Model V213

32/64-Channel 16-bit Scanning ADC

INSTRUCTION MANUAL

January 30, 2001

CONTENTS

V213 QUICK REFERENCE GUIDE

APPENDIX

FIGURES

TABLES

Getting Started

This manual is organized into four sections:

- The *Introduction* section describes the capabilities of the V213 and how it can be used in a wide variety of applications.

- The *Unpacking and Installation* section explains how to install the V213 into a VXI chassis. In addition, a procedure for installing field upgrades for the V213 is also provided.

- The *Programming Information* section explains how to access and control the V213. The first half of the section explains the use of the Configuration Registers. These general purpose registers are standard registers defined by VXI which are used to identify the module and control interrupts. The second half of the section explains the use of the Operational Registers. The registers are specific for operation and control of the V213. These registers are used to control scan rates, scan lists, TTL trigger lines, and etc.

- The *Appendix* provides additional information that may be helpful in the use of the V213. Topics include register layout, module calibration procedures, grounding techniques when connecting field wiring, and programming examples.

The following is a list of some of the terms used throughout this manual:

A16 Space - As described in the VXI specification, this mnemonic is used to describe the first 64 kBytes of address space. Every VXI module is automatically allocated a 64 byte block of this address space (also known as Configuration Registers). The exact location is determined by the logical address of the module.

A32 Space - As described in the VXI specification, this mnemonic is used to describe the 4 Giga-byte block of address space provided. Any module can request a block of this address space from information contained in its Configuration Registers. This memory block is also called the Operational Registers.

Configuration Registers - See A16 Space.

D16 - As described in the VXI specification, this mnemonic is used to describe a single 16-bit data transfer.

D16 BLK - As described in the VXI specification, this mnemonic is used to describe a 16-bit block transfer.

D32 - As described in the VXI specification, this mnemonic is used to describe a single 32-bit data transfer. This mode is not supported by all Slot-0 controllers. Check the owners manual for the Slot-0 before attempting this type of transfer.

D32 BLK - As described in the VXI specification, this mnemonic is used to describe a 32-bit block transfer. This is the fastest transfer type supported by the V213. This mode is most beneficial when retrieving data from the ADC (i.e. through Ping/Pong). However, this mode may not be supported by the Slot-0 controller. Check the owners manual for the Slot-0 before attempting this type of transfer.

Logical Address - Every VXI module is given a unique logical address. There are a total of 256 logical addresses with 0 reserved for the Slot-0 controller.

Operational Registers - See A32 Space.

Resource manager - Also referred to as Resman. This software is made available by the manufacturer of the Slot-0 controller and is used to set logical addresses and Operational Registers of all the modules in the system.

R.T.I. - Referred To Input. In any ADC system that has gain (like the V213), a voltage reading RTI refers to the point where the signal enters the module. It is the reading before gain is applied.

R.T.O. - Referred To Output. In any ADC system that has gain (like the V213), a voltage reading RTO refers to the point where gain has been applied to the signal entering the ADC. It is the reading after gain is applied.

Scan List - This list indicates the order in which the V213 converts each channel's voltage.

Scan Rate - This rate indicates the speed at which the V213 is reset to the first item in its scan list.

Section I. Introduction

V213

The V213 base module is a 32 channel, 50 kHz scanning ADC. Programmable gains of 1 to 2000 are provided in a 1, 2, 5 progression. Fixed, 2-pole RC filters are available on all options. Cutoff frequencies of 10 Hz through 1 kHz are available in a 1, 2, 5 progression. The V213 has a fixed point, digital signal processor (DSP) to perform self tests, provide calibration, and check for limit violations.

In addition, the V213 will accept I/O and digital expansion cards to enhance its capabilities. An I/O expansion card may be added to provide an additional 32 channels of fixed filters (10 Hz through 1kHz). Other types of cards are available to provide 32 channels of analog output (DAC), 64 channels of TTL I/O, or an option that provides 16 channels of analog output and 32 channels of TTL I/O. These cards are available as factory or field upgrades. In addition, future options of the V213 will provide digital expansion cards as well. These options will enhance the storage and capabilities of the V213. Some of these options include 1 or 4 Meg multibuffer cards, Digi-bus cards, and an additional DSP card. Although these options will only be available as a factory upgrade, the current V213 has been designed with these cards in mind. As these new options are made available, any V213 may be sent back to the factory to be upgraded.

