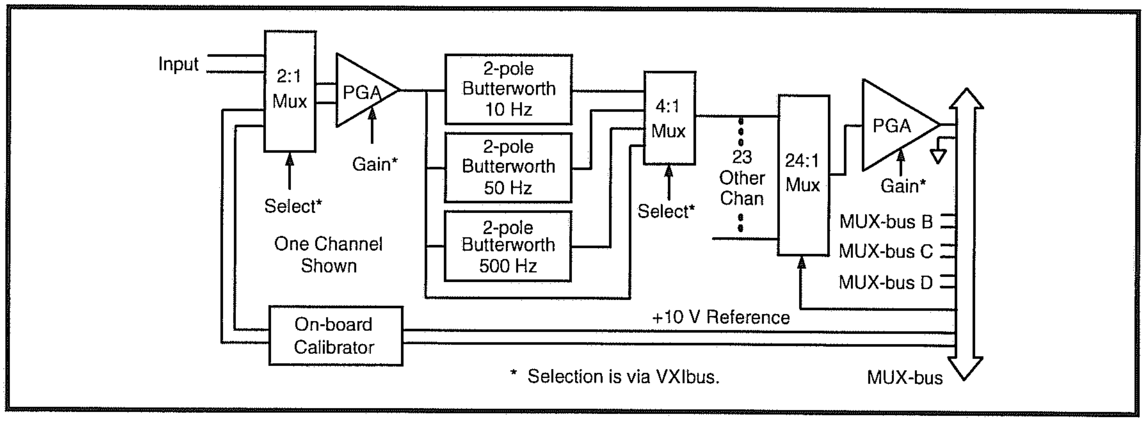
The Model V243 is a single-width, C-size, register-based, VXIbus module that is designed for thermocouple and other low-bandwidth, low-level signal applications. It provides up to 96 differential-input channels with programmable gain per channel and 2-pole, active, low-pass Butterworth filters on each channel. Filter cutoff frequencies of 10, 50 and 500 Hz plus bypass are software-selectable. Software-selectable pre-filter gains of 1, 10, and 100 are provided for each channel. Both filter cutoff and pre-filter gains are programmable in groups of eight channels. A common, multiplexed, post-filter gain of 1, 2, 5, 10, or 20 is selectable on a channel by channel basis. This allows a programmable gain per channel from 1 to 2000 (±10 V to ±5 mV full scale range).

The V243 is designed to work with the V208 16-bit, 100,000 Sample/second ADC module which supports up to 2048 input channels using MUX-bus™. Up to 11 V243s can be placed in adjacent slots to the right of the V208 to provide a full 1056 active input channels.

From one to eleven V243s combined with the V208 provide a low-noise analog subsystem with built-in, per-channel calibration that is traceable to NIST standards. The V208 includes a precision reference source that is calibrated at the factory, and each V243 includes a precision calibrator. Each input channel can be connected to the calibrator output, ground, or the analog input under software control. In addition, for maximum accuracy, a small correction factor for each calibrator output is stored in a EEPROM within each V243. This correction factor can be applied during the calibration process to obtain maximum accuracy.

The V243 supports both static and dynamic configuration. It may be accessed using A24/A16, D16 data

V243 Block Diagram (one channel shown)



Item	Specification		
General			
Number of channels	16, 32, 48, 64 or 96, differential input		
Gain ranges	1, 2, 5, 10, 20, 50, 100, 200, 500, 1000, 2000		
Cutoff frequencies	10 Hz, 50 Hz, 500 Hz, and Bypass		
Input			
Input range			
Common mode:	±10.24 V		
Differential:	±10.24 V		
Input protection	±35 V continuous		
Input impedance			
ZAxx Options:	>100 MΩ		
VAxx Options:	10 MΩ		
Input impedance (Over temperature)			
ZAxx Options:	20 MΩ min; 100 MΩ typical		
VAxx Options	10 MΩ		
Transfer Characteristics			
Linearity error	±0.0015% FSR		
Initial accuracy, RTI*	After automatic calibration using a V208:		
	ZAxx Options	VAxx Options	
Gain = 1	±(900 µV + 0.01% of reading)	±(900 µV + 0.04% of reading)	
Gain = 10	±(90 µV + 0.01% of reading)	±(90 µV + 0.04% of reading)	
Gain = 100	±(9 µV + 0.01% of reading)	±(9 µV + 0.04% of reading)	
Gain = 2000	±(2 µV + 0.04% of reading)	±(2 µV + 0.04% of reading)	
Offset stability, RTI	0.5 µV/°C @ gain ≥ 1000		
Gain stability	15 ppm/°C		
Bandwidth	5 kHz		
Common mode rejection	-120 dB typical, -110 dB min at gain ≥ 100		
Noise, RTI	1 µV RMS typical, gain ≥ 1000		
Channel-to-channel crosstalk, RTI			
Gain = 1 to Gain = 1	90 dB		
Gain = 1 to Gain = 2000	135 dB		
I/O Connector Type	68P High Density (24 channels per connector)		
Power Requirements (quiescent)	+5V	+24V	-24V
V243-VA11	1.8 A	220 mA	190 mA
V243-VA31	1.8 A	360 mA	300 mA
V243-VA41, V243 -ZA11	1.8 A	460 mA	370 mA
V243-VA61	2.0 A	650 mA	520 mA
V243-VA91, V243-ZA21	2.0 A	850 mA	670 mA
Environmental and Mechanical			
Temperature range			
Operational	0°C to + 50°C		
Storage	-25°C to + 75°C		
Relative humidity	0 to 85%, non-condensing to 40°C		
Cooling requirements	10 CFM		
Dimensions	340 mm x 233.35 mm x 30.48 mm (C-size VXIbus)		
Front-panel potential	Chassis ground		

** Includes V208 quantization error.*

Ordering Information

Model V243-ZA11	48-channel, Low-level Multiplexer
Model V243-ZA21	96-channel, Low-level Multiplexer
Model V243-VA11	16-channel, Low-level Multiplexer with Open Thermocouple Detection
Model V243-VA31	32-channel, Low-level Multiplexer with Open Thermocouple Detection
Model V243-VA41	48-channel, Low-level Multiplexer with Open Thermocouple Detection
Model V243-VA61	64-channel, Low-level Multiplexer with Open Thermocouple Detection
Model V243-VA91	96-channel, Low-level Multiplexer with Open Thermocouple Detection

Please note: The V208 must be used as the host ADC for the V243 (not the V207).

Related Products

Model V208	16-bit, 100,000 Sample/second ADC Subsystem
Model 5868-Bxyz	Cable—68S High Density to Unterminated
Model 5868-Dxyz	Cable—68S High Density to 68P High Density
Model V750-ZB11	Termination Assembly for 48 Channel (V243-VA41)
Model V765-ZA11	Rack-mount Termination Panel
Model V765-ZB11	Rack-mount Termination Panel with Improved CMR
Model V792-ZA11	Rack-mount Isothermal Termination Panel
Model V792-ZB11	Rack-mount Isothermal Termination Panel with Improved CMR

Unpacking and Installation

At KineticSystems, static precautions are observed from production, test, and packaging of the module. This includes using static proof mats and wrist straps. Please observe these same precautions when unpacking and installing the module whenever possible.

The Model V243 is shipped in an anti-static bag within a styrofoam packing container. Carefully remove the module from its static-proof bag and prepare to set the logical address switches to the appropriate value.

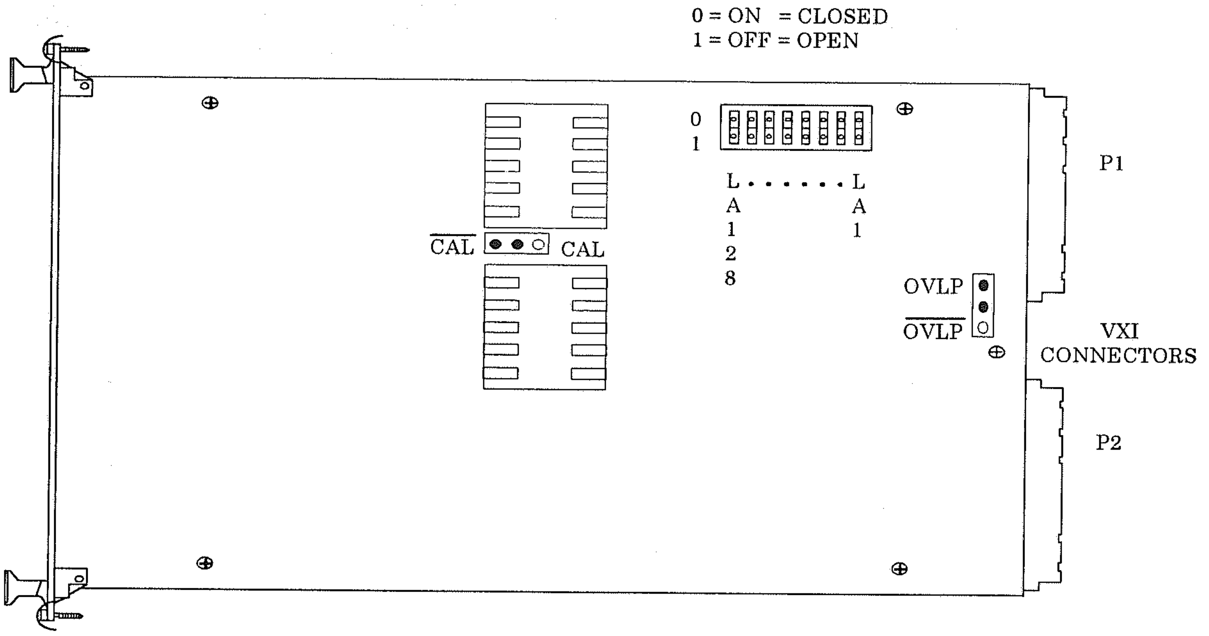


FIGURE 1 - V243 Switch Locations

Configuration

There is one set of user configurable switches on the V243. All eight switches, #1 (MSB) to #8 (LSB), are for the logical address. The logical address may be set from 1 to 254 as a statically configured device. If the module is set for logical address 255 (all switches open), then the V243 will be dynamically configured by the resource manager. A logical address of 0 is not valid since it is reserved for the Slot 0 controller. Logical address 255 is the factory default setting.

A three position strap is located between two heatsinks. This strap is used to enable write access to a correction table used during the module (factory) calibration. The V243 can be recalibrated if this strap is moved to the enabled position (right). The factory default for the cal strap is to the disabled position to protect accidental erasure of the correction table.

There is also a three position strap located between the P1 and P2 connectors. A strap may be placed on the upper two strap posts to enable this modules overlap detect capability (factory default). When enabled on all of the signal conditioning modules, the V243 will be able to detect if multiple modules are attempting to output a voltage out on MUX-bus at the same time. An overlap may occur if a module is incorrectly programmed. More detail is provided on page 12 about overlap detect and how it relates to MUX-bus.

Module Insertion

The V243 is a C-sized, single width, VXIbus module. Except for Slot 0 or Slot 1, it can be mounted in any unoccupied slot in a C-size VXIbus mainframe. Since the V243 is a signal conditioning module that uses MUX-bus, this module must be plugged into the VXI mainframe to the right of the V208 or another signal conditioning module in order to use MUX-bus. MUX-bus is a proprietary local bus that the signal conditioning modules (like the V243) use to send voltages to the V208.

CAUTION: TURN MAINFRAME POWER OFF WHEN INSERTING OR REMOVING MODULE

WARNING: REMEMBER TO REMOVE INTERRUPT ACKNOWLEDGE DAISY-CHAIN JUMPERS PRIOR TO INSERTING THIS MODULE IN THE BACKPLANE OR USE AN AUTOCONFIGURING BACKPLANE

If the mainframe does not have an autoconfiguring backplane, special care is necessary when installing the V243. To insure proper interrupt acknowledge cycles from the V243 module, the daisy-chain Interrupt Acknowledge jumper must be removed before the module is installed in a slot. Conversely, daisy-chain jumpers must be installed in any empty slot between the V243 and the Slot 0 Controller.

Front Panel Description

LEDs

The Add Rec (address received) LED lights to indicate the V243 is being accessed through VXI. The Overlap (overlap detect) LED will light to indicate that another module tried to output its voltage out on the MUX-bus at the same time the V243 did. If the overlap detect LED is on, the V243 will not output any voltages out on MUX-bus.

Connectors

The V243 has two or four 68 pin, high density SCSI type connectors depending on the option. Each connector has 24 differential inputs as well as a buffered output of the first channel on that connector, a calibration output used during module calibration, and an isothermal reference input. See Figure 4 and Table 4 (Pages 38 and 39) for the precise pinout descriptions. The on board, 10 volt reference can be calibrated by adjusting the voltage through the pot labeled Ref Adj (Reference Adjust) on the front panel. The "on board" reference voltage can be monitored through the connectors labeled "+" and "-".

V750-Isothermal Terminal Assembly

The V750-ZB11 is an isothermal termination assembly specifically designed for the V243. This assembly is capable terminating up to 48 channels. The isothermal reference is connected to the isothermal channel located on the J3 connector. This isothermal channel may be entered into the V243 scan list as channel 97. A detailed explanation on the use of this reference is on page 11.

In cases where the V243 is used in applications other than with thermocouples, the V750 can be used as a general purpose termination assembly. The reference is simply not used to measure the temperature of the assembly.

To remove the cover of the V750 pry at points 1 and 2 with a screwdriver until the bottom of the cover is clear of the front side of the termination assembly. The cover should easily lift off the assembly.

To put the cover back on, insert the pegs at points 3 and 4. Then push down at points 1 and 2 until the latches in front are hooked in the slots.

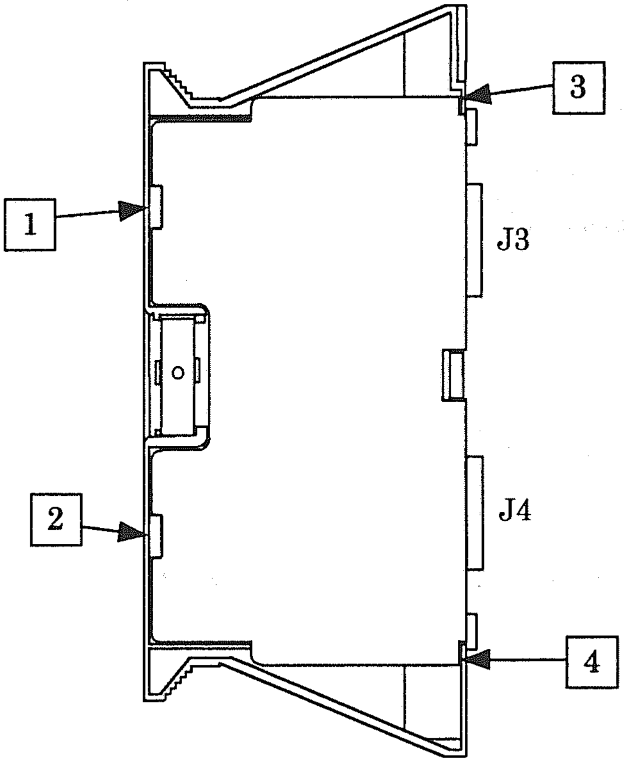


FIGURE 2 - V750- ZB11

Programming Information

VXIbus Addressing

The V243 is classified as an extended register device which means it has registers that occupy A16 and A24 space. In accordance with the VXI specification, A16 space means that only 16 address bits are decoded by the V243. A16 registers occupy the address space from 0 to 65535. A24 space means that 24 address bits are decoded by the V243. A24 registers occupy the address space from 0 to 16777215.

The configuration registers are located in A16 space and include the standard registers defined by VXI as well as additional, general purpose registers. From these registers, information about the specific module can be read, the base address for the A24 registers can be controlled, and the interrupt level can be set. For example, by reading certain registers in A16 space, the following information about this module can be found just by knowing the module's logical address:

Manufacturer:	KineticSystems
Module Type:	V243
Option:	ZA21
Serial Number:	20
Firmware Version:	1.0
Hardware Version:	1.0

In general, any configuration register can be accessed simply by knowing the module's logical address (set by the logical address switches) and the register's offset. A complete list of A16 register descriptions and their offsets is available starting at pages D-1 and 24.

$$A16\_ADDRESS = C000_{16} + (LOGICAL\_ADDRESS \bullet 40_{16}) + A16\_REGISTER\_OFFSET$$

Manufacturer: can be found by looking at bits 11 through 0 at register offset 0_{16}. If the value is $F29_{16}$, this indicates that the module was built by KineticSystems.

Module Type: can be found by looking at bits 11 through 0 at register offset 2_{16}. If the value is a BCD (Binary Coded Decimal) number 243, this indicates that the module is a V243.

Option: can be found by reading the four character string located at registers at offsets 20_{16} and 22_{16}. If the strings "ZA" is found at offset 20_{16} and "21" is found at offset 22_{16}, then the specific type of module is the V243-ZA21 meaning the this module does not have open thermocouple detection and has 96 channels. The suffix registers are additional registers defined by KineticSystems to aid in identifying a module. It is not a standard register defined by the VXI spec.

Serial Number: can be found reading the 32-bit, unsigned number stored in registers at offsets A_{16} and C_{16}. For example, if a the serial number was 20, the values in the Serial

Number High Register would be 0_{16} and the value in the Serial Number Low Register would be 14_{16}.

Firmware Version: can be found by reading the BCD number stored in bits 15 through 8 of the Version Number Register at offset E_{16}.