The V213 is a single-width, C-size, register-based, VXIbus module which takes advantage of many of the features of the VXI standard. The VXI C-size specification, while it uses the same bus specification as VME, provides some significant system-level enhancements. These include the definition of 8 TTL and 2 ECL trigger lines, a 10 MHz ECL clock for system timing, module ID line for geographic addressing and module identification, an analog summing bus, and a 12-line daisy-chained Local Bus. These portions of the VXI standard provide a solid basis for a comprehensive systems-level approach.

VXI Trigger Lines

Eight of the pins on the P2 connector of the VXI bus are defined as TTL trigger lines. These open collector lines provide a wired-OR function that is suited to their use in communicating event information between modules.

A practical concept in using these lines is that of event sources and event sinks. Any one of several modules may generate an event on a specific trigger line, and one or more modules may be programmed to respond to this event, including the module which is the source of the event.

Examples of event sources are scan rate, limit checking and front-panel external trigger inputs. Examples of event sinks are transient capture triggers to initiate the capture and local storage of a data segment, and sample clocks to synchronize sampling of input signals or outputting of DAC or digital data.

Consider the case where several V213s are required to trigger simultaneously. The occurrence of a limit condition on any of the modules can cause a trigger line to be asserted and a transient capture initiated across all modules. Meanwhile, another trigger line has been programmed as the source for the scan rate.

Overall Architecture

Trigger lines can be used to synchronize sampling across multiple modules. A precision reference from the ADC, or an external calibration signal can be used for calibration and end-to-end system checkouts.

Ping-pong registers hold the data from the previous complete data scan for all channels. The set of registers accessible from the VXI bus are loaded synchronously with the start of a scan and are valid until the start of the next scan. This technique provides the full interval between scans for data access.

The multibuffer memory option provides critical buffering between the Slot-0 processor and the ADC to prevent data loss due to processor latency when switching buffers and processing data. The multibuffer memory also can provide a transient capture function where a fixed block of data is captured in memory based on the occurrence of a trigger event. Possible trigger events might include an external trigger or one or more of some selected analog inputs exceeding a programmed threshold. The multibuffer option is a future factory upgrade for the V213.

The Digi-bus option provides a private bus to stream data to processing elements such as a DSP which can perform functions such as FFT, digital filtering, or signal averaging. Trigger lines provide information such as the start of a transient and the start of an ADC scan. The Digi-bus option is also a future factory upgrade for the V213.

System Calibration/Diagnostics

In any system, particularly with larger data acquisition systems, verifying the proper functioning of the system and the calibration of analog I/O is essential. Generally, it is possible to check out some digital system components through exercising the hardware under software control.

To audit analog channels to any degree requires that a series of known analog signals be injected into each channel of the system. This can be accomplished either by operator intervention or by providing the capability to switch known calibration signals into the input under software control. Switching of calibration signals by hand can be very time-consuming and is subject to errors.

For these reasons, KineticSystems has chosen an implementation with full end-to-end calibration features. To accomplish this goal with the architecture described, a known signal must be injected at the input to the signal conditioning. This is accomplished by a programmable active attenuator or an external user-supplied front-panel calibration signal.

In addition to the software access to calibration signals, a fixed-point DSP is provided on all options for calibration and diagnostics of the V213.

System Configuration Validation

One of the issues, particularly in larger systems, is ensuring that the proper modules with the proper options are installed and configured in the system. This is especially important when you cannot accurately validate the current system configuration because of system size or physical access to areas in a distributed system. Verifying proper installation can be particularly frustrating when modules of the same type or model number can be configured with a number of internal options. Unfortunately, this is an increasingly common practice because of the high densities that can be achieved today and the relatively large card size of VXI. A related issue is the capability to identify and trace the history of specific modules within a larger system.

The VXI standard requires (or in some cases, suggests) that certain register conventions be followed. These register conventions partially address these issues. Standardized registers include the manufacturer ID assigned by the VXI Consortium, a device-type or model identifier, a serial number, and a version number or revision number for hardware and firmware.

An extension of this concept provides a module option identifier as well as some amount of user-writable EEPROM. The EEPROM provides the capability to record in nonvolatile memory any option, calibration, or other module-specific information that may be important to system operation and/or maintenance.

Since these registers are accessible by software, it is possible to develop software to verify system configuration at startup, as well as track modules for maintenance purposes.

Private Digital Path, Digi-bus

The VXI Local Bus provides 12 user-defined lines that are daisy-chained to adjacent modules. For digital signals, these lines provide a simple way to implement private data paths between system components that follow natural data flow, such as between an ADC and a digital signal processor (DSP) and/or a large block of memory.

In many applications, there is a requirement to perform FFTs, digital filtering, and signal averaging. Other applications include a need to store huge amounts of data.

The VXI Local Bus again provides a convenient mechanism to implement a private digital bus (Digi-bus) for moving synchronous data between modular elements of the system. By using a private synchronous bus as opposed to the main VXI backplane bus, issues regarding bus contention and latency are avoided.

For example, the bus interface can be implemented on an ADC as a data source while modules such as a DSP or memory module are implemented as data sinks. Figure 1 on page 14 illustrates a couple of examples on how Digi-bus could be used to interconnect a DSP or memory module.

The selection of a bus protocol is important. Multiple sources and sinks as well as device addressing must be defined. As protocol complexity and general-purpose utility grows, effective bandwidth drops and bus latency increases. Since this is a dedicated private bus, an effective solution is a simple synchronous protocol. This protocol may be based on the concept of a frame of data where each data source has pre-assigned locations within a frame to place data, and data sinks can extract data from known locations within a frame. One data source acts as the master, providing any necessary timing signals. The entire data frame is available to all modules connected to the Digi-bus segment. Since the Local Bus is propagated from slot to slot, multiple Digi-bus segments may co-exist in a common VXI chassis. Data frames are generated synchronously with the data acquisition or sample clock.

Since a particular sink module may only require selected data from a frame or, in some cases, only every Nth sample, a data selection scheme must be provided. Digi-bus implements a simple bit map of the frame that selects data based on position within a frame.

The direction for data flow is less obvious. The key element here is the VXI Slot-0 controller, which is defined by the standard to be in the left-most slot. Since the Slot-0 is either a computer in its own right (smart controller) or connected to a computer, the possibility of passing digital Local Bus data directly to the Slot-0 should not be precluded. As a result, the Digi-bus must exit to the left.

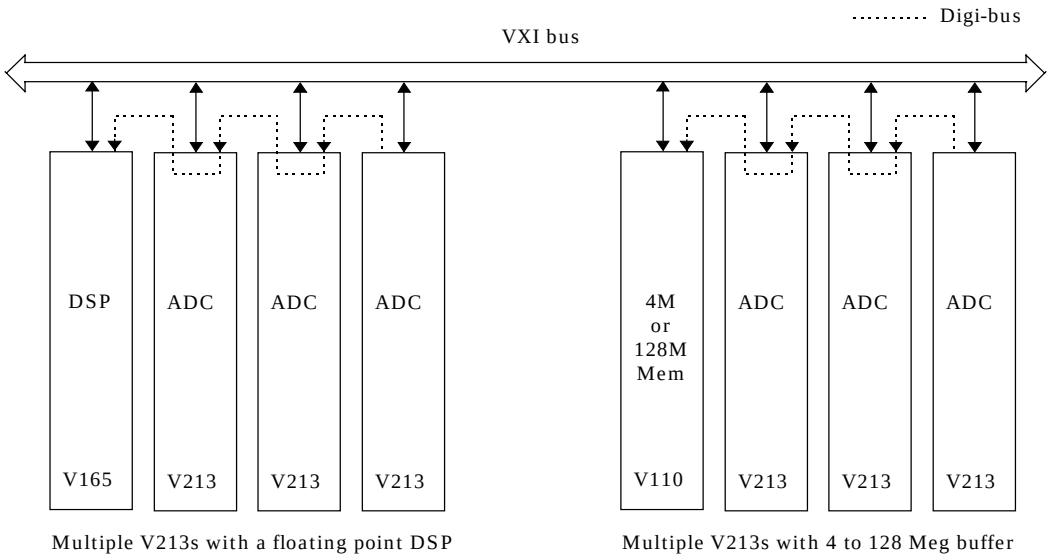


Figure 1 - Typical Digi-bus Configuration

Conclusions

The V213 is a single slot solution with multiple options for a wide variety of applications. Multiple options for signal conditioning are available including analog in, analog out, and TTL I/O. Multiple storage options are available as a single slot solution. A simple ping/pong buffer is available on all options. Also available are 1 or 4 Meg buffer cards for increased storage capacity.