Hardware Version: can be found by reading the BCD number stored in bits 7 through 0 of the Version Number Register at offset E_{16}.

In addition to the Suffix Registers, KineticSystems has left registers at offsets 24_{16} through $3E_{16}$ open for user definition. These User Defined Registers can also be used to identify the module (i.e., with an internal identification number). These registers can only be written one at a time and only at 3 mS intervals, but once they have been written, the data is retained even after power has been removed.

The operational registers are located in A24 space and include the registers specific to V243 modules. This address space is configured by the resource manager. These registers include channel setup and calibration registers. In general, any operational register can be accessed simply by knowing the value in the modules Offset Register (Operational register in A24 space at offset 6).

$$A24\_ADDRESS = (OFFSET\_REGISTER\_VALUE \bullet 100_{16}) + A24\_REGISTER\_OFFSET$$

Prefilter Gain Registers (page 34) - Controllable in groups of eight channels.

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	IMUX					FLTR 1	FLTR 0							FGN1	FGN0
(1)	(0)	(1)	(1)	(1)	(1)	(0)	(0)	(1)	(1)	(1)	(1)	(1)	(1)	(0)	(0)

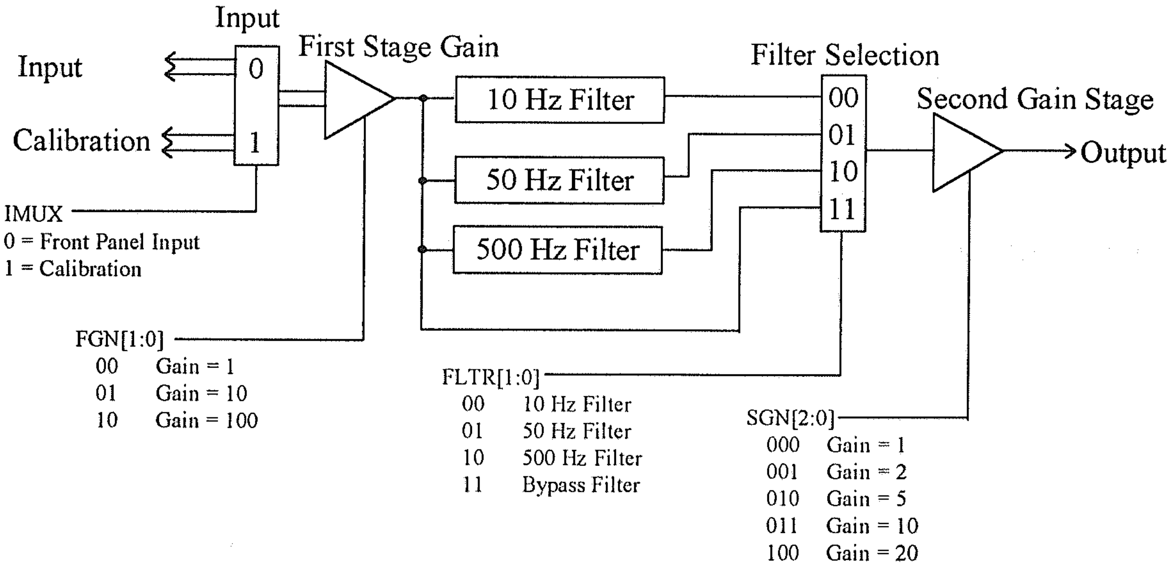
( ) Power Up Value

Post Gain Registers (page 35) - Controllable on a per channels basis.

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
													SGN2	SGN1	SGN0
(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(?)	(?)	(?)	(?)	(?)	(?)	(?)	(?)

( ) Power Up Value

FIGURE 3 - Channel Setup



Channel Setup

The V243 has two separate gain stages. The first gain stage is applied before the filter and is controllable in groups of eight channels. First stage gain can be changed by writing bits 1 and 0 of the Prefilter Gain Register. First stage gain may be set to 1, 10, or 100. The second stage gain is applied after the filter and is controllable on a per channel basis. Second stage gain is set by writing bits 2 to 0 of the Postfilter Gain Register. Second stage gain may be set to 1, 2, 5, 10, or 20.

In general, it is best to apply the most possible gain in the first stage. For example, a gain of 10 can be accomplished by applying the gain of 10 in the first or second stage. In this case, a gain of 10 should be applied in the first stage for best noise immunity.

During normal operation, the input of each channel can be selected to connect to the 68 pin, high density, SCSI II type connector by setting bit 14 of the Prefilter Gain Register to zero. Setting this bit to a one will set the channel to the calibration signal. During calibration, each channel can be selected to connect to the calibration voltage. The calibration voltage is set by writing to the Calibration Register (Operation register in A24 space at offset 2_{16})

Isothermal Channels

In addition to the analog input channels, the V243 can have up to four isothermal reference channels, one for each connector. When the V243 is connected to an isothermal panel or assembly, these reference channels will output voltage 100 mV/°C. These reference channels are considered channels 97 through 100 when setting up MUX-bus. Although these channels go through the same second stage gain (post-filter) and MUX-bus paths as the other channels, the reference must always have a gain of 1 and does not get calibrated along with the other channels. Since the second stage gain can be applied on a pre channel basis, setting the gain to one for the reference channel will not restrict the gain settings for any one channel.

Open Thermocouple Detection

The V243-ZB11, V243 ZB21, and the V243-VAxx options have the capability to detect open thermocouples. The detection is enabled by setting bit 7 of the Configuration Register (Operational Register in A24 space offset , 0_{16} page 32). Once enabled, any channel that is open will be set to positive full scale, but any other channel will remain in its normal range. Because current is injected into each channel while the open thermocouple detection is enabled, the channel's accuracy will be affected. Therefore, when detection is enabled, no channels may be calibrated nor can valid data be taken.

MUX-bus

The MUX-bus makes use of the VXI P2 LBUS (local bus) connector lines. Signal conditioning modules supporting the MUX-bus must be plugged into a slot to the right of the V208 or another signal conditioning module.

The MUX-bus pin definition (P2) is:

LBUS00	MUX path A signal hi
LBUS01	MUX path A signal lo
LBUS02	MUX path B signal hi
LBUS03	MUX path B signal lo
LBUS04	MUX path C signal hi
LBUS05	MUX path C signal lo
LBUS06	MUX path D signal hi
LBUS07	MUX path D signal lo
LBUS08	Reference Hi
LBUS09	Reference Lo
LBUS10	Overlap Detect
LBUS11	MUX-bus Clock

The MUX-bus clock generates the control signal that causes the output multiplexers on the signal conditioning modules to increment to the next channel.

Reference Hi and Reference Lo provide a +10 volt differential signal to adjacent signal conditioning modules, allowing all channels to be referenced to a common standard. The entire system can be calibrated to this reference under software control. This greatly simplifies system calibration.

Overlap Detect is used by the signal conditioning modules to detect if two or more modules are attempting to output their voltages at the same time. This helps prevent conflicts caused by an incorrectly programmed Scan RAM (Scan RAM is used to control MUX-bus). Should an overlap condition occur, the V243 will stop outputting any voltages out on MUX-bus.

Scan RAM

> NOTE: As a general rule, the V208 or any other receiving MUX module, should be the FIRST to be put in "setup" mode and the LAST to be put in "run" mode.

Table 1 shows an example of how Scan RAM might be setup in a case using a V243, V252, and a V208. In the example, 16 channels of V243 are assigned to the first 16 channels of V208. Eight Channels of the V252 (16 channel filter module) is then assigned to the next eight channels on the V208 for a total of 24 channels.

TABLE 1 - Scan RAM / Scan Table Example

Scan Table Pointer	V243 Scan Table Data	V252 Scan Table Data	V208 Scan Table Data	Channel Enabled
$2E_{16}$	$1000\ 0000\ 0000\ 0111_2$	$1100\ 0000\ 0000\ 0111_2$	$1000\ 0000\ 0000\ 0111_2$	V252 Channel 8
$2C_{16}$	$0000\ 0000\ 0000\ 0110_2$	$0100\ 0000\ 0000\ 0110_2$	$0000\ 0000\ 0000\ 0110_2$	V252 Channel 7
$2A_{16}$	$0000\ 0000\ 0000\ 0101_2$	$0100\ 0000\ 0000\ 0101_2$	$0000\ 0000\ 0000\ 0101_2$	V252 Channel 6
28_{16}	$0000\ 0000\ 0000\ 0100_2$	$0100\ 0000\ 0000\ 0100_2$	$0000\ 0000\ 0000\ 0100_2$	V252 Channel 5
26_{16}	$0000\ 0000\ 0000\ 0011_2$	$0100\ 0000\ 0000\ 0011_2$	$0000\ 0000\ 0000\ 0011_2$	V252 Channel 4
24_{16}	$0000\ 0000\ 0000\ 0010_2$	$0100\ 0000\ 0000\ 0010_2$	$0000\ 0000\ 0000\ 0010_2$	V252 Channel 3
22_{16}	$0000\ 0000\ 0000\ 0001_2$	$0100\ 0000\ 0000\ 0001_2$	$0000\ 0000\ 0000\ 0001_2$	V252 Channel 2
20_{16}	$0000\ 0000\ 0000\ 0000_2$	$0100\ 0000\ 0000\ 0000_2$	$0000\ 0000\ 0000\ 0000_2$	V252 Channel 1
$1E_{16}$	$0100\ 0000\ 0000\ 1111_2$	$0000\ 0000\ 0000\ 1111_2$	$0000\ 0000\ 0000\ 1111_2$	V243 Channel 16
$1C_{16}$	$0100\ 0000\ 0000\ 1110_2$	$0000\ 0000\ 0000\ 1110_2$	$0000\ 0000\ 0000\ 1110_2$	V243 Channel 15
$1A_{16}$	$0100\ 0000\ 0000\ 1101_2$	$0000\ 0000\ 0000\ 1101_2$	$0000\ 0000\ 0000\ 1101_2$	V243 Channel 14
18_{16}	$0100\ 0000\ 0000\ 1100_2$	$0000\ 0000\ 0000\ 1100_2$	$0000\ 0000\ 0000\ 1100_2$	V243 Channel 13
16_{16}	$0100\ 0000\ 0000\ 1011_2$	$0000\ 0000\ 0000\ 1011_2$	$0000\ 0000\ 0000\ 1011_2$	V243 Channel 12
14_{16}	$0100\ 0000\ 0000\ 1010_2$	$0000\ 0000\ 0000\ 1010_2$	$0000\ 0000\ 0000\ 1010_2$	V243 Channel 11
12_{16}	$0100\ 0000\ 0000\ 1001_2$	$0000\ 0000\ 0000\ 1001_2$	$0000\ 0000\ 0000\ 1001_2$	V243 Channel 10
10_{16}	$0100\ 0000\ 0000\ 1000_2$	$0000\ 0000\ 0000\ 1000_2$	$0000\ 0000\ 0000\ 1000_2$	V243 Channel 9
E_{16}	$0100\ 0000\ 0000\ 0111_2$	$0000\ 0000\ 0000\ 0111_2$	$0000\ 0000\ 0000\ 0111_2$	V243 Channel 8
C_{16}	$0100\ 0000\ 0000\ 0110_2$	$0000\ 0000\ 0000\ 0110_2$	$0000\ 0000\ 0000\ 0110_2$	V243 Channel 7
A_{16}	$0100\ 0000\ 0000\ 0101_2$	$0000\ 0000\ 0000\ 0101_2$	$0000\ 0000\ 0000\ 0101_2$	V243 Channel 6
8_{16}	$0100\ 0000\ 0000\ 0100_2$	$0000\ 0000\ 0000\ 0100_2$	$0000\ 0000\ 0000\ 0100_2$	V243 Channel 5
6_{16}	$0100\ 0000\ 0000\ 0011_2$	$0000\ 0000\ 0000\ 0011_2$	$0000\ 0000\ 0000\ 0011_2$	V243 Channel 4
4_{16}	$0100\ 0000\ 0000\ 0010_2$	$0000\ 0000\ 0000\ 0010_2$	$0000\ 0000\ 0000\ 0010_2$	V243 Channel 3
2_{16}	$0100\ 0000\ 0000\ 0001_2$	$0000\ 0000\ 0000\ 0001_2$	$0000\ 0000\ 0000\ 0001_2$	V243 Channel 2
0_{16}	$0100\ 0000\ 0000\ 0000_2$	$0000\ 0000\ 0000\ 0000_2$	$0000\ 0000\ 0000\ 0000_2$	V243 Channel 1

Please Note: The Scan Tables for each MUX-bus module are identical except only one module has the enable bit set at any one time. If this rule is not observed, an overlap condition will occur because two modules will try to output voltages on MUX-bus at the same time.

V243 and V252 Scan RAM (starts at A24 offset 200_{16})

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
EOL	Enabl								Channel Number						
(?)	(?)	(?)	(?)	(?)	(?)	(?)	(?)	(?)							

( ) Power Up Value

V208 Scan RAM (starts at A32 offset 1000_{16})

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
EOL (?)	Enabl (?)	(?)	(?)	(?)	Channel Number										

( ) Power Up Value

For this example:

MUX-bus on the V208 always scans at 200 kHz (5 μS convert time)

If there are 24 channels in the Scan Table and since the V208 is a scanning ADC, then the scan rate must be less than or equal to 200 kHz / 24 channels = 8333 Hz. The scan rate determines how often the MUX-bus modules run through the Scan Table.

Let scan rate = 8 kHz, t_{SCAN} = 1 / 8 kHz = 125 μS.

Every 125 μS:

1.) Scan Table Pointer is set to 0_{16}.
V243 reads 4000_{16}. Enable bit is set. Channel 1 is indicated.
V243 outputs its channel 1 voltage.
V252 reads 0000_{16}. Enable bit is not set.
V252 waits.
V208 reads 0000_{16}.
V208 reads its channel 1 voltage.

2.) Scan Table Pointer is set to 2_{16}.
V243 reads 4001_{16}. Enable bit is set. Channel 2 is indicated.
V243 outputs its channel 2 voltage.
V252 reads 0001_{16}. Enable bit is not set.
V252 waits.
V208 reads 0001_{16}.
V208 reads its channel 2 voltage.

.

.

17.) Scan Table Pointer is set to $2E_{16}$.

.

.

.

24.) Scan Table Pointer is set to $2E_{16}$.
V243 reads 8007_{16}. End of list bit is set. Enable bit is not set.
V243 waits for next tick of MUX-bus clock to start at beginning of Scan Table
V252 reads $C007_{16}$. End of list bit is set. Enable bit is set. Channel 8 is indicated.
V252 outputs its channel 8 voltage, and waits for next tick of MUX-bus clock to start at beginning of Scan Table.
V208 reads 8007_{16}. End of list bit is set.
V208 reads its channel 24 voltage and waits for next tick of MUX-bus clock to start at beginning of Scan Table.

The Scan RAM contains the Scan Table that must be setup properly on all MUX-bus modules. The Scan RAM is 2048 words long, and the Scan Table is a subset of the Scan RAM. Each module scans through its Scan Table sequentially at 200 Khz. At any time, the current data in the list dictates which module is to connect a channel to the MUX-bus.

The following is a list of steps to follow when setting up MUX-bus. There is also a programming example of this procedure called "setup()" in Appendix B (page B6).

Step 1.) Get access to Scan RAM.

Before the Scan Table can be written into Scan RAM, the V243 and the V208 must be put in "setup" mode by resetting the ADSEL bit in the Configuration Register. Once a module is put in "setup" mode, MUX-bus is halted and access is allowed to that module's Scan RAM. If access to the Scan RAM is attempted while in "run" mode, the V243 will indicate it was unable to comply by asserting a bus error. As a general rule, the V208 or any other receiving MUX-bus module, should be the FIRST to be put in "setup" mode, and the LAST to be put in "run" mode.

Step 2.) Write Scan Table into Scan RAM.

The Scan Table consists of the enable bit, a channel number, and an end of list bit. The Scan Table is organized the same on all modules whether it sources or receives voltages on the MUX-bus.

Bit 14 is the enable bit that connects the specified channel to MUX-bus. The enable bit must be set on only one sourcing module at any one time in the Scan Table (only one voltage may be out on MUX-bus at one time). If this rule is not implemented correctly, an overlap condition will occur once MUX-bus is started.

The channel is specified by the first 7 bits of Scan RAM. However, not all 100 channels (96 differential signals plus 4 isothermal references) of the V243 are available at any one time. The MUX-bus has four analog paths (labeled A through D) and is distributed evenly across all 100 channels of the V243. MUX-bus has these four distinct paths to increase bandwidth and lengthen settling time for better overall accuracy. The restriction, however, is that not all channels are available at any time. The very first channel in the Scan Table must be available on Path A, and the next channel from Path B, and so on. The Scan Table must continue to follow this A →B →C →D order.

Here is the list of the paths and what channel is available from each. Please notice that the channels available on a path increment by four (the number of paths). These channels are distributed the same on all signal conditioning modules using MUX-bus.

Path A	Channels 1,5,9,13,17,...97 are available.	Equation: 4N + 1
Path B	Channels 2,6,10,14,18,...98 are available.	Equation: 4N + 2
Path C	Channels 3,7,11,15,19,...99 are available.	Equation: 4N + 3
Path D	Channels 4,8,12,16,20,..100 are available.	Equation: 4N + 4
		where N = 0 to 23

Bit 15 is the end of list bit. This bit must be set on the last location of the Scan Table on all MUX-bus modules.

Step 3.) Startup MUX-bus.

Finally, all modules using MUX-bus should be put in "run" mode starting with the signal conditioning modules (V243, V252, etc) first. The V208 must be the last module to be put in "run" mode.

MUX-bus Rules

Should an overlap condition occur on a MUX-bus module, most likely one of the following rules were not implemented properly.

Rule #1.) The V208 must be the FIRST to be put in "setup" mode and the LAST to be put in "run" mode. This ensures that all MUX-bus modules remain in sync.

Rule #2.) The Enable bit should be set in only ONE module and any given Scan Table location. This ensures that only one module outputs a channel out on MUX-bus at one time.

Rule #3.) The End of List bit should be set in EVERY module at the same Scan Table location. All MUX-bus modules must have the same Scan Table size. This ensures modules remain in sync.

Rule #4.) The channels in the Scan Table must follow the MUX-bus path order. The Scan Table must continue to follow this A →B →C →D order. (see Table 2 on page 17 for a more detailed explanation)

Overlap Detect Checklist

Should an overlap condition occur on a MUX-bus module, this detailed list may help in finding the specific cause of the overlap condition.

Cause #1.) If an overlap occurred on only 1 MUX-bus module, then that module was programmed to output a channel at an inappropriate time. To increase bandwidth of MUX-bus and improve accuracy, MUX-bus has 4 distinct paths (A through D). However, any given channel on any given module is available only to one path. MUX-bus must run in order from Path A through to Path D. The following table gives an example of a properly programmed Scan Table.

TABLE 2 - V243 Scan Table, MUX-bus Path

A24 Offset	MUX-bus Path	Available Channels	Data	Channel Enabled
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
20E$_{16}$	Path D	4,8,12,16,...100	0100 0000 0000 0111$_2$	V243 Channel 8
20C$_{16}$	Path C	3,7,11,15,...99	0100 0000 0000 0110$_2$	V243 Channel 7
20A$_{16}$	Path B	2,6,10,14,...98	0100 0000 0000 0101$_2$	V243 Channel 6
208$_{16}$	Path A	1,5,9,13,...97	0100 0000 0000 0100$_2$	V243 Channel 5
206$_{16}$	Path D	4,8,12,16,...100	0100 0000 0000 0011$_2$	V243 Channel 4
204$_{16}$	Path C	3,7,11,15,...99	0100 0000 0000 0010$_2$	V243 Channel 3
202$_{16}$	Path B	2,6,10,14,...98	0100 0000 0000 0001$_2$	V243 Channel 2
200$_{16}$	Path A	1,5,9,13,...97	0100 0000 0000 0000$_2$	V243 Channel 1

Cause #2.) If an overlap occurred on two MUX-bus modules, then one of two possibilities caused the overlap. One possibility is that both modules were programmed to output a channel at an inappropriate time (see Cause #1). The other possibility is that two MUX-bus module tried to output voltages at the same time. This would occur if the enable bit in the Scan Table was set at the same location on those two modules.

Cause #3.) If an overlap occurred on multiple, but not all, MUX-bus modules, then most likely a combination of Causes #1 and #2 occurred.

Cause #4.)If an overlap occurred on all MUX-bus modules, then one of two possibilities caused the overlap. One possibility is that the V208 was not the last to be put in "run" mode. The other possibility is that the End of List bit was not set at the same location in each of their Scan Tables.

TABLE 3 - Calibration Register Values

Cal Voltage	On board Source	MUX bus Source
+10.0 V	8091_{16}	0091_{16}
+5.0 V	$80A1_{16}$	$00A1_{16}$
+2.0 V	$80C1_{16}$	$00C1_{16}$
+1.0 V	8092_{16}	0092_{16}
+0.5 V	$80A2_{16}$	$00A2_{16}$
+0.2 V	$80C2_{16}$	$00C2_{16}$
+0.1 V	8094_{16}	0094_{16}
+0.05 V	$80A4_{16}$	$00A4_{16}$
+0.02 V	$80C4_{16}$	$00C4_{16}$
+0.01 V	8098_{16}	0098_{16}
+0.005 V	$80A8_{16}$	$00A8_{16}$
+0.002 V	$80C8_{16}$	$00C8_{16}$
-0.002 V	8148_{16}	0148_{16}
-0.005 V	8128_{16}	0128_{16}
-0.01 V	8118_{16}	0118_{16}
-0.02 V	8144_{16}	0144_{16}
-0.05 V	8124_{16}	0124_{16}
-0.1 V	8114_{16}	0114_{16}
-0.2 V	8142_{16}	0142_{16}
-0.5 V	8122_{16}	0122_{16}
-1.0 V	8112_{16}	0112_{16}
-2.0 V	8141_{16}	0141_{16}
-5.0 V	8121_{16}	0121_{16}
-10.0 V	8111_{16}	0111_{16}

Note: For best results, use the on board source for the calibration voltages. The module is calibrated using its on board source.

Module Calibration

In order to achieve the high degree of accuracy of the V243, error coefficients for both offset and gain are stored in the Correction Table (non-volatile RAM) on the V243.

Coefficients are stored for the calibrator to compensate for any gain error. These coefficients are applied in software during channel calibration to compute the true voltage applied to a channel during gain calibration. These coefficients are stored in the Correction Table (page 37) starting at A24 offset 2010_{16}.

Coefficients are also stored to compensate for channel to channel offset errors. These coefficients are applied during channel offset calibration and are stored starting at A24 offset 2040_{16}.

It is recommended that this calibration procedure be performed approximately every six months. The calibration date is also stored in the Correction Table to help track the last calibration date. The date is stored at the beginning of the Correction Table starting at A24 offset 2000_{16}.

The following is a list of steps to follow in performing the V243 module calibration. There is also a programming example in Appendix B which performs the module calibration called "Module.c" (page B11). The documented code example will refer directly to the following steps:

1. The CAL strap must be moved to the enabled position (page 4).
2. The V243 should have a minimum of a 30 minute warm-up period.
3. Connect CAL OUT (pins 32 and 66 of any front panel connector) to a precision voltage meter.
4. Set all channels of the V243 to the front panel connectors. This can be done by writing all Prefilter Gain Registers with a 0.
5. Gain Calibration - repeat steps 5a to 5c for all 12 differential, calibrator voltages. In the following equation, these expected calibrator voltages are referred to as CALIBRATOR ranging from 10 volts to 2 mV. The calibrator voltages are set by writing to the Calibration Register (Operational register in A24 space at offset 2_{16}). A table of the precise register setting is available on page 17. Make sure to also set bits 15 and 13 of the Calibration Register to use the internal voltage reference and to enable calibration voltage out of the front panel connectors. Also, make sure to wait for the calibrator voltage to settle before any measurements are taken.
   a) Use a precision meter to measure the positive calibration voltage (average 10 points). The measured positive calibration voltage is referred to as PCAL in the equation from step 5c.
   b) Use a precision meter to measure the negative calibration voltage (average 10 points). The measured negative calibration voltage is referred to as NCAL in the equation from step 5c.

c) Store the error coefficient into the Correction Table (page 37) as a 16-bit signed integer according to the following equation:

$$GAINCOEFFICIENT = \left(\frac{PCAL - NCAL}{CALIBRATOR \bullet 2} - 1\right) 10^6$$

Please note: Must wait a minimum of 3 ms after writing coefficient before writing again.

6. If offset calibration is unnecessary, skip to step #8.
7. Offset Calibration - for every channel repeat steps 7a through 7g.
   a) Connect all channels to the front panel connectors with 10 Hz filters and First Stage Gain set to 100. This can be done by writing all Prefilter Gain Registers with a 2_{16}. Wait for filters to settle to 16-bit accuracy.
   b) Set the second stage gain for all channels to a gain of 20. This can be done by writing 4 to all Post Gain Registers.
   c) Measure the front panel offset voltage (RT0) using the V208. Use this equation to convert the COUNTS read from the V208 into voltage RTO. This front panel ground measurement is referred to as FP OFFSET in the equation in step 7g.

$$VOLTAGE(RTO) = (COUNTS - 32768) \bullet 312.5\mu Volts$$

   d) Connect all channels to the calibrator. The channels should still have the 10 Hz filters selected and the First Stage Gain set to 100. This can be done by writing all Prefilter Gain Registers with a 4002_{16}.
   e) Set the calibrator voltage to ground and wait for filters to settle to 16-bit accuracy.
   f) Measure the calibrator offset voltage (RT0) using the equation from step 7c. The calibrator ground measurement is referred to as CAL OFFSET in the equation in step 7g.
   g) Store the offset error coefficient into the Correction Table (page 37) as a 16-bit signed integer using the following equation:

$$OFFSETCOEFFICIENT = \left(\frac{FPOFFSET - CALOFFSET}{2000}\right) 10^9$$

Please note: Must wait a minimum of 3 ms after writing each coefficient before writing again.

8. Store calibration date into the Correction Table (page 37).
   Please note: Must wait a minimum of 3 ms after writing each part of the date before writing again.

Channel Calibration

Once the module has been calibrated, each channel on the V243 can be calibrated using the onboard calibrator all under software control. Offset and gain error coefficients for each channel must be stored in software and be applied to all voltage readings from those channels. It is important not to confuse these error coefficients in software with those stored in the Correction Table on the V243. The error coefficients in the Correction Table are used during the channel calibration only. The following is a list of steps to follow in performing the V243 channel calibration:

1. The V243 should be allowed at least a 30 minute warm-up period.
2. Setup all channels to the chosen gain and filter selection.
3. Gain Calibration - for every channel, repeat steps 3a through 3f.
   a) Set the channel so it is set to the calibration voltage. This is done by setting bit 14 (only) of the Prefilter Gain Registers. Do not change the value of the other bits in this register.
   b) Set the calibration voltage so the channels have positive full scale voltage. The full scale voltage is dependent on the gain setting for that channel. The expected positive full scale calibration voltage will be referred to as $\text{POSCAL}_{\text{EXPECTED}}$ in the following equation. The calibration voltage is set by writing to the calibration Register (Operational register in A24 space at offset 2_{16}). A table of the register settings is available of page 17. Also, be sure to wait for the voltage to settle to 16-bit accuracy once the calibration voltage is set.
   c) Measure the positive full scale voltage (average a minimum of 20 points). Use the equation below to convert the COUNTS read from the V208 into voltage RTO. The measured positive full scale calibration voltage is referred to as $\text{POSCAL}_{\text{MEASURED}}$ in the equation.

$$VOLTAGE(RTO) = (COUNTS - 32768) \bullet 312.5\mu Volts$$

   d) Set the calibration voltage so the channels have negative full scale voltage. The full scale voltage is dependent on the gain setting for that channel. The expected negative full scale calibration voltage will be referred to as $\text{NEGCAL}_{\text{EXPECTED}}$ in the equation in step 3f. The calibration voltage is set by writing to the Calibration Register (Operational register in A24 space at offset 2_{16}). A table of the register settings is available of page 17. Also, be sure to wait for the voltage to settle to 16-bit accuracy once the calibration voltage is set.
   e) Measure the negative full scale voltage (average a minimum of 20 points) using the equation in step 3c. The measured negative full scale calibration voltage is referred to as $\text{NEGCAL}_{\text{MEASURED}}$ in the equation in step 3f.

f) Read the gain error for the calibrator voltage from the Correction Table (page 37). The gain error correction term is referred to as GAIN COEF in the following equation.

$$GAIN = \frac{POSCAL_{MEASURED} - NEGCAL_{MEASURED}}{\left(POSCAL_{EXPECTED} - NEGCAL_{EXPECTED}\right) \bullet \left(1 + GAINCOEF \bullet 10^{-6}\right)}$$

Record the gain for the channel as calculated above.

4. Offset Calibration - for every channel
   a) Set the calibration voltage to ground. Write 4000_{16} to the Calibration Register (Operation register in A24 space at offset 2_{16}). Wait for the filter to settle to 16-bit accuracy.
   b) Measure the offset voltage (average a minimum of 20 points) using the equation in step 3c. The measured offset voltage is referred to as $\mathrm{OFFSET}_{\mathrm{MEASURED}}$ in the following equation.
   c) Read the offset error for the calibrator voltage from the Correction Table (page 37). The offset error correction term is referred to as OFFSET COEF in the following equation.

$$OFFSET = \frac{OFFSET_{MEASURED}}{GAIN} + \left(OFFSETCOEF \bullet 10^{-9}\right)$$

Record the offset error for the channel as calculated above.

5. Applying the channel calibration coefficients - Example with V208

The V208 represents voltage in an offset binary format. In the following two equations, this variable will be referred to as COUNTS.

When calibration is unnecessary, the gain is derived simply from the gain settings in the Prefilter Gain and Post Filter Gain Registers. This uncalibrated gain is represented with the variable $\mathrm{GAIN}_{\mathrm{UNCALIBRATED}}$ in the following equation. Convert counts to uncalibrated voltage using the following equation:

$$UNCALIBRATEDVOLTAGE = \frac{\left(COUNTS - 32768\right) \bullet 312.5\mu Volts}{GAIN_{UNCALIBRATED}}$$

For best accuracy, convert counts to calibrated voltage using the following equation:

$$CALIBRATEDVOLTAGE = \frac{\left(COUNTS - 32768\right) \bullet 312.5\mu Volts}{GAIN} - OFFSET$$

Suggestions for Highest Accuracy

1.) For best accuracy, it is recommended (but not necessary) to keep channels with similar gains grouped together.

2.) For best accuracy, it is recommended (but not necessary) that channels be organized with the gain set in decreasing order. The MUX-bus scan list should also follow this order.

3.) Unused channels should be grounded.

4.) Refer to Appendix A for a discussion of the recommended grounding scheme. In general, the best solution is to use shielded balanced pair wire with the shield grounded at the V243.

V243 Configuration Registers

ID / Logical Address Register

00_{16}	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
(R)	(0)	(1)	(0)	(0)	(1)	(1)	(1)	(1)	(0)	(0)	(1)	(0)	(1)	(0)	(0)	(1)

( ) Power Up Value

Bit(s)	Mnemonic	Meaning
15,14	Device Class	This is an Extended Register-Based Device.
13,12	Address Space	This module requires the use of A16/A24 address space.
11-00	Manufacture's ID	3881 ($F29_{16}$) for KineticSystems.

00_{16}	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
(W)									LA7	LA6	LA5	LA4	LA3	LA2	LA1	LA0

( ) Power Up Value

For WRITE transactions, bits fifteen through eight are not used. These bits may be written with any data pattern. In Dynamically Configured systems (and the Logical Address switches were set to a value of 255), bits seven through zero are written with the Logical Address value. This register is used by the resource manager to dynamically set a module's logical address.

Device Type Register

02_{16}	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
(R)	(1)	(0)	(0)	(1)	(0)	(0)	(1)	(0)	(0)	(1)	(0)	(0)	(0)	(0)	(1)	(1)

( ) Power Up Value

This READ ONLY register returns 9243_{16}.

Bit(s)	Mnemonic	Meaning
15-12	Required Memory	The V243 requires 16 Kilo bytes of additional memory space.
11-00	Model Code	Identifies this device as a V243 (243_{16}).

Status/Control Register

04_{16}	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
(Mixed)	A24 ENA (0)	Modid (0)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	Ready (0)	(1)	(1)	Soft Reset (0)

( ) Power Up Value

The bit assignments for the Status/Control register are defined as follows:

Bit(s)	Mnemonic	Meaning
15	A24 ENA	This bit is written with a "1" to enable A24 addressing and reset to "0" to disable these registers. **This bit must be set to allow access to the module's Operational Registers.** Reads of this bit indicate its current state. This bit is reset to "0" by the assertion of SYSRESET*. This bit is set by the resource manager once the offset register has been written.
14	Modid*	This read only bit is set to a "1" if the module is not selected with the MODID line on P2. A "0" in this bit location indicates the device is selected via a high state on its P2 MODID line. This bit is used by the resource manager to dynamically configure the V243.
13-04	Not Used	These bits are not used and are read as "1s".
03	Ready	A "1" in this bit indicates the successful completion of register initialization.
02-01	Not Used	These bits are not used and are read as "1s".
00	Soft Reset	Writing a "1" to this bit forces the device into the Soft Reset State. While in this state, the module will only allow access to its Configuration Registers. Writing a "0" to this bit will the signal the V243 to begin executing its self test. **This bit must be cleared along with the Pass and Ready bits set before any access to the Operational Registers is allowed.**

Offset Register

06_{16}	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
(RW)	OFF 15 (0)	OFF 14 (0)	OFF 13 (0)	OFF 12 (0)	OFF 11 (0)	OFF 10 (0)	OFF 9 (0)	OFF 8 (0)	OFF 7 (0)	OFF 6 (0)	OFF 5 (0)	OFF 4 (0)	OFF 3 (0)	OFF 2 (0)	OFF 1 (0)	OFF 0 (0)

( ) Power Up Value

After SYSRESET* and prior to self test all bits are reset to "0". Otherwise, a read or write

$$A24BASEADDRESS = OFF[15:0] \bullet 100_{16}$$

defines the base address of the module's A24 registers.

Attribute Register

08_{16}	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
(R)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	IR* (0)	IH* (1)	IC* (0)

( ) Power Up Value

This read only register returns $FFFA_{16}$ on READ transactions. Write transactions to this register have no effect and its usage is reserved for future definition.