For larger systems, the V213 has other options available. The use of the VXI Local Bus for a private synchronous digital bus provide a powerful tool for high-performance data acquisition and control. The Digi-bus provides a guaranteed synchronous data flow path between input, processing and buffering, and output elements of the system. Flexible use of trigger lines allows you to maintain critical synchronism between different system components.

Section II. 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 V213 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.

Configuration

There is one set of user configurable switches on the V213. 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 V213 is a statically configured device, it is the users responsibility to unsure that no two modules are set to the same logical address. If the module is set for logical address 255 (all switches open), then the V213 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.

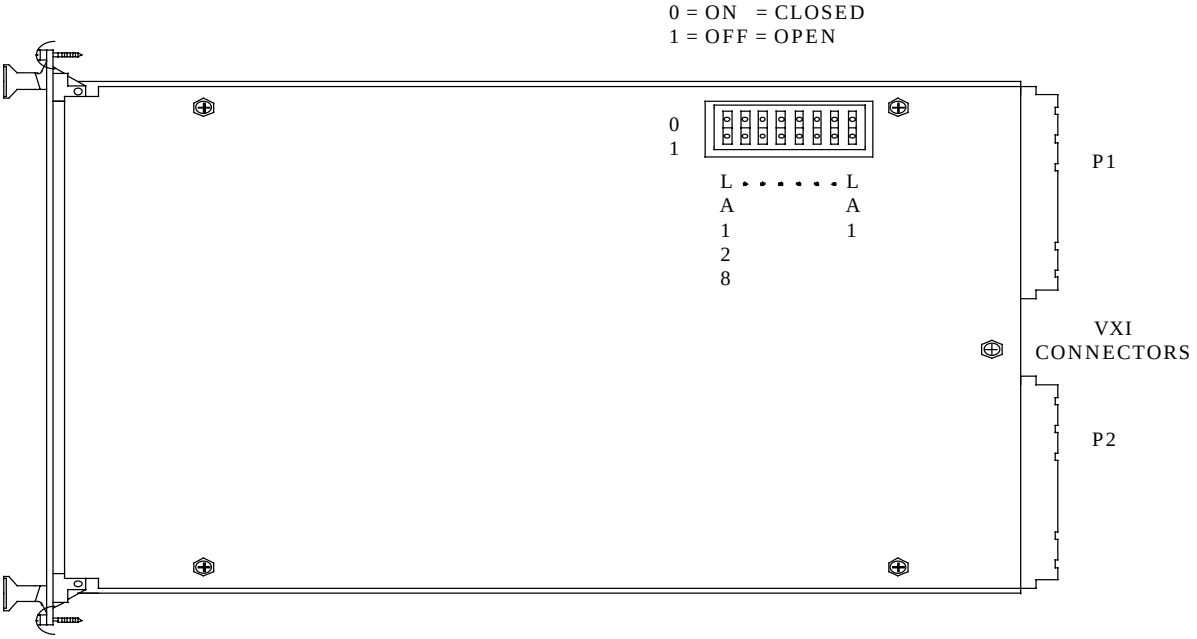


Figure 2 - V213 Switch Locations

I/O Expansion Cards

The V213 has I/O expansion cards available to increase the number of channels or add additional functionality to the card. These options may be purchased as factory or field upgrades. In the case of factory upgrades, the I/O expansion card is tested and integrated with a base card of the V213. Usually, the base card of the V213 is ordered with the factory upgrade. However, the base card may also be returned to the factory for integration. In the case where the base card has already been purchased, this factory upgrade option is recommended. However, a field upgrade kit may also be purchased.

In the case of the field upgrade, the I/O card is tested and shipped as a separate unit. Two surface mount IC's (PROMs) are provided with the latest version of firmware for the DSP. The new PROMs are not required for operation with the I/O card. Rather, it gives the DSP the capability to support the I/O card with self test algorithms and calibration (if necessary).

Refer to figure 3 while installing the field upgrade of the I/O expansion card.

1.) Once the shield has been removed, the sockets should be visible just below the heatsinks. Compare the label on the PROMs currently installed with the one that came with the upgrade kit. If the labels are identical, then the current PROMs do not need to be removed and step 2 can be ignored.

2.) Carefully remove the current PROMs using an extraction tool for 32 pin PLCC devices. Install the new PROMs according to figure 3. Pay special attention as to which IC goes into which socket and its orientation. Once placed in the socket, pin one of the IC should face toward the top of the board and the label should be right-side up.

3.) Place the I/O expansion card into the three inter-board connectors located on the main card. The inter-board connectors are shrouded and keyed to help prevent a misalignment of pins. Four screws are provided with the field upgrade kit to securely fit the card into place. Place the shield back over the card.

4.) The V213 will recognize the I/O expansion card and perform a selftest. If the upgrade was successful, the failed LED will turn off approximately five seconds after SYSRESET is deserted.

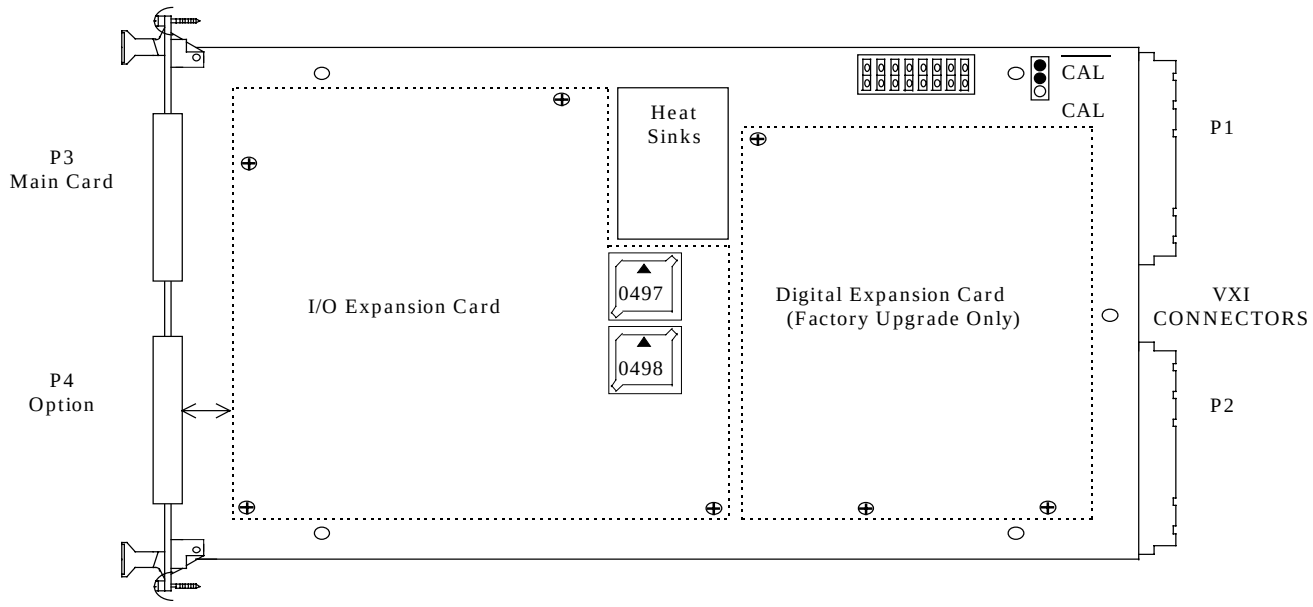


Figure 3 - V213 Upgrades and Strap Locations

Digital Expansion Cards

The V213 is capable of accepting future digital expansion cards to add additional functionality to the card. These options may only be purchased as a factory upgrade. Although the V213 does not yet offer any of these upgrades, the V213 was designed with several upgrades in mind. These upgrades include a 1 Meg multibuffer, 4 Meg multibuffer, Digi-bus, and DSP cards.

Module Insertion

The V213 is a C-sized, single width, VXIbus module. Except for Slot-0, it can be mounted in any unoccupied slot in a C-size VXIbus mainframe.

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 V213. To insure proper interrupt acknowledge cycles from the V213 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 V213 and the Slot-0 Controller.

Once the V213 is placed in the appropriate slot, make sure the screws located on the front panel are screwed into the mainframe. The shield of the high density connectors are connected to the front panel and ground to provide additional protection from static electricity. To provide maximum protection, the front panel should be screwed into the chassis. Otherwise, the static electricity must discharge through the V213 potentially causing unexpected behavior.