Bit(s)	Mnemonic	Meaning
15-03	Reserved	These bits are read as "1s" and reserved for future definition.
02	IR*	This bit is read as a "0" to signify that the V243 is capable of generating interrupts.
01	IH*	This bit is read as a "1" and indicates the V243 is not capable of Interrupt Handler Control.
00	IS*	This bit is set to "0" to indicate the V243 has Interrupt Status Reporting capability.

Serial Number High

$0A_{16}$	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
(R)	SER 31 (?)	SER 30 (?)	SER 29 (?)	SER 28 (?)	SER 27 (?)	SER 26 (?)	SER 25 (?)	SER 24 (?)	SER 23 (?)	SER 22 (?)	SER 21 (?)	SER 20 (?)	SER 19 (?)	SER 18 (?)	SER 17 (?)	SER 16 (?)

( ) Power Up Value

Serial Number Low

$0C_{16}$	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
(R)	SER 15 (?)	SER 14 (?)	SER 13 (?)	SER 12 (?)	SER 11 (?)	SER 10 (?)	SER 9 (?)	SER 8 (?)	SER 7 (?)	SER 6 (?)	SER 5 (?)	SER 4 (?)	SER 3 (?)	SER 2 (?)	SER 1 (?)	SER 0 (?)

( ) Power Up Value

These READ ONLY registers indicate the serial number of the module. Each module is given a unique serial number. The serial number is represented by a 32-bit unsigned integer. The least significant bits (LSBs) reside in the Serial Number Low register while the most significant bits (MSBs) are in the Serial Number High register. Writing to these registers will have no effect and its use is reserved. For example, assume the module's serial number is 10064_{16} (65636). A read of the Serial Number High register returns 0001_{16} (1 => 1 * 65536); and the Serial Number Low register returns 0064_{16} (100).

Version Number Register

$0E_{16}$	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
(R)	Firmware Version # (?)				Firmware Revision # (?)				Hardware Version # (?)				Hardware Revision # (?)			

( ) Power Up Value

This READ ONLY register indicates the hardware and firmware revision number of the module. A write to this register has no effect on its contents. The fields of this register are explained as follows:

Bits	Mnemonic	Meaning
15-12	Firmware Version #	Firmware Version Number
11-08	Firmware Revision #	Firmware Revision Number
07-04	Hardware Version #	Hardware Version Number
03-00	Hardware Revision #	Hardware Revision Number

The combination of Firmware Version Number and Firmware Revision Number indicate the module's firmware version level. These two fields contain two four bit integers and are joined to form the level.

The combination of Hardware Version Number and Hardware Revision Number indicate the module's hardware version level. These two fields contain two four bit integers and are joined to form the level.

Reserved

10_{16}-18_{16}	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
(R)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)

( ) Power Up Value

Interrupt Status Register

$1A_{16}$	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
(RW)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	Logical Address FF_{16}							

( ) Power Up Value

This READ ONLY register is defined as follows:

Bit(s)	Mnemonic	Meaning
15-8	Reserved	These bits are read as "1s".
7-0	Logical Address	These bits contain the Logical Address of the V243 during an interrupt acknowledge cycle. These bits are read as all ones during a non-interrupt acknowledge cycle.

Interrupt Control Register

1C$_{16}$	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
(RW)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	Over Mask (1)	IREN * (1)	(1)	IRQ2 (1)	IRQ1 (1)	IRQ0 (1)	(1)	(1)	(1)

( ) Power Up Value

Using this register to control interrupts is defined as follows:

Bits(s)	Mnemonic	Meaning
15 - 09	Not Used	These bits are not used and should be written as "1s"
8	Over Mask	A "1" in this bit masks (prevents) an overlap from Generating an interrupt request. This bit must be written with a "0" to allow the overlap to generate an interrupt.
7	IREN*	A one in this bit is used to disable interrupt generation. This bit prevents any source from generating an interrupt. A zero in this field enables interrupt generation.
6	Not Used	This bit is reserved for use during interrupt handling. Since the V243 is not capable of interrupt handling, this bit should always be written with a "1".
5-3	IRQ2*-IRQ0*	This 3-bit field selects the VXIbus interrupt line associated with the interrupt according to the following table:

IRQ2* (D05)	IRLQ* (D04)	IRQ0* (D03)	Interrupt Request Line
0	0	0	IRQ7
0	0	1	IRQ6
0	1	0	IRQ5
0	1	1	IRQ4
1	0	0	IRQ3
1	0	1	IRQ2
1	1	0	IRQ1
1	1	1	Disconnected

Bits(s)	Mnemonic	Meaning
2-0	Not Used	These bits are reserved for selecting an interrupt handler line. The V243 does not have interrupt handler capabilities, these bits should always be written with "1s".

Subclass Register

$1E_{16}$	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
(R)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(0)

( ) Power Up Value

Reads of this register return $FFFE_{16}$. Writes to this register have no effect. The read contents are defined as follows:

Bit(s)	Meaning
15	This bit indicates that the V243 is a VXIbus defined Extended Device.
14-0	These bits indicate that this is an Extended Register Based Device.

Suffix High Register

<table>
<tr><td>20_{16}</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>09</td><td>08</td><td>07</td><td>06</td><td>05</td><td>04</td><td>03</td><td>02</td><td>01</td><td>00</td></tr>
<tr><td>(R)</td><td colspan="16">Suffix High Register
(?)</td></tr>
</table>

( ) Power Up Value

Suffix Low Register

<table>
<tr><td>22_{16}</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>09</td><td>08</td><td>07</td><td>06</td><td>05</td><td>04</td><td>03</td><td>02</td><td>01</td><td>00</td></tr>
<tr><td>(R)</td><td colspan="16">Suffix Low Register
(?)</td></tr>
</table>

( ) Power Up Value

The following two registers are KineticSystems defined and hold the module's suffix. The suffix determines the particular option of the module. This information can be used to remotely establish available channel count, filtering options, etc. of the module. For further information on each option, refer to the Ordering Information section of this manual.

The module's suffix is always composed of four ASCII characters. The Suffix High register contains the first two characters; while, the last two characters are in the Suffix Low register. For instance, assume the module is a model V243-VA91. The module's suffix is "VA91". Converting this to ASCII yields 56413931_{16}.

User Defined Registers

24_{16}	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
(RW)	User Defined Register (?)															

$3E_{16}$	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
(RW)	User Defined Register (?)															

( ) Power Up Value

Offsets 24_{16} through $3E_{16}$ are READ/WRITE registers and may be used to store user defined data. These registers are contained in non-volatile EEPROM. Special Note: Special care must be taken while writing information into these. Once a register has been written, a minimum of 3mS must pass before any User Defined Register or Correction Table Registers may be read or written to.

V243 Operational Registers

Configuration Register (Mixed)

00_{16}	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
(Mixed)	(1)	(1)	(1)	(1)	Con 3 (?)	Con 2 (?)	Con 1 (?)	Con 0 (?)	Therm (0)	Ovrlp (0)	Adsel (0)	(1)	(1)	(1)	(1)	(1)

( ) Power Up Value

Bits(s)	Mnemonic	Meaning
11-08	Con[3:0]	1111_2 No Termination Assembly 1101_2 V750-ZB11 Isothermal Termination Assembly (16, 32, or 48 Channel Options only)
7	Therm	When set, open thermocouple is enabled
6	Ovrlp	Overlap detected by V243 on MUX bus, cleared by writing this bit with a zero.
5	Adsel	Address select bit for MUX bus 0 Setup Mode 1 Run Mode

Calibration Register

02_{16}	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
(RW)	Refs (1)	GND (0)	Calout (0)	(1)	(1)	(1)	(1)	Cal 8 (0)	Cal 7 (0)	Cal 6 (0)	Cal 5 (0)	Cal 4 (0)	Cal 3 (0)	Cal 2 (0)	Cal 1 (0)	Cal 0 (0)

( ) Power Up Value

Bits(s)	Mnemonic	Meaning
15	Refs	0 MUX bus 1 On board source (recommended)
14	GND	When set, the calibration voltage is grounded for offset measurements.
13	Calout	When set, the calibration voltage is switched out on pins 32 and 66 of P3, P4, P5, P6. In this way, multiple V243s may have their Calout voltage daisy chained together into one precision meter. That way, the Calout could be enabled one at a time on each V243 making the module calibration all controllable in software.
8	Cal 8	Cal-
7	Cal 7	Cal+
6	Cal 6	Selects Reference Source Multiplied by .2
5	Cal 5	Selects Reference Source Multiplied by .5
4	Cal 4	Selects Reference Source Multiplied by 1
3	Cal 3	Scale Reference Source Multiplied by .001
2	Cal 2	Scale Reference Source Multiplied by .01
1	Cal 1	Scale Reference Source Multiplied by .1
0	Cal 0	Scale Reference Source Multiplied by 1

At least one bit in each group must be set to a "1". See page 18, Table 3.

Output Buffer Register

04_{16}	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
(RW)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	Out 7 (0)	Out 49 (0)	Out 2 (0)	Out 1 (0)

( ) Power Up Value

Bits(s)	Mnemonic	Meaning
3 - 0	Out [i]	Output buffer control for channel "i" 1 Filtered Output (filter controlled by Prefilter Gain Register) 0 Non-filtered Output

$\$ 10_{16}$ - Channel 1 - 8 Prefilter Gain Register (RW)
$\$ 18_{16}$ - Channel 9 - 16 Prefilter Gain Register (RW)
$\$ 20_{16}$ - Channel 17 - 24 Prefilter Gain Register (RW)
$\$ 28_{16}$ - Channel 25 - 32 Prefilter Gain Register (RW)
$\$ 30_{16}$ - Channel 33 - 40 Prefilter Gain Register (RW)
$\$ 38_{16}$ - Channel 41 - 48 Prefilter Gain Register (RW)
$\$ 40_{16}$ - Channel 49 - 56 Prefilter Gain Register (RW)
$\$ 48_{16}$ - Channel 57 - 64 Prefilter Gain Register (RW)
$\$ 50_{16}$ - Channel 65 - 72 Prefilter Gain Register (RW)
$\$ 58_{16}$ - Channel 73 - 80 Prefilter Gain Register (RW)
$\$ 60_{16}$ - Channel 81 - 88 Prefilter Gain Register (RW)
$\$ 68_{16}$ - Channel 89 - 96 Prefilter Gain Register (RW)

	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
(RW)		IMUX					FLTR 1	FLTR 0							FGN1	FGN0
	(1)	(0)	(1)	(1)	(1)	(1)	(0)	(0)	(1)	(1)	(1)	(1)	(1)	(1)	(0)	(0)

( ) Power Up Value

Bits(s)	Mnemonic	Meaning
14	IMUX	Input Selection 0 Line 1 Local Calibration (voltage set by Calibration Register)
9 - 8	FLTR[1:0]	Filter Selection 00_2 10 Hz Butterworth Filter 01_2 50 Hz Butterworth Filter 10_2 500 Hz Butterworth Filter 11_2 Bypass
1 - 0	FGN[1:0]	First Stage Gain Selection 00_2 Gain = 1.0 01_2 Gain = 10.0 10_2 Gain = 100.0 11_2 Invalid

Postfilter Gain Register

100_{16} - $1C6_{16}$	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
(RW)														SGN2	SGN1	SGN0
	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(?)	(?)	(?)	(?)	(?)	(?)	(?)	(?)

( ) Power Up Value

The second stage gain get be set on a per channel basis. There is one register for each channel starting at offset 100_{16}. The offset may be computed for a given channel number by the following equation.

$$OFFSET = FE_{16} + (CHANNEL \bullet 2)$$

The second stage gain for the isothermal channels should always be set to a gain of 1. The isothermal channels for the P3, P4, P5, and P6 are $1C0_{16}$, $1C2_{16}$, $1C4_{16}$, and $1C6_{16}$ respectively. The isothermal channels are labeled channels 97 to 100.

Bits(s)	Mnemonic	Meaning
2 - 0	SGN[2:0]	Second stage gain selection 000_2 Gain = 1.0 001_2 Gain = 2.0 010_2 Gain = 5.0 011_2 Gain = 10.0 100_2 Gain = 20.0 101_2 Invalid 110_2 Invalid 111_2 Invalid

Note: A bus error will occur if a write is attempted while in 'run mode'.

V243 Scan Ram

200_{16}-1200_{16}	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
(RW)	EOL (?)	Enabl (?)	(?)	(?)	(?)	(?)	(?)	(?)	(?)	Channel Number						

( ) Power Up Value

Bits(s)	Mnemonic	Meaning
15	EOL	End of list
14	Enabl	Enables CH[6:0]
6 - 0	CH[6:0]	Selects channel number on the V243 to be placed out on the MUX bus.

0000000_2	Channel 1
0000001_2	Channel 2
0000010_2	Channel 3
.	.
.	.
1011110_2	Channel 95
1011111_2	Channel 96
1100000_2	Isothermal Reference P3 (Channel 97)
1100001_2	Isothermal Reference P4 (Channel 98)
1100010_2	Isothermal Reference P5 (Channel 99)
1100011_2	Isothermal Reference P6 (Channel 100)

Note: A bus error will occur if a write is attempted while in 'run mode'.

V243 Correction Table

2000_{16} - $21FE_{16}$	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
(RW)	(?)	(?)	(?)	(?)	(?)	(?)	(?)	(?)	(?)	(?)	(?)	(?)	(?)	(?)	(?)	(?)

( ) Power Up

Value

To improve calibrated accuracy, correction coefficients for the calibrator are stored in non-volatile RAM. The calibration date is also stored. The following table shows the register offset in relation to the coefficient stored.

Offset	Function
\$ 2000_{16}	The month of the last calibration date. (1 to 12)
\$ 2002_{16}	The day of the last calibration date. (1 to 31)
\$ 2004_{16}	The year of the last calibration date. (example 1995)
\$ 2006_{16}	User definable.
· ·	
\$ $200E_{16}$	User definable.
\$ 2010_{16}	Gain correction coefficient for +- 10 volts.
\$ 2012_{16}	Gain correction coefficient for +- 5 volts.
\$ 2014_{16}	Gain correction coefficient for +- 2 volts.
\$ 2016_{16}	Gain correction coefficient for +- 1 volt.
\$ 2018_{16}	Gain correction coefficient for +- 500 mV.
\$ $201A_{16}$	Gain correction coefficient for +- 200 mV.
\$ $201C_{16}$	Gain correction coefficient for +- 100 mV.
\$ $201E_{16}$	Gain correction coefficient for +- 50 mV.
\$ 2020_{16}	Gain correction coefficient for +- 20 mV.
\$ 2022_{16}	Gain correction coefficient for +- 10 mV.
\$ 2024_{16}	Gain correction coefficient for +- 5 mV.
\$ 2026_{16}	Gain correction coefficient for +- 2 mV.
\$ 2028_{16}	Gain Coefficient checksum
\$ $203E_{16}$	Reserved.
\$ 2040_{16}	Offset correction coefficient for channel 1.
\$ 2042_{16}	Offset correction coefficient for channel 2.
· ·	
\$ $20FC_{16}$	Offset correction coefficient for channel 95.
\$ $20FE_{16}$	Offset correction coefficient for channel 96.
\$ 2100_{16}	Offset coefficient checksum.
· ·	
\$ $21FE_{16}$	Reserved.

The gain correction coefficient is stored as a signed 16-bit integer. This coefficient need only be used during gain calibration of a given channel. This coefficient should not be used on any isothermal reference.

The offset correction coefficient is also stored as a signed 16-bit integer. This coefficient need only be used during offset calibration of a given channel. This coefficient should not be used on any isothermal reference.

Equations for using both coefficients are explained on pages 19-21.

Coefficient Checksums

The gain coefficient checksum is stored as an unsigned 16-bit integer. It is calculated by adding the correction coefficients of memory locations 2010 to 2026 and is stored at memory location 2028.

The offset coefficient checksum is stored as an unsigned 16-bit integer. It is calculated by adding the correction coefficients of memory locations 2040 to the final channel location (this is option dependent) and is stored at memory location 2100.

Special Note: Special care must be taken while writing information into these registers. The CAL strap must be moved to the enabled position to enable writes to the Correction Table. Once a register has been written, a minimum of 3ms must pass before any other Correction Table or User Defined Registers may be read or written to. The CAL strap must be returned to the disable position to prevent inadvertently overwriting these locations.