Termination of Unused Channels

As with most scanning ADCs, the V213 multiplexes all of its input channels down to a single gain path ahead of the ADC. Gain can be set on a per channel basis via a GAIN RAM that is synchronized with channel scanning. While this type of architecture reduces the high cost of having an independent gain circuit for each channel, it does require the user to take a few precautions. If unused channel inputs are left unterminated, it is possible that data for the channel in SCAN RAM that follows an unterminated channel will be affected. This is because the gain instrumentation amp on the V213 cannot recover from saturation before the next channel is converted. This is primarily a problem at high gain settings and sample rates. **It is therefore recommended that all unused channels be terminated to ground.**

It should also be noted that a fault condition such as an open thermocouple can cause saturation of the V213 gain amplifier as well. It is possible to use the DSP limit checking function to identify a saturated channel and switch the input MUX for this channel on the V213 to ground. This will prevent channel crosstalk from occurring. See page 68 for Limit Checking Opcodes.

V750-Isothermal Terminal Assembly

The V750-ZC11 is an isothermal termination assembly specifically designed for the V213. This assembly is capable of terminating the base 32 channels and the I/O expansion cards channels. The isothermal reference is connected to the isothermal channel located on the J3 connector. Channel 1 of the V213 may be configured as the isothermal reference channel by setting bit 9 of the Control Register (Operational Register in A32 space at offset 00_{16} page 44). In this case, the gain for channel 1 must be set to 20 by writing 4_{16} to the first location in Gain RAM at A32 space offset 300_{16}. Procedures for taking calibrated measurements (included the isothermal reference channel) is provided on page 39.

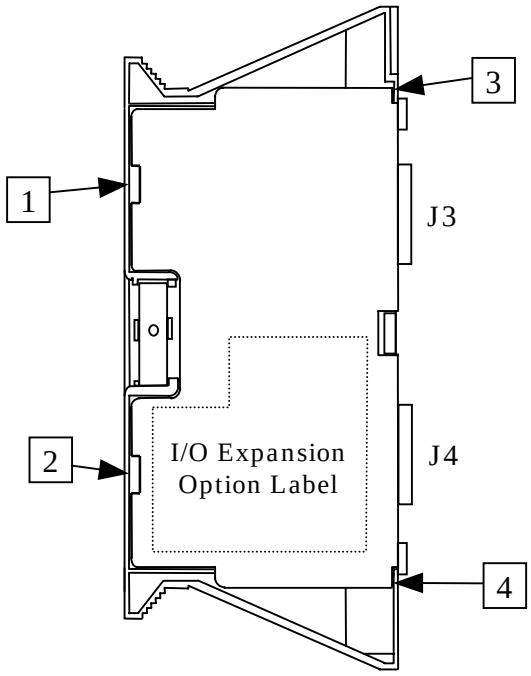
If the termination assembly is to be used with an I/O expansion card, a label is provided with the expansion card to be placed on the cover of the termination assembly. Refer to Figure 4 for the placement of the label on the termination assembly.

In cases where the V213 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 lift easily off of 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 4 -V750-ZC11 Termination Assembly



Front Panel Description

LEDs

After power up, the Failed LED will be on for at most five seconds while self tests are being run. During this time, the ADC, DSP, and DIG LEDs may also turn on periodically. If the Failed LED stays on after five seconds, then the self test failed. When self tests are complete, all other LEDs will be off.

Failed LED	The LED will turn on whenever self tests are run or if the scan rate is set too high (page 44).
Add Rec LED	Add Rec (address received) LED lights to indicate the V213 is being accessed through VXI.
ADC LED	The ADC LED will light while the V213 is storing data in its dual-ported memory (ping-pong). Specifically, this means that the V213 is receiving a clock, running through its scan list, and storing the data into its ping-pong buffer.
Int Src LED	The Int Src LED will light while the V213 is requesting an interrupt.
DSP LED	The DSP LED will turn on while the DSP is either performing self tests or is checking for limit violations from the ADC.
DIG LED	The DIG (digital expansion active) LED lights to indicate that the digital expansion card is storing data from the ADC.

Connectors

The V213 has two 68 pin, high density connectors. The P3 connector has 32 differential inputs as well as an isothermal reference input. The connector labeled P3 connects to channels 1 through 32. The P4 connector is routed to the I/O expansion card. The pinout description of this connector depends on the option of the I/O expansion card being used. See Figure 5 and Table 4 (Pages 61 and 64) for the precise pinout descriptions. Two SMBs are provided as an external Scan and Trig input. The Scan SMB can be used to provide the V213 with an external scan input. The V213 can route this signal to one of the 8 TTL trigger lines available on VXIbus to synchronize with other V213s. The Trig SMB can be used to provide the V213 with an external trigger input. This trigger is used to trigger data storage of the digital expansion card. This signal can also be routed to one of the 8 TTL trigger lines on VXIbus. A 2-pin LEMO is used to provide an external calibration input/output. The V213 has its own internal precision source used during calibration. The V213 does, however, have the capability to output its internal source through the 2-pin LEMO. In this case, a precision meter can be used as the standard to measure the internal source. In other cases, if a precision source is to be used, the 2-pin LEMO can be configured as an input. In this way a path is provided to calibrate channels using an external source.

Section III. Programming Information

VXIbus Addressing

The V213 is classified as an extended register device which means it has registers that occupy A16 and A32 space. In accordance with the VXI specification, A16 space means that only 16 address bits are decoded by the V213. A32 space means that 32 address bits are decoded by the V213.

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 A32 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:	V213
Base Card Option:	AAA1
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 page 24.

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

As defined by the VXI specification, these Configuration Registers will support D16 transfers only.

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 binary coded decimal (BCD) number 213, this indicates that the module is a V213.

Base Card Option: can be found by reading the four character string in registers located at offsets 20_{16} and 22_{16}. If the string "AA" is found at offset 20_{16} and "A1" is found at offset 22_{16}, then the specific type of module is the V213-AAA1 meaning the this module has 10 Hz filters in channels 1 through 32. 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 binary coded decimal (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. However, once they have been written, the data is retained even after power has been removed.

The Operational Registers are located in A32 space and include the registers specific to V213 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 (Configuration register in A16 space at offset 6_{16}). A complete list of A32 register descriptions and their offsets is available starting at page 44.

$$A32\_ADDRESS = \left(OFFSET\_REGISTER\_VALUE \cdot 10000_{16}\right) + A32\_REGISTER\_OFFSET$$

All Operation Registers support D16 transfers. Depending on the register, D16 BLK, D32, and, D32 BLK may also be supported. These capabilities are marked under each of the registers descriptions in this manual.

V213 Configuration Registers, A16 Space

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)	(1)	(0)	(1)	(0)	(0)	(1)	(1)	(1)	(1)	(1)	(0)	(0)	(1)

( ) Power Up Value

On READ transactions the V213 returns $5F29_{16}$.

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/A32 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)	(0)	(1)	(1)	(1)	(0)	(0)	(1)	(0)	(0)	(0)	(0)	(1)	(0)	(0)	(1)	(1)

( ) Power Up Value

This READ ONLY register returns 7213_{16}.

Bit(s)	Mnemonic	Meaning
15-12	Required Memory	The V213 requires 8 Meg bytes of additional memory space.
11-00	Model Code	Identifies this device as a V213 (213_{16}).

Status/Control Register

04_{16}	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
(Mixed)	A32 ENA	Modid											Ready	Pass	Sys Inb.	Soft Reset
	(0)	(0)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(0)	(0)	(0)	(0)

( ) Power Up Value

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

Bit(s)	Mnemonic	Meaning
15	A32 ENA	This bit is written with a "1" to enable A32 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 V213.
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	Pass	A "0" indicates the V213 has failed or is currently executing its self test. A "1" in this bit indicates the module's self test has passed.
01	Sys. Inb.	(Sysfail Inhibit) Writing a "1" to this bit disables the V213 from driving the SYSFAIL* line. Reads of this bit indicate its current state.
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 V213 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)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)

( ) Power Up Value

After SYSRESET* and prior to self test all bits are reset to "0". Otherwise, a read or write defines the base address of the module's A32 registers. Bits 07-00 of this register are not used and should always be written with zeros. As shown above bits 15-08 map directly onto VXI address lines A31-A24. For example, if bits OF15-0F08 contain 12_{16} the base address for the module's Operational Registers becomes 12000000_{16}.