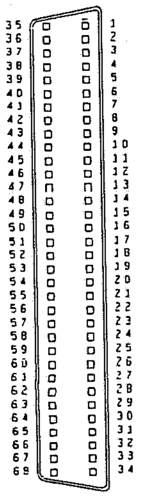


FIGURE 4 - 68 Pin SCSI II Connectors: P3, P4, P5, and P6

TABLE 4 - 68 Pin SCSI II Connectors: P3, P4, P5, and P6 Pinout

Pin #	P3 Description	P4 Description	P5 Description	P6 Description
1	Channel 1 +	Channel 25 +	Channel 49 +	Channel 73 +
35	Channel 1 -	Channel 25 -	Channel 49 -	Channel 73 -
2	Channel 2 +	Channel 26 +	Channel 50 +	Channel 74 +
36	Channel 2 -	Channel 26 -	Channel 50 -	Channel 74 -
3	Channel 3 +	Channel 27 +	Channel 51 +	Channel 75 +
37	Channel 3 -	Channel 27 -	Channel 51-	Channel 75 -
4	Channel 4 +	Channel 28 +	Channel 52 +	Channel 76 +
38	Channel 4 -	Channel 28 -	Channel 52 -	Channel 76 -
5	Channel 5 +	Channel 29 +	Channel 53 +	Channel 77 +
39	Channel 5 -	Channel 29 -	Channel 53 -	Channel 77 -
6	Channel 6 +	Channel 30 +	Channel 54 +	Channel 78 +
40	Channel 6 -	Channel 30 -	Channel 54 -	Channel 78 -
7	Channel 7 +	Channel 31 +	Channel 55 +	Channel 79 +
41	Channel 7 -	Channel 31 -	Channel 55 -	Channel 79 -
8	Channel 8 +	Channel 32 +	Channel 56 +	Channel 80 +
42	Channel 8 -	Channel 32 -	Channel 56 -	Channel 80 -
9	Channel 9 +	Channel 33 +	Channel 57 +	Channel 81 +
43	Channel 9 -	Channel 33 -	Channel 57 -	Channel 81 -
10	Channel 10 +	Channel 34 +	Channel 58 +	Channel 82 +
44	Channel 11 -	Channel 34 -	Channel 58 -	Channel 82 -
11	Channel 11 +	Channel 35 +	Channel 59 +	Channel 83 +
45	Channel 11 -	Channel 35 -	Channel 59 -	Channel 83 -
12	Channel 12 +	Channel 36 +	Channel 60 +	Channel 84 +
46	Channel 12 -	Channel 36 -	Channel 60 -	Channel 84 -
13	Channel 13 +	Channel 37 +	Channel 61 +	Channel 85 +
47	Channel 13 -	Channel 37 -	Channel 61 -	Channel 85 -
14	Channel 14 +	Channel 38 +	Channel 62 +	Channel 86 +

Pin #	P3 Description	P4 Description	P5 Description	P6 Description
48	Channel 14 -	Channel 38 -	Channel 62 -	Channel 86 -
15	Channel 15 +	Channel 39 +	Channel 63 +	Channel 87 +
49	Channel 15 -	Channel 39 -	Channel 63 -	Channel 87 -
16	Channel 16 +	Channel 40 +	Channel 64 +	Channel 88 +
50	Channel 16 -	Channel 40 -	Channel 64 -	Channel 88 -
17	Channel 17 +	Channel 41 +	Channel 65 +	Channel 89 +
51	Channel 17 -	Channel 41 -	Channel 65 -	Channel 89 -
18	Channel 18 +	Channel 42 +	Channel 66 +	Channel 90 +
52	Channel 18 -	Channel 42 -	Channel 66 -	Channel 90 -
19	Channel 19 +	Channel 43 +	Channel 67 +	Channel 91 +
53	Channel 19 -	Channel 43 -	Channel 67 -	Channel 91 -
20	Channel 20 +	Channel 44 +	Channel 68 +	Channel 92 +
54	Channel 20 -	Channel 44 -	Channel 68 -	Channel 92 -
21	Channel 21 +	Channel 45 +	Channel 69 +	Channel 93 +
55	Channel 21 -	Channel 45 -	Channel 69 -	Channel 93 -
22	Channel 22 +	Channel 46 +	Channel 70 +	Channel 94 +
56	Channel 22 -	Channel 46 -	Channel 70 -	Channel 94 -
23	Channel 23 +	Channel 47 +	Channel 71 +	Channel 95 +
57	Channel 23 -	Channel 47 -	Channel 71 -	Channel 95 -
24	Channel 24 +	Channel 48 +	Channel 72 +	Channel 96 +
58	Channel 24 -	Channel 48 -	Channel 72 -	Channel 96 -
25	Ground	Ground	Ground	Ground
59	Ground	Ground	Ground	Ground
26	Ground	Ground	Ground	Ground
60	Ground	Ground	Ground	Ground
27				
61				

Pin #	P3 Description	P4 Description	P5 Description	P6 Description
28				
62				
29				
63				
30				
64				
31	Chan 1 Out +	Chan 25 Out +	Chan 49 Out +	Chan 73 Out +
65	Chan 1 Out -	Chan 25 Out -	Chan 49 Out -	Chan 73 Out -
32	Cal Out +	Cal Out +	Cal Out +	Cal Out +
66	Cal Out -	Cal Out -	Cal Out -	Cal Out -
33	Isothermal P3 -	Isothermal P4 -	Isothermal P5 -	Isothermal P6 -
67	Isothermal P3 +	Isothermal P4 +	Isothermal P5 +	Isothermal P6 +
34	No Connect	No Connect	No Connect	No Connect
68	No Connect	No Connect	No Connect	No Connect

APPENDIX

Driving Balanced Analog Inputs from Unbalanced Sources

Robert T. Cleary
KineticSystems Corporation
900 N. State Street
Lockport, Illinois 60441

Abstract - Most low to medium-speed analog-to-digital channels include a balanced input in order to provide some degree of rejection of common-mode voltage. Unfortunately, most signal sources are unbalanced with one side grounded. The two most-often-asked questions regarding the connection of unbalanced sources such as transducer amplifiers to balanced inputs such as A/D converters are:

1. Should coaxial or balanced-pair cable be used?
2. Should the cable be grounded at one end or at both ends?

This technical note shows test results that indicate the optimum cable type (shielded balanced pair) as well as the best grounding practice (generally, grounding at both ends).

I. THE TEST SET-UP

The test set-up is shown in Figure 1. The output impedance of a typical amplifier is generally low (100 ohms down to a fraction of an ohm). This is simulated with a 10 ohm resistor. Since no actual signal is present, any resulting voltage would be noise. The test includes 50-foot cables between the simulated signal source and the balanced input. A wideband instrumentation amplifier represents the balanced input. This amplifier is monitored by a wideband oscilloscope. The noise environment is that of a typical development laboratory. The electrical conduit grounds for the source and loads are approximately 50 feet from each other. The coaxial cable to the oscilloscope is quite short; the instrumentation amplifier and oscilloscope share the same receptacle ground to limit noise generated from that connection.

II. COAXIAL VS. BALANCED CABLE

The noise performance for coaxial as well as shielded balanced-pair cable is shown in Figure 2. Note that Figure 2(a) shows approximately ±500 millivolts of high-frequency noise present when coaxial cable is used and that cable is grounded only at the source end. Figure 2(b) shows the effect of grounding the coaxial cable at both ends. The high-frequency "hash" is replaced by a ±100 millivolt signal that is predominately 60 Hz and its harmonics. This is expected because any potential difference between the two conduit systems results in a voltage being developed across the cable shield that is also the signal return for the analog system. Figure 2(c) shows the resulting interference when a balanced-pair cable is used and it is grounded at both ends. A later measurement with higher resolution on the oscilloscope [Figure 4(a)] shows the noise level to be approximately ±3 millivolts. This is an improvement of 167:1 (or 45 dB) over Figure 2(a).

III. GROUNDING BALANCED-PAIR CABLES

Because the grounding of coaxial cables results in shield-conducted power line noise, one is often reluctant to ground both ends of a balanced-pair or twinaxial cable. However, the balanced-pair cable should generally be grounded at both ends to minimize noise pickup. The only exception to this is when one end is "floating" (not connected to ground); this situation will be dealt with later in this note.

The shield of a balanced-pair cable can be grounded at both ends because, in a balanced system, the shield is not a signal-carrying conductor for normal-mode signals as it is in a coaxial cable. The results of grounding a balanced-pair cable at one end only are shown in Figure 3. Figure 3(a) shows the shield connected at the load only, while Figure 3(b) shows the shield connection at the source only. Both of these approaches give unacceptable results. The noise level is nearly as high as that found with the coaxial cable in Figure 2(a).

IV. GROUNDED OR FLOATING SOURCES/LOADS

Thus far we have described the grounding recommendations and cable type when the source is grounded. Figure 4(a) uses the same set-up as was shown in Figure 2(c): balanced-pair cable with the shield grounded at both ends. The oscilloscope is now set to a higher sensitivity (5 millivolts/division compared with 100 millivolts/division in Figure 2). If the signal source is "floating" (isolated from ground) as in the case of a passive source (a thermocouple without amplifier, for example) or an amplified transducer that is powered from a ground-isolated source, then the connection should be as shown in Figure 4(b). The source-end shield is connected to the "common" of the source, not ground. Note that the "floating" source gives a noise level of ±1 millivolt, rather than the ±3 millivolts for the grounded source in Figure 4(a).

The signal conductors associated with the balanced input must always have some path to ground in order to supply the low bias current needed by the instrumentation amplifier. In the case where a "floating" source is used, this return path is via the cable shield and through the load-end ground connection.

If the load is isolated instead of the source (when using an isolated instrumentation amplifier, for example), then the source-end shield should be connected to ground, while the load-end shield is connected to the "common" of the load circuit. If both source and load are "floating," then one end should be connected to ground in addition to the shield connection to both circuit "commons." This is primarily for safety so that capacitive coupling to a high-voltage conductor in the same cable tray will not cause the analog cable to become charged.

V. Filtering the Signal

Another way to reduce the noise into a balanced circuit is to limit the bandwidth of the signal presented to the instrumentation amplifier. The cutoff frequency that can be used depends upon the signal rate-of-change to be monitored. Figure 5 shows the effect of adding an eight kilohertz single-pole passive low-pass filter. The components for this filter with a 3 dB loss at 8 kilohertz are shown in Figure 7. Note in Figure 5(a) that this filter reduced the noise from ±500 millivolts to ±3 millivolts for the coaxial cable case. The results are even better when combining the balanced-pair cable grounded at both ends with the filter. The noise in this case is reduced to ±500 microvolts. The filter should be located on the module or within the same rack (at the termination panel, for example) so that high-frequency noise is not picked up after the filter.

Reducing the cut-off frequency will decrease the noise further as well as to limit the system frequency response. For example, an option on the KSC 3512 A/D module contains input filters with a 10 Hz 3 dB rolloff to attenuate 50 or 60 Hz normal-mode noise. The 3527 A/D module integrates many readings over a one-cycle period of power-line frequency to attenuate the effects of noise from that source. Other noise will also be reduced by this averaging technique.

VI. Actual Signal Distribution Demonstrated

Thus far we have shown the noise response by using a resistor to simulate the signal source. We then used a Hewlett-Packard oscillator producing a 1000 Hz signal with an amplitude of 1.5 volts peak-to-peak as the signal source and transmitted this signal over the 50-foot cable. The results are shown in Figure 6. Note the high-frequency "hash" with the coaxial cable in Figure 6(a) and the "clean" signal in Figure 6(b).

VII. Other Considerations

The level of noise, even when using balanced-pair cable, is affected by the exposure and the relative noise potential between the transmitting and receiving ends of the analog path. When analog signals are transmitted between racks that are in close proximity, noise performance is generally improved by bonding the racks together with one or more large conductors. The important consideration is not how "clean" a ground is, but rather how much "noise" is present when measured from one part of the system to another part of that same system. This is the same effect that causes us not to feel that the earth's surface is rotating at nearly 1,000 miles per hour because all parts that we have contact with are rotating together.

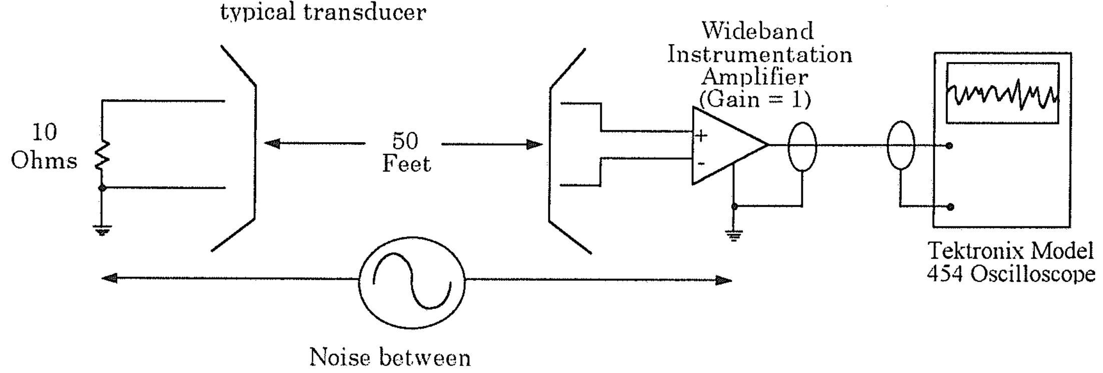
The cable type and grounding methods shown here also apply to balanced-line binary signaling such as that provided with RS-422. The EIA RS-422 specification indicates that the shield should be grounded at both ends.

VIII. Conclusions

When receiving balanced analog signals, the following conclusions can be drawn:

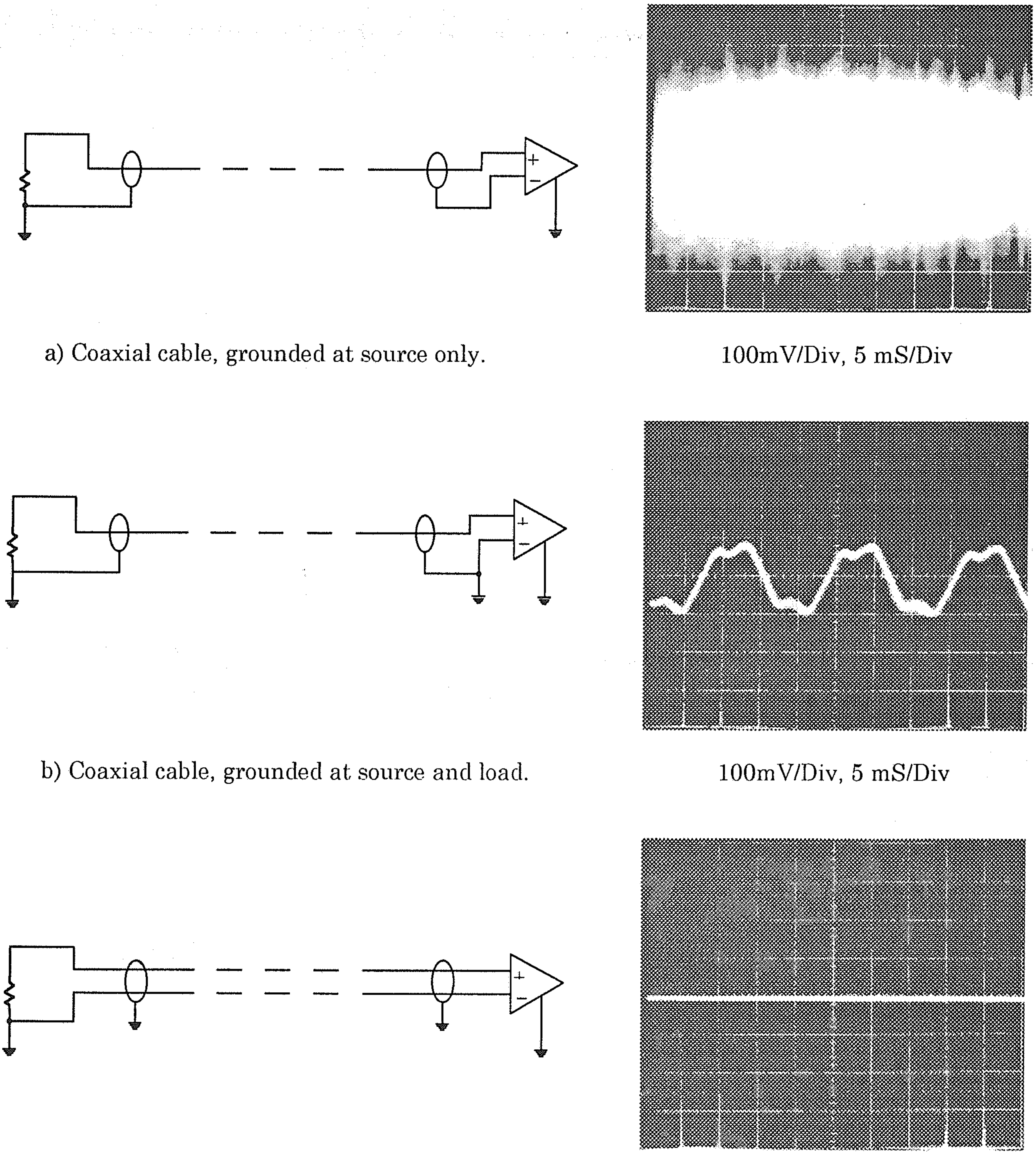
1. Shielded balanced-pair cable should be used with both ends grounded when the source and load are ground-referenced [Figure 4(a)].
2. If the source is isolated, then the source-end shield should be connected to the "common" of the source circuit [Figure 4(b)].
3. If the load is isolated, then the load-end shield should be connected to the "common" of the load circuit.
4. A filter can be added to reduce the noise bandwidth of the input signal.

Note: 10 Ohm resistor simulates low-impedance typical transducer



Wideband Instrumentation Amplifier (Gain = 1)
10 Ohms
50 Feet
+
-
Tektronix Model 454 Oscilloscope
Noise between

Figure 1. The noise test set-up with a simulated unbalanced source.



a) Coaxial cable, grounded at source only.

100mV/Div, 5 mS/Div

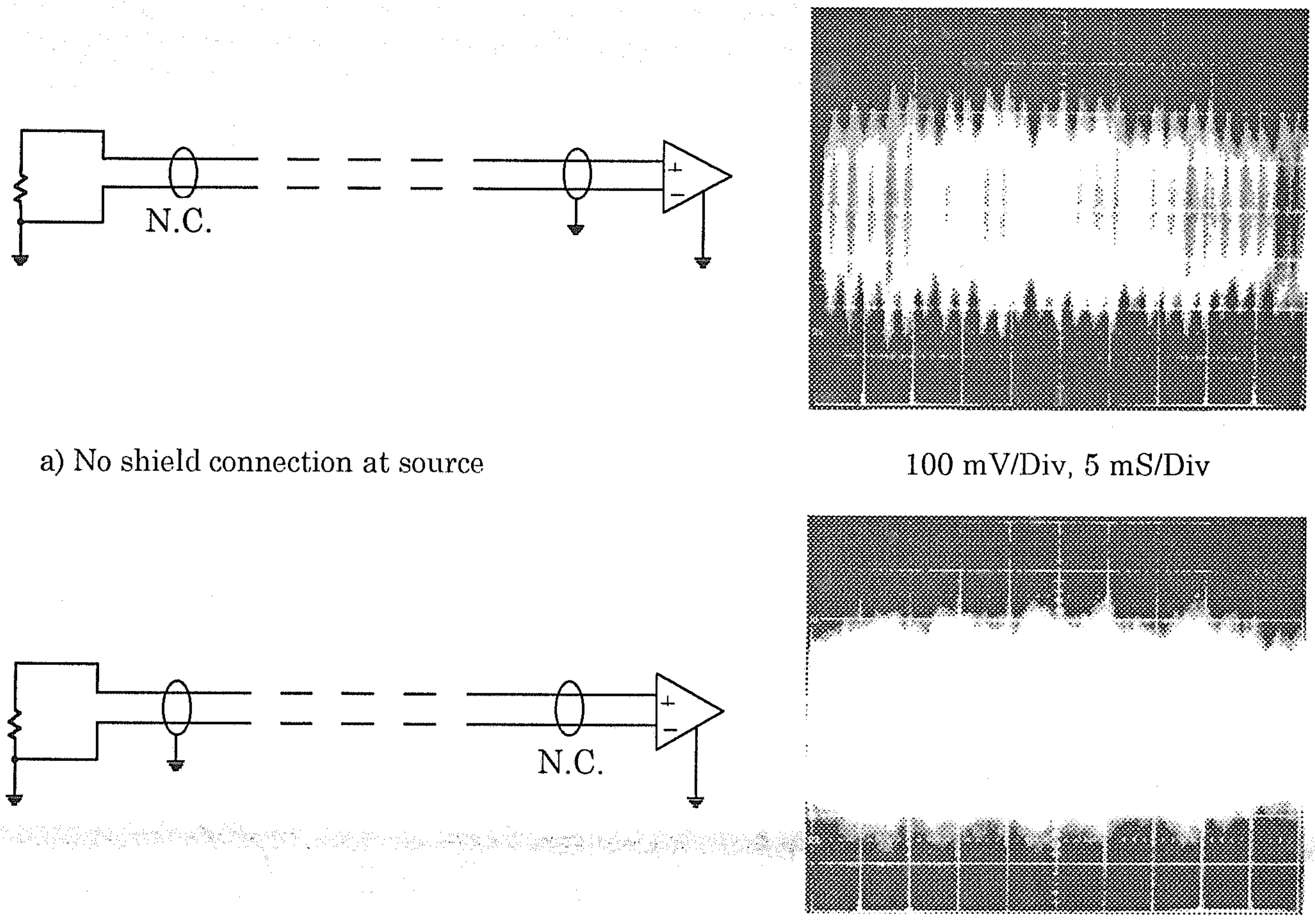
b) Coaxial cable, grounded at source and load.

100mV/Div, 5 mS/Div

c) Shielded balanced pair, grounded at both ends

100mV/Div, 5 mS/Div

Figure 2. Noise comparison between coaxial cable and shielded balanced pair.



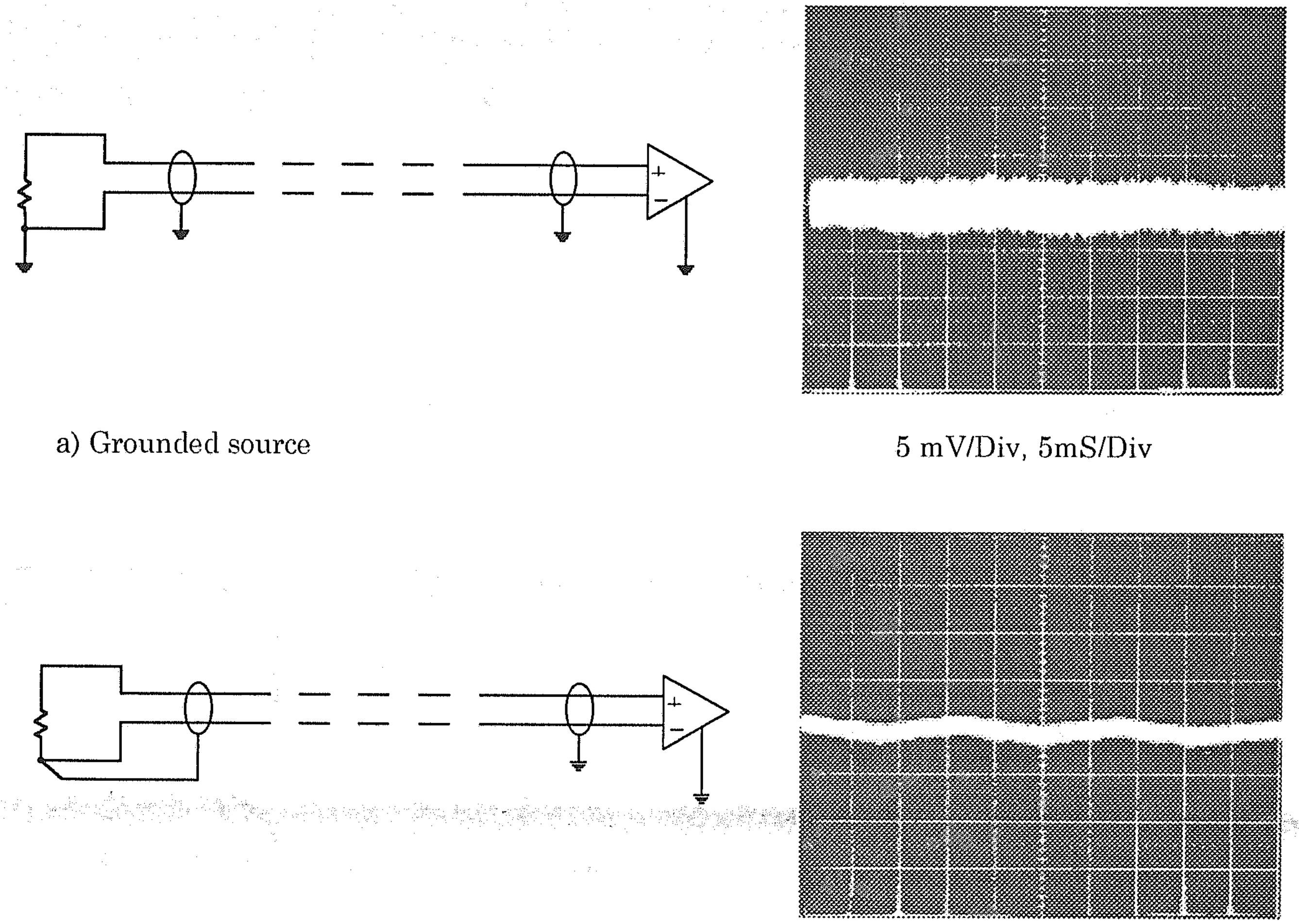
a) No shield connection at source

100 mV/Div, 5 mS/Div

b) No shield connection at load.

100 mV/Div, 5 mS/Div

Figure 3. Shielded balanced pair with a ground connection at one end only



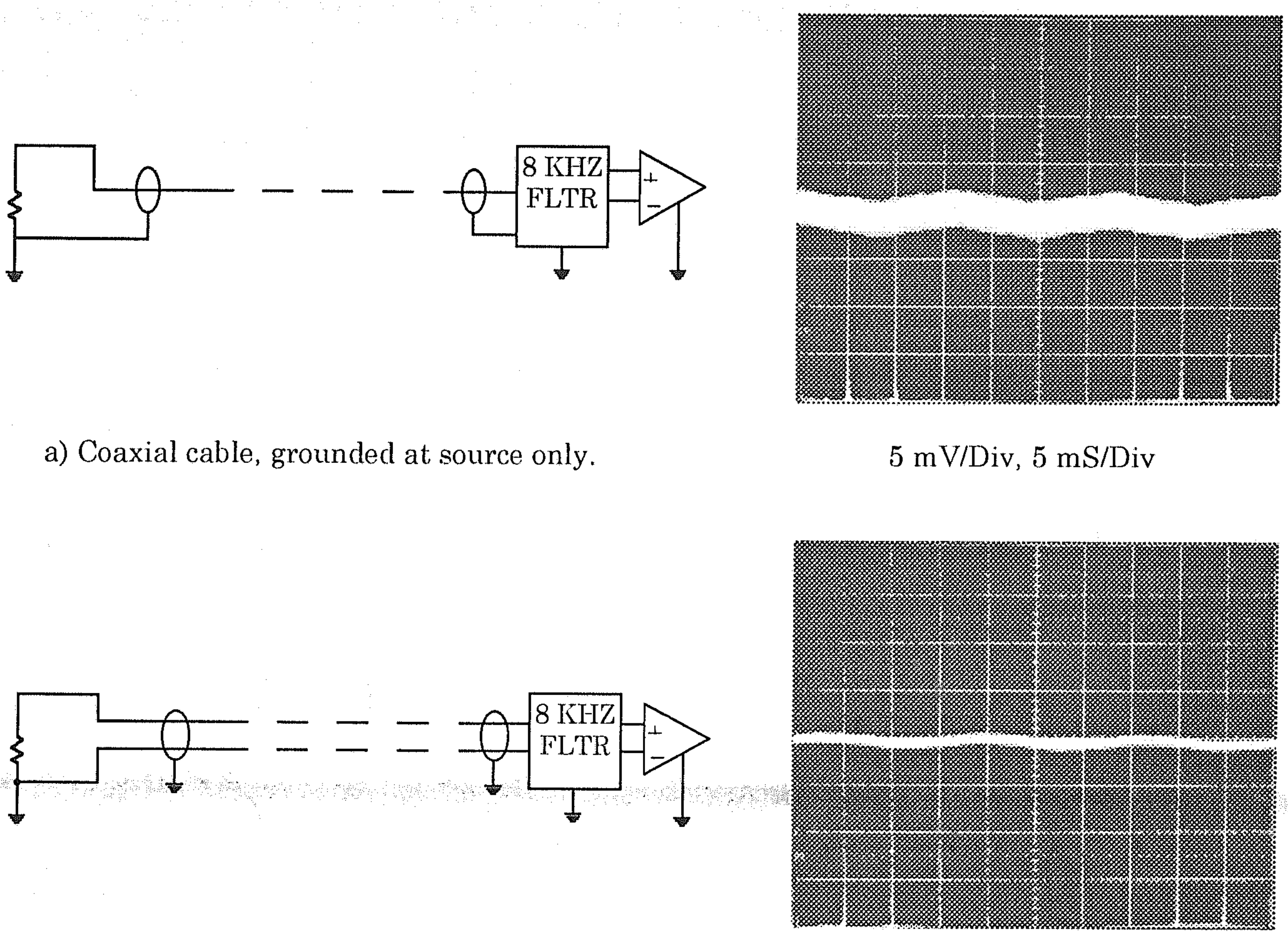
a) Grounded source

5 mV/Div, 5mS/Div

b) Floating source

5 mV/Div, 5 mS/Div

Figure 4. Noise comparison with a grounded and a floating source



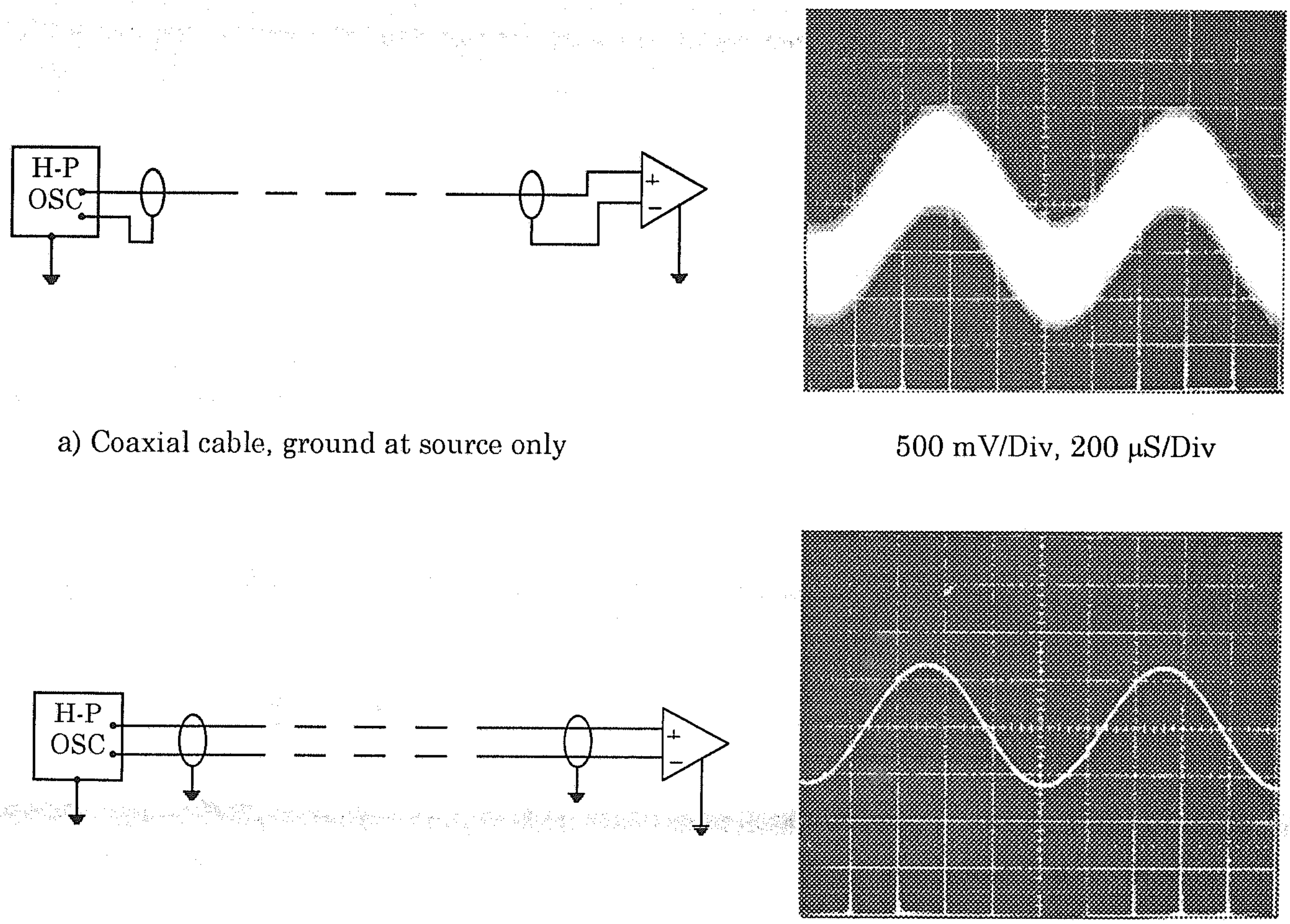
a) Coaxial cable, grounded at source only.

5 mV/Div, 5 mS/Div

b) Shielded balanced pair, ground at both ends

5 mV/Div, 5mS/Div

Figure 5. The offset of an 8 kHz filter on noise response.



a) Coaxial cable, ground at source only

500 mV/Div, 200 μS/Div

b) Shielded balanced pair, ground at both ends

500 mV/Div, 200 μS/Div

Figure 6. 1.5 volt p-p. 1000 Hz signal via coaxial and balanced-pair cables.