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 V213 is capable of generating interrupts.
01	IH*	This bit is read as a "1" and indicates the V213 is not capable of Interrupt Handler Control.
00	IS*	This bit is set to "0" to indicate the V213 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

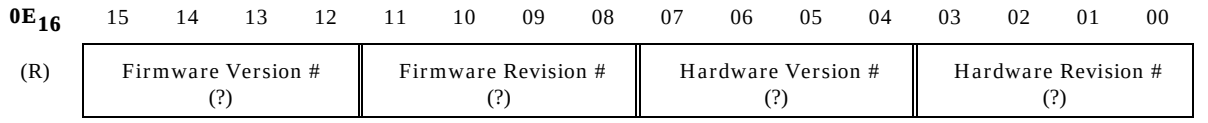
Serial Number Low

$0C_{16}$	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



( ) 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.

Interrupt Status Register

$1A_{16}$	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
(R)	IO EXP (0)	DIG EXP (0)	DSP IO (0)	DSP ALM (0)	END SCAN (0)	TTL TRIG (0)	EXT TRIG (0)	ERR (0)	Logical Address **(FF_{16})**							

( ) Power Up Value

This READ ONLY register is defined as follows:

Bit(s)	Mnemonic	Meaning
15	IO EXP	I/O Expansion - This bit is set to a "1" by the I/O expansion card. The meaning of this interrupt status bit is dependent on the expansion card used. Clearing of this bit is also dependent on the expansion card.
14	DIG EXP	Digital Expansion - This bit is set to a "1" by the digital expansion card. The meaning of this interrupt status bit is dependent on the expansion card used. Clearing of this bit is also dependent on the expansion card.
13	DSP IO	DSP IO - This bit is set to a "1" when the DSP has entered data into the I/O register (Operational register in A32 space at offset 06_{16}). This bit is cleared by a read from the I/O register.
12	DSP ALM	DSP Alarm - This bit is set to a "1" when the DSP has indicated an alarm condition. For example, if the DSP is checking for limit violations, the DSP can be instructed to generate this alarm condition when a violation occurs. This bit is cleared after the interrupt acknowledge cycle.
11	END SCAN	End of Scan - This bit is set to a "1" when the V213 has completed its scan. This function is intended for use with single scans. This bit is cleared after the interrupt acknowledge cycle.
10	TTL TRIG	TTL Trigger - This bit is set to a "1" if the V213 receives a trigger from the selected TTL trigger line available from VXIbus. The selection of the TTL trigger line and the enable is from the TTL Trigger Register (Operational register in A32 space at offset 06_{16} page 48). This bit is cleared after the interrupt acknowledge cycle.

9	EXT TRIG	External Trigger - This bit is set to a “1” if the V213 receives an external trigger. This bit is cleared after the interrupt acknowledge cycle.
8	ERR	Error - This bit is set to a “1” if the V213 scan rate was too fast. For example, if the scan list included 20 channels, the V213's maximum scan rate is 2500 Hz. If a 5000 Hz scan rate was selected, the V213 will generate an interrupt with this bit set. This bit will only be cleared when the V213 in taken out of “run” mode.
07-00	Logical Address	These bits contain the Logical Address of the V213 during an interrupt acknowledge cycle. These bits are read as all ones during a non-interrupt acknowledge cycle.

During an interrupt acknowledge cycle the V213 enables this register onto the D15-D00 data lines. As shown above, the data returned will indicate the current interrupt status as well as the V213's current logical address. Please note that since any enabled source can generate an interrupt, more than one status bit can be set during the acknowledge cycle.

Enabling interrupts, disabling the Interrupt Mask, and selecting an Interrupt Request Line is controlled by the Interrupt Control Register (next page).

Interrupt Control Register

$1C_{16}$	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
(R)	MSK I/O* (1)	MSK DIG* (1)	MSK DSP* (1)	MSK ALM* (1)	MSK SCN* (1)	MSK TTL* (1)	MSK EXT* (1)	MSK ERR* (1)	EN* (1)	(1)	IRQ2 (1)	IRQ1 (1)	IRL0 (1)	(1)	(1)	(1)

( ) Power Up Value

Using this register to control interrupts is defined as follows:

Bits(s)	Mnemonic	Meaning
15	MSK I/O*	A “1” in this bit masks (prevents) the I/O expansion card from generating an interrupt request. This bit must be written with a "0" to allow the I/O expansion card to generate an interrupt.
14	MSK DIG*	A “1” in this bit masks (prevents