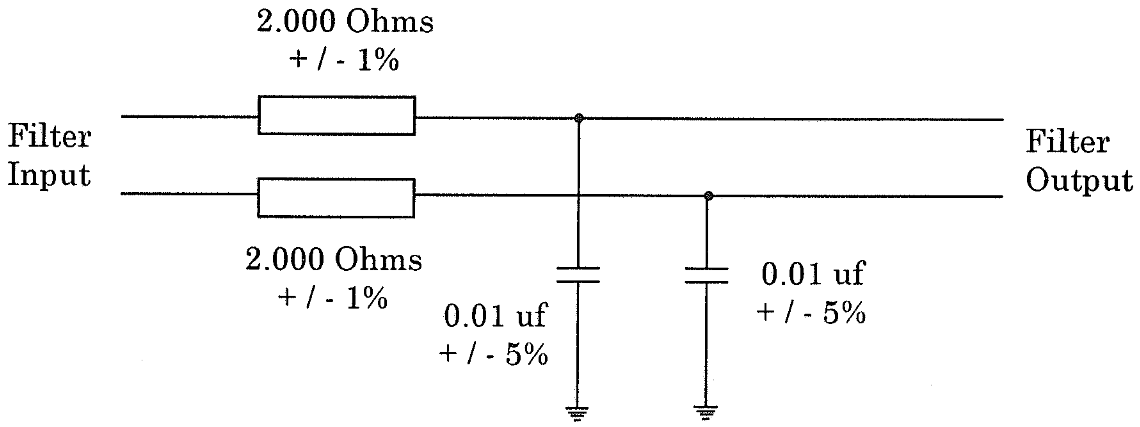


Figure 7. Details of the 8 kilohertz lowpass filter.

```
/*************************************************************************/
/*                                                                       */
/*                                                                       */
/*     Program Name : Channel.c                                          */
/*                                                                       */
/*                                                                       */
/*     This program example is a C/C++ code example for the National     */
/*     Instruments controller.  This program will calibrate all 48 or    */
/*     96 channels (depending on the option).  It will then take a       */
/*     calibrated measurement from each channel (average of 1000         */
/*     readings)                                                         */
/*                                                                       */
/*     V243 Logical Address = 5                                          */
/*     V208 Logical Address = 6                                          */
/*                                                                       */
/*************************************************************************/
#include <stdlib.h>
#include <stdio.h>
#include <time.h>

typedef unsigned short uint16;
typedef short int16;
typedef unsigned long uint32;

int16 number_channels();
void read_V208();
void setup();

uint16  const      A24SuperData = 0x6;
uint16  const      A32SuperData = 0x7;
uint16  const      V243 = 5;
uint16  const      V208 = 6;
float   const      Gain = 2000.0;
uint16  const      PostGainRegisterValue = 0x4;
uint16  const      PreGainRegisterValue = 0x2;
uint16  const      PosCalRegisterValue = 0x80A8;
uint16  const      NegCalRegisterValue = 0x8128;
float   const      AveragePoints = 1000.0;
float   const      CalAveragePoints = 1000.0;
uint16  const      GainCoefOffset = 0x2040;

/* array for possible 96 channels */
float          volt_data[97];

void main()
 {
 uint16 data,access;
 uint32 address,A24_V243,A32_V208;
```

```
int16  error,i,j,k,channel;
int16  offset_coef,gain_coef;
float  offset[97],gain[97],calibrated_voltage,uncalibrated_voltage;
float  poscal_measured[97],negcal_measured[97],offset_measured[97];

printf("\n\n                V243 Channel Calibration\n\n\n");

InitVXIlibrary();

/* Get A24 base address for V243      */
error = GetDevInfo(V243, 12, &A24_V243);
if (error != 0 ) {
  printf("Error: GetDevInfo() returned %d\n",error);
  CloseVXIlibrary();
  exit(1);
  }

/* Get A32 base address for V208      */
error = GetDevInfo(V208, 12, &A32_V208);
if (error != 0 ) {
  printf("Error: GetDevInfo() returned %d\n",error);
  CloseVXIlibrary();
  exit(1);
  }

/* Step 2                        */
/* setup MUX-bus                   */
/* all available channels (16, 32, 48, 64, or 96)        */
setup();

/* Step 3a                       */
/* Set V243 Prefilter Gain registers        */
/* 10 Hz filters, PreGain=100, Set channel    */
/* to calibration source              */
data = 0x4000 | PreGainRegisterValue;
for (i=0; i<number_channels(); i+=8) {
  address = A24_V243 + 0x10 + i;
  error = VXIout(A24SuperData,address,2,data);
  if (error != 0 ) {
    printf("Error: VXIout() returned %d\n",error);
    CloseVXIlibrary();
    exit(1);
    }
  }

/* Step 3b                             */
/* set positive calibration voltage, use on board source  */
/* at gain of 2000, 5mV is full scale               */
```

```
address = A24_V243 + 0x2;
data = PosCalRegisterValue;
error = VXIout(A24SuperData,address,2,data);
if (error != 0 ) {
  printf("Error: VXIout() returned %d\n",error);
  CloseVXIlibrary();
  exit(1);
  }

wait(2500); /* wait for 10 hz filter and cal voltage to settle */

/* Step 3c          */
for (channel=1; channel<=number_channels(); channel++)
  poscal_measured[channel] = 0.0;
for (i=1; i<=(int16)CalAveragePoints; i++) {
  read_V208();
  for (channel=1; channel<=number_channels(); channel++)
    poscal_measured[channel] += volt_data[channel];
  }
for (channel=1; channel<=number_channels(); channel++)
  poscal_measured[channel] /= CalAveragePoints;

/* Step 3d                              */
/* set negative calibration voltage, use on board source  */
/* at gain of 2000, -5mV is negative full scale          */
address = A24_V243 + 0x2;
data = NegCalRegisterValue;
error = VXIout(A24SuperData,address,2,data);
if (error != 0 ) {
  printf("Error: VXIout() returned %d\n",error);
  CloseVXIlibrary();
  exit(1);
  }

wait(2500); /* wait for 10 hz filter and cal voltage to settle */

/* Step 3e          */
for (channel=1; channel<=number_channels(); channel++)
  negcal_measured[channel] = 0.0;
for (i=1; i<=(int16)CalAveragePoints; i++) {
  read_V208();
  for (channel=1; channel<=number_channels(); channel++)
    negcal_measured[channel] += volt_data[channel];
  }
for (channel=1; channel<=number_channels(); channel++)
  negcal_measured[channel] /= CalAveragePoints;

/* Step 3f                        */
/* get error coefficient for gain = 2000  */
address = A24_V243 + GainCoefOffset;
error = VXIin(A24SuperData,address,2,&gain_coef);
if (error != 0 ) {
  printf("Error: VXIin() returned %d\n",error);
  CloseVXIlibrary();
  exit(1);
  }

/* poscal_expected-negcal_expected = 20.48 / Gain      */
for (channel=1; channel<=number_channels(); channel++)
  gain[channel] = (poscal_measured[channel] - negcal_measured[channel]) /
                         ( (20.48 / Gain) * (1.0 + (gain_coef*0.000001)) );

/* Step 4a                           */
/* set calibration voltage to ground  */
address = A24_V243 + 0x2;
data = 0x4000;
error = VXIout(A24SuperData,address,2,data);
if (error != 0 ) {
  printf("Error: VXIout() returned %d\n",error);
  CloseVXIlibrary();
  exit(1);
  }

wait(2500); /* wait for 10 hz filter and cal voltage to settle */

/* Step 4b          */
for (channel=1; channel<=number_channels(); channel++)
  offset_measured[channel] = 0.0;
for (i=1; i<=(int16)CalAveragePoints; i++) {
  read_V208();
  for (channel=1; channel<=number_channels(); channel++)
    offset_measured[channel] += volt_data[channel];
  }
for (channel=1; channel<=number_channels(); channel++)
  offset_measured[channel] /= CalAveragePoints;

/* Step 4c          */
for (channel=1; channel<=number_channels(); channel++) {
  /* get offset coefficient for channel      */
  address = A24_V243 + 0x2040 + ((channel-1)*2);
  error = VXIin(A24SuperData,address,2,&offset_coef);
  if (error != 0 ) {
    printf("Error: VXIin() returned %d\n",error);
    CloseVXIlibrary();
    exit(1);
    }
  offset[channel] = (offset_measured[channel]/gain[channel]) + (offset_coef*1.0E-9);
  }

/* Set V243 Prefilter Gain registers         */
/* 10 Hz filters, PreGain=100, Set channel    */
/* to front panel connectors                  */
data = 0x2;
for (i=0; i<number_channels(); i+=8) {
  address = A24_V243 + 0x10 + i;
  error = VXIout(A24SuperData,address,2,data);
  if (error != 0 ) {
    printf("Error: VXIout() returned %d\n",error);
    CloseVXIlibrary();
    exit(1);
    }
  }

wait(2500); /* wait for 10 hz filter and cal voltage to settle */

/* Step 5         */
for (channel=1; channel<=number_channels(); channel++) {
  uncalibrated_voltage = 0.0;
  for (i=1; i<=(int16)AveragePoints; i++) {
    read_V208();
    uncalibrated_voltage += volt_data[channel];
    }
  uncalibrated_voltage /= AveragePoints;
  calibrated_voltage = (uncalibrated_voltage/gain[channel]) - offset[channel];
  printf("Channel %d      Voltage = %2.8f mVolts\n",channel,calibrated_voltage*1000.0);
  }

CloseVXIlibrary();
return;
}
```

```
/*************************************************************************/
/*                                                                       */
/*                                                                       */
/*  File Name : Subs.c                                                   */
/*                                                                       */
/*  This file contains the subroutines used by Module.c and Channel.c    */
/*                                                                       */
/*  Please Note: This file does not include                              */
/*  National Instruments routines (VXIin, VXIout, etc.)                  */
/*                                                                       */
/*************************************************************************/

typedef unsigned short uint16;
typedef short int16;
typedef unsigned long uint32;

int16 number_channels();
void read_V208();
void setup();

extern uint16   const     A24SuperData;
extern uint16   const     A32SuperData;
extern uint16   const     V243;
extern uint16   const     V208;
extern uint16   const     PostGainRegisterValue;

/* array for possible 96 channels */
extern float         volt_data[97];

/* Returns the number of channels (16, 32, 48, 64, or 96) depending     */
/* on the option indicated by the Suffix Register           */
int16 number_channels() {
 uint16    data, data2, offset;
 int16     error;

 offset = 0x22;
 error = VXIinReg(V243,offset,&data);
 if (error != 0 ) {
  printf("Error: VXIinReg() returned %d\n",error);
  CloseVXIlibrary();
  exit(1);
  }

 offset = 0x20;
 error = VXIinReg(V243,offset,&data2);
 if (error != 0 ) {
  printf("Error: VXIinReg() returned %d\n",error);
  CloseVXIlibrary();
```

```
      exit(1);
      }

    data >>= 8;
    data &= 0xFF;

    data2 >>= 8;
    data2 &= 0xFF;

    if (data == '1' && data2 ==‘Z’)
      return 48;
    else if (data == '2' && data2 == ‘Z’)
      return 96;
    else if (data == ‘1’ && data2 == ‘V’)
      return 16;
    else if (data == ‘3’ && data2 == ‘V’)
      return 32;
    else if (data == ‘4’ && data2 == ‘V’)
      return 48;
    else if (data == ‘6’ && data2 == ‘V’)
      return 64;
    else if (data == ‘9’ && data2 == ‘V’)
      return 96;
    else
      printf("Error: V243 option not found by number_channels() \n");

    return -1;
    }

/* Reads all channel's voltages and stores them in  */
/* global variable volt_data[channel]                */
void read_V208() {
  uint32    address,A32_V208;
  uint16    data,V243;
  int16     error,i;

  error = GetDevInfo(V208, 12, &A32_V208);
  if (error != 0 ) {
    printf("Error: GetDevInfo() returned %d\n",error);
    CloseVXIlibrary();
    exit(1);
    }

  /* read data from Ping Pong */
  for (i=1; i<=number_channels(); i++) {
    address = A32_V208 + 0x2FFE + (i*2);
    error = VXIin(A32SuperData,address,2,&data);
    if (error != 0 ) {
```

```
    printf("Error: VXIin() returned %d\n",error);
    CloseVXIlibrary();
    exit(1);
    }

   /* offset binary, convert to volts      */
   volt_data[i] = (data - 32768) * .0003125;
   }
  }

/* Setup MUX-bus for 48 or 96 channels depending on */
/* V243 option                                      */
void setup(){
 uint32   address,A32_V208,A24_V243;
 uint16   data;
 int16    error,i;

 error = GetDevInfo(V208, 12, &A32_V208);
 if (error != 0 ) {
  printf("Error: GetDevInfo() returned %d\n",error);
  CloseVXIlibrary();
  exit(1);
  }

 error = GetDevInfo(V243, 12, &A24_V243);
 if (error != 0 ) {
  printf("Error: GetDevInfo() returned %d\n",error);
  CloseVXIlibrary();
  exit(1);
  }

 /* V208 in setup mode */
 address = A32_V208 + 0x6;
 data = 0x0;
 error = VXIout(A32SuperData,address,2,data);
 if (error != 0 ) {
  printf("Error: VXIout() returned %d\n",error);
  CloseVXIlibrary();
  exit(1);
  }

 /* V243 in setup mode */
 address = A24_V243;
 data = 0;
 error = VXIout(A24SuperData,address,2,data);
 if (error != 0 ) {
  printf("Error: VXIout() returned %d\n",error);
  CloseVXIlibrary();
 exit(1);
 }

/* V208 set to number of V243 channels */
address = A32_V208 + 0x1000;
for (i=1; i<number_channels(); i++) {
 data = i-1;
 error = VXIout(A32SuperData,address,2,data);
 if (error != 0 ) {
  printf("Error: VXIout() returned %d\n",error);
  CloseVXIlibrary();
  exit(1);
  }
 address += 2;
 }
data = 0x8000 | (i-1);
error = VXIout(A32SuperData,address,2,data);
if (error != 0 ) {
 printf("Error: VXIout() returned %d\n",error);
 CloseVXIlibrary();
 exit(1);
 }

/* V243 set to number of V243 channels */
address = A24_V243 + 0x200;
for (i=1; i<number_channels(); i++) {
 data = 0x4000 | (i-1);
 error = VXIout(A24SuperData,address,2,data);
 if (error != 0 ) {
  printf("Error: VXIout() returned %d\n",error);
  CloseVXIlibrary();
  exit(1);
  }
 address += 2;
 }
data = 0xC000 | (i-1);
error = VXIout(A24SuperData,address,2,data);
if (error != 0 ) {
 printf("Error: VXIout() returned %d\n",error);
 CloseVXIlibrary();
 exit(1);
 }

/* V243 Postfilter gain to 20 */
address = A24_V243 + 0x100;
data = PostGainRegisterValue;
for (i=1; i<=number_channels(); i++) {
 error = VXIout(A24SuperData,address,2,data);
```

```
  if (error != 0 ) {
    printf("Error: VXIout() returned %d\n",error);
    CloseVXIlibrary();
    exit(1);
    }
  address += 2;
  }

/* start clock */
/* 1000 Hz -> 100 channels or less             */
/* will work for 100kHz and 200kHz V208s   */
address = A32_V208;
data = 0x48;
error = VXIout(A32SuperData,address,2,data);
if (error != 0 ) {
  printf("Error: VXIout() returned %d\n",error);
  CloseVXIlibrary();
  exit(1);
  }

address = A24_V243; /* V243 in run mode */
data = 0x20;
error = VXIout(A24SuperData,address,2,data);
if (error != 0 ) {
  printf("Error: VXIout() returned %d\n",error);
  CloseVXIlibrary();
  exit(1);
  }

address = A32_V208 + 0x6; /* V208 in run mode */
data = 0x1;
error = VXIout(A32SuperData,address,2,data);
if (error != 0 ) {
  printf("Error: VXIout() returned %d\n",error);
  CloseVXIlibrary();
  exit(1);
  }

}
```

```
/************************************************************************/
/*                                                                      */
/*                                                                      */
/*   Program Name : Module.c                                            */
/*                                                                      */
/*                                                                      */
/*   This program example is a C/C++ code example for the National      */
/*   Instruments controller.  This program will calibrate the V243      */
/*   for both offset and gain and store those errors in the V243's      */
/*   Correction Table.  Make sure cal strap is set to the enabled       */
/*   position (see Configuration section in the V243 manual for details). */
/*                                                                      */
/*   This program performs the module calibration as dictated           */
/*   by the steps previously explained in this manual.  This            */
/*   program assumes steps 1 and 2 are already complete.                */
/*                                                                      */
/*   This program was revised to reflect the simplified module          */
/*   calibration procedure.                                             */
/*                                                                      */
/*   V243 Logical Address = 5                                           */
/*   V208 Logical Address = 6                                           */
/*                                                                      */
/************************************************************************/
#include <stdlib.h>
#include <stdio.h>
#include <time.h>

typedef unsigned short uint16;
typedef short int16;
typedef unsigned long uint32;

/* header for subroutines described in subs.c              */
int16 number_channels();
void read_V208();
void setup();

/* Additional subroutines not listed in subs.c                  */
/* wait(n)     - subroutine will wait 'n' micro-seconds          */
/*               before returning.                               */
/* ReadDVM(&v)  - subroutine returns 'v' voltage read from       */
/*               precision volt meter                            */

uint16  const    A24SuperData = 0x6;
uint16  const    A32SuperData = 0x7;
uint16  const    V243 = 5;
uint16  const    V208 = 6;
```

```
/* Note: Offset calibration done ALWAYS at gain of 2000        */
/*      Offset coefficients stored in nVolts, Refereed to Input  */
/*      which is gain independent.                               */
float   const     Gain = 2000.0;
uint16  const     PostGainRegisterValue = 0x4;
uint16  const     PreGainRegisterValue = 0x2;
uint16  const     PosCalRegisterValue = 0x80A8;
uint16  const     NegCalRegisterValue = 0x8128;
float   const     AveragePoints = 1000.0;
uint16  const     GainCoefOffset = 0x2040;

/* array for possible 96 channels */
float           volt_data[97];

typedef struct{
     uint16   pos,neg;
     float    voltage;
     } CAL_TYPE;

void main()
 {
 uint16 data,offset;
 uint32 address,A24_V243,A32_V208;
 int16  error,i,j,k,channel;
 float  fp_offset[97],cal_offset[97];
 double voltage,pcal,ncal,calibrator,range,expect;

 struct tm *time_buffer;
 long   ltime;

 /* This array holds all of the calibration register settings  */
 CAL_TYPE cal[12] = { 0x091, 0x0111, 10.00,
                      0x0A1, 0x0121,  5.00,
                      0x0C1, 0x0141,  2.00,
                      0x092, 0x0112,  1.00,
                      0x0A2, 0x0122,  0.50,
                      0x0C2, 0x0142,  0.20,
                      0x094, 0x0114,  0.10,
                      0x0A4, 0x0124,  0.05,
                      0x0C4, 0x0144,  0.02,
                      0x098, 0x0118,  0.01,
                      0x0A8, 0x0128,  0.005,
                      0x0C8, 0x0148,  0.002 };


printf("\n\n           V243 Module Calibration\n\n\n");

InitVXIlibrary();
```

```
/* Get A24 base address for V243      */
error = GetDevInfo(V243, 12, &A24_V243);
if (error != 0 ) {
  printf("Error: GetDevInfo() returned %d\n",error);
  CloseVXIlibrary();
  exit(1);
  }

/* Get A32 base address for V208      */
error = GetDevInfo(V208, 12, &A32_V208);
if (error != 0 ) {
  printf("Error: GetDevInfo() returned %d\n",error);
  CloseVXIlibrary();
  exit(1);
  }

/* Step 4                             */
/* reset V243 Prefilter Gain registers         */
data = 0;
for (i=1; i<=12; i++) {
  address = A24_V243 + (0x8 * (i+1));
  error = VXIout(A24SuperData,address,2,data);
  if (error != 0 ) {
    printf("Error: VXIout() returned %d\n",error);
    CloseVXIlibrary();
    exit(1);
    }
  }

printf("\n\nCalibrator Gain Calibration\n");

for (i=0; i<12; i++) {
  printf("  Measuring voltage +- %f Volts\n",cal[i].voltage);

  /* Step 5a                                          */
  /* set positive calibration voltage, use on board source       */
  address = A24_V243 + 0x2;
  data = cal[i].pos | 0xA000;
  error = VXIout(A24SuperData,address,2,data);
  if (error != 0 ) {
    printf("Error: VXIout() returned %d\n",error);
    CloseVXIlibrary();
    exit(1);
    }
```

```
/* wait for voltage to settle */
wait(2500);
pcal = 0.0;
for (j=1; j<=10; j++) {
  ReadDVM(&voltage);
  pcal += voltage;
  }
pcal /= 10.0;

/* Step 5b                                                */
/* set negative calibration voltage, use on board source        */
address = A24_V243 + 0x2;
data = cal[i].neg | 0xA000;
error = VXIout(A24SuperData,address,2,data);
if (error != 0 ) {
  printf("Error: VXIout() returned %d\n",error);
  CloseVXIlibrary();
  exit(1);
  }

/* wait for voltage to settle */
ncal = 0.0;
wait(2500);
for (j=1; j<=10; j++) {
  ReadDVM(&voltage);
  ncal += voltage;
  }
ncal /= 10.0;

/* Step 5c                                                */
/* correction factor can be stored if error less than 3.27%    */
voltage = pcal - ncal;
expect = cal[i].voltage * 2.0;
range = expect * .03;
if ( voltage < expect-range || voltage > expect+range ) {
  printf("Error:  Calibrator voltage out of range to store gain correction value\n");
  printf("        Calibrator voltage @ +- %f volts\n",cal[i].voltage);
  printf("        Voltage(+CAL) = %2.8f volts,  Voltage(-CAL) = %2.8f volts\n",pcal,ncal);
  printf("        Volt calibrated = %2.8f volts +- %f, Volt Measured = %2.8f volts\n",
        expect,range,voltage);
  printf("        Difference = %2.8f \n",expect-voltage);
  exit(1);
  }
else {
  calibrator = cal[i].voltage;
  data = (uint16)( (((pcal-ncal) / (calibrator*2.0)) - 1.0) * 1000000.0);
  printf("    Coefficient stored = %d \n",data);
  address = A24_V243 + 0x2010 + (i*2);
```

```
   error = VXIout(A24SuperData,address,2,data);
   if (error != 0 ) {
       printf("Error: VXIout() returned %d\n",error);
       CloseVXIlibrary();
       exit(1);
       }

   wait(3);
   }
  }

 printf("\n\nCalibration Offset\n");

 /* Step 7a                              */
 /* set all channels to front panel      */
 data = PreGainRegisterValue;
 for (i=0; i<number_channels(); i+=8) {
  address = A24_V243 + 0x10 + i;
  error = VXIout(A24SuperData,address,2,data);
  if (error != 0 ) {
   printf("Error: VXIout() returned %d\n",error);
   CloseVXIlibrary();
   exit(1);
   }
  }

/* setup MUX-bus                                       */
/* all available channels (16, 32, 48, 64, or 96)      */
/* Step 7b                                             */
/* Set gain for all channels to a gain of 20           */
setup();

wait(2500); /* wait for 10 hz filter and cal voltage to settle */

/* Step 7c                                             */
/* Take offset measurements from front panel           */
for (channel=1; channel<=number_channels(); channel++)
 fp_offset[channel] = 0.0;
for (i=1; i<=(int16)AveragePoints; i++) {
 read_V208();
 for (channel=1; channel<=number_channels(); channel++)
  fp_offset[channel] += volt_data[channel];
 }
for (channel=1; channel<=number_channels(); channel++)
 fp_offset[channel] /= AveragePoints;
```

```
/* Step 7d                    */
/* set all channels to calibrator          */
data = 0x4000 | PreGainRegisterValue;
for (i=0; i<number_channels(); i+=8) {
 address = A24_V243 + 0x10 + i;
 error = VXIout(A24SuperData,address,2,data);
 if (error != 0 ) {
  printf("Error: VXIout() returned %d\n",error);
  CloseVXIlibrary();
  exit(1);
  }
 }

/* Step 7e                              */
/* set calibrator voltage to ground               */
data = 0x4000;
address = A24_V243 + 0x2;
error = VXIout(A24SuperData,address,2,data);
if (error != 0 ) {
 printf("Error: VXIout() returned %d\n",error);
 CloseVXIlibrary();
 exit(1);
 }

/* wait for 10 hz filter and cal voltage to settle */
wait(2500);

/* Step 7f                              */
/* Take offset measurements from calibrator            */
for (channel=1; channel<=number_channels(); channel++)
 cal_offset[channel] = 0.0;
for (i=1; i<=(int16)AveragePoints; i++) {
 read_V208();
 for (channel=1; channel<=number_channels(); channel++)
  cal_offset[channel] += volt_data[channel];
 }
for (channel=1; channel<=number_channels(); channel++)
 cal_offset[channel] /= AveragePoints;

/* Step 7g                    */
for (channel=1; channel<=number_channels(); channel++) {
 address = A24_V243 + 0x2040 + ((channel -1) * 2);
 data = (uint16)(((fp_offset[channel]-cal_offset[channel])/2000.0) * 1.0E9);
 printf("  Channel %d Offset Coefficient = %d nVolts \n",channel,data);
 error = VXIout(A24SuperData,address,2,data);
```

```
    if (error != 0 ) {
     printf("Error: VXIout() returned %d\n",error);
     CloseVXIlibrary();
     exit(1);
     }
    wait(3);
    }

  /* Step 8                              */
  time( &ltime );
  time_buffer= gmtime( &ltime );
  printf("\n\nRecording Date - %s\n",asctime(time_buffer));

 /* store month */
  address = A24_V243 + 0x2000;
  data = time_buffer->tm_mon + 1;    /* month = 1 to 12 */
  error = VXIout(A24SuperData,address,2,data);
  if (error != 0) {
     printf("Error: VXIout() returned %d\n",error);
     CloseVXIlibrary();
     exit(1);
     }
  wait(3);

  /* store day */
  address = A24_V243 + 0x2002;
  data = time_buffer->tm_mday;    /* day = 1 to 31 */
  error = VXIout(A24SuperData,address,2,data);
  if (error != 0) {
     printf("Error: VXIout() returned %d\n",error);
     CloseVXIlibrary();
     exit(1);
     }
  wait(3);

  /* store year */
  address = A24_V243 + 0x2004;
  data = time_buffer->tm_year + 1900;     /* year = 1990 ...  */
  error = VXIout(A24SuperData,address,2,data);
if (error != 0) {
    printf("Error: VXIout() returned %d\n",error);
    CloseVXIlibrary();
    exit(1);
    }
  wait(3);
```

```
CloseVXIlibrary();
return;
}
```

APPENDIX C- Additional Specifications

Offset and Gain Accuracy of the V243-ZAxx Using the V208

Gain	LSB	Offset Typical	Offset Max	Gain Typical	Gain Max
1 (t)	312.5 µV	450 µV	900 µV (q)	0.005 %	0.010 %
2	156.25 µV	225 µV	450 µV (q)	0.005 %	0.010 %
5	62.5 µV	90 µV	180 µV (q)	0.005 %	0.010 %
10 (t)	31.25 µV	45 µV	90 µV (q)	0.005 %	0.010 %
20	15.625 µV	22.5 µV	45 µV (q)	0.005 %	0.010 %
50	6.25 µV	9 µV	18 µV (q)	0.005 %	0.010 %
100 (t)	3.125 µV	4.5 µV	9 µV (q)	0.005 %	0.010 %
200	1.5625 µV	2 µV	4 µV (q)	0.005 %	0.010 %
500	625 ηV	2 µV	4 µV	0.005 %	0.020%
1000	312.5 ηV	1 µV	2 µV	0.015 %	0.040 %
2000 (t)	156.25 ηV	1 µV	2 µV	0.015 %	0.040 %

Offset and Gain Accuracy of the V243-ZBxx and V243- VAxx Using the V208

Gain	LSB	Offset Typical	Offset Max	Gain Typical	Gain Max
1 (t)	312.5 µV	450 µV	900 µV (q)	0.025 %	0.040 %
2	156.25 µV	225 µV	450 µV (q)	0.025 %	0.040 %
5	62.5 µV	90 µV	180 µV (q)	0.025 %	0.040 %
10 (t)	31.25 µV	45 µV	90 µV (q)	0.025 %	0.040 %
20	15.625 µV	22.5 µV	45 µV (q)	0.025 %	0.040 %
50	6.25 µV	9 µV	18 µV (q)	0.025 %	0.040 %
100 (t)	3.125 µV	4.5 µV	9 µV (q)	0.025 %	0.040 %
200	1.5625 µV	2 µV	4 µV (q)	0.025 %	0.040 %
500	625 ηV	2 µV	4 µV	0.025 %	0.040 %
1000	312.5 ηV	1 µV	2 µV	0.025 %	0.040 %
2000 (t)	156.25 ηV	1 µV	2 µV	0.025 %	0.040 %

Note: Gains indicated with a '(t)' are gains tested to the listed accuracy during the production test and are therefore listed in the V243 data sheet. All other offset/gain accuracy specifications are expected, but not guaranteed. Gains indicated with a '(q)' are gains in which the offset error is dominated by quantization error of the V208.

V243 Configuration Registers, A16 Space

	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00	Page
00_{16} (r)	0	1	0	0	1	1	1	1	0	0	1	0	1	0	0	1	24
00_{16} (w)	Logical Address Register																24
02_{16}	1	1	0	1	0	0	1	0	0	1	0	1	0	0	1	0	24
04_{16}	A24	Modid											Ready			Reset	25
06_{16}	Offset Register																26
08_{16}	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	0	26
$0A_{16}$	Serial Number High																27
$0C_{16}$	Serial Number Low																27
$0E_{16}$	Fver3	Fver2	Fver1	Fver0	Frev3	Frev2	Frev1	Frev0	Hver3	Hver2	Hver1	Hver0	Hver3	Hver2	Hver1	Hver0	28
10_{16}-18_{16}	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	28
$1A_{16}$	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	28
$1C_{16}$	1	1	1	1	1	1	1	OVER MASK	*IREN	1	IRQ2	IRQ1	IRQ0	1	1	1	29
$1E_{16}$	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	30
20_{16}	Suffix High Register																30
22_{16}	Suffix Low Register																30
24_{16}-$3E_1$	User Defined Register																31

V243 Operational Registers, A24 Space

	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00	Page
00_{16}	1	1	1	1	CON3	CON2	CON1	CON0	Therm	Ovrlp	Adsel						32
02_{16}	REFS	GND	CalOu	1	1	1	1	CAL8	CAL7	CAL6	CAL5	CAL4	CAL3	CAL2	CAL1	CAL0	32
04_{16}	1	1	1	1	1	1	1	1	1	1	1	1	Out73	Out49	Out25	Out1	33
06_{16}-$0E_{16}$	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
10_{16}	Channel 1-8 Prefilter Gain Register																34
12_{16}-16_{16}	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
18_{16}	Channel 9-16 Prefilter Gain Register																34
$1A_{16}$-$1E_{16}$	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
20_{16}	Channel 17-24 Prefilter Gain Register																34
22_{16}-26_{16}	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
28_{16}	Channel 25-32 Prefilter Gain Register																34
$2A_{16}$-$2E_{16}$	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
30_{16}	Channel 33-40 Prefilter Gain Register																34
32_{16}-36_{16}	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
38_{16}	Channel 41-48 Prefilter Gain Registers																34
$3A_{16}$-$3E_{16}$	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
40_{16}	Channel 49-56 Prefilter Gain Registers																34
42_{16}-46_{16}	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
48_{16}	Channel 57-64 Prefilter Gain Registers																34
$4A_{16}$-$4E_{16}$	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
50_{16}	Channel 65-72 Prefilter Gain Registers																34
52_{16}-56_{16}	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
58_{16}	Channel 73-80 Prefilter Registers																34
$5A_{16}$-$5E_{16}$	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
60_{16}	Channel 81-88 Prefilter Registers																34
62_{16}-66_{16}	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
68_{16}	Channel 89-96 Prefilter Registers																34
$6A_{16}$-FE_{16}	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	

WARRANTY

KineticSystems Company, LLC warrants its standard hardware products to be free of defects in workmanship and materials for a period of one year from the date of shipment to the original end user. Software products manufactured by KineticSystems are warranted to conform to the Software Product Description (SPD) applicable at the time of purchase for a period of ninety days from the date of shipment to the original end user. Products purchased for resale by KineticSystems carry the original equipment manufacturer's warranty.

KineticSystems will, at its option, either repair or replace products that prove to be defective in materials or workmanship during the warranty period.

Transportation charges for shipping products to KineticSystems shall be prepaid by the purchaser, while charges for returning the repaired warranty product to the purchaser, if located in the United States, shall be paid by KineticSystems. Return shipment will be made by UPS, where available, unless the purchaser requests a premium method of shipment at their expense. The selected carrier shall not be construed to be the agent of KineticSystems, nor will KineticSystems assume any liability in connection with the services provided by the carrier.

The product warranty may vary outside the United States and does not include shipping, customs clearance, or any other charges. Consult your local authorized representative or reseller for more information regarding specific warranty coverage and shipping details.

PRODUCT SPECIFICATIONS AND DESCRIPTIONS IN THIS DOCUMENT SUBJECT TO CHANGE WITHOUT NOTICE.

KINETICSYSTEMS SPECIFICALLY MAKES NO WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE OR ANY OTHER WARRANTY EITHER EXPRESSED OR IMPLIED, EXCEPT AS IS EXPRESSLY SET FORTH HEREIN. PRODUCT FAILURES CREATED BY UNAUTHORIZED MODIFICATIONS, PRODUCT MISUSE, OR IMPROPER INSTALLATION ARE NOT COVERED BY THIS WARRANTY.

THE WARRANTIES PROVIIDED HEREIN ARE THE PURCHASER'S SOLE AND EXCLUSIVE REMEDIES ON ANY CLAIM OF ANY KIND FOR ANY LOSS OR DAMAGE ARISING OUT OF, CONNECTED WITH, OR RESULTING FROM THE USE, PERFORMANCE OR BREACH THEREOF, OR FROM THE DESIGN, MANUFACTURE, SALE, DELIVERY, RESALE, OR REPAIR OR USE OF ANY PRODUCTS COVERED OR FURNISHED BY KINETICSYSTEMS INCLUDING BUT NOT LIMITED TO ANY CLAIM OF NEGLIGENCE OR OTHER TORTIOUS BREACH, SHALL BE THE REPAIR OR REPLACEMENT, FOB FACTORY, AS KINETICSYSTEMS MAY ELECT, OF THE PRODUCT OR PART THEREOF GIVING RISE TO SUCH CLAIM, EXCEPT THAT KINETICSYSTEMS' LIABILITY FOR SUCH REPAIR OR REPLACEMENT SHALL IN NO EVENT EXCEED THE CONTRACT PRICE ALLOCABLE TO THE PRODUCTS OR PART THEROF WHICH GIVES RISE TO THE CLAIM. IN NO EVENT SHALL KINETICSYSTEMS BE LIABLE FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, INCLUDING LOSS OF PROFITS.

Products will not be accepted for credit or exchange without the prior written approval of KineticSystems. If it is necessary to return a product for repair, replacement or exchange, a Return Authorization (RA) Number must first be obtained from the Repair Service Center prior to shipping the product to KineticSystems. The following steps should be taken before returning any product:

1. Contact KineticSystems and discuss the problem with a Technical Service Engineer.
2. Obtain a Return Authorization (RA) Number.
3. Initiate a purchase order for the estimated repair charge if the product is out of warranty.
4. Include a description of the problem and your technical contact person with the product.
5. Ship the product prepaid with the RA Number marked on the outside of the package to:

KineticSystems Company, LLC
Repair Service Center
900 North State Street
Lockport, IL 60441

Telephone: (815) 838-0005
Facsimile: (815) 838-4424
Email: tech-serv@kscorp.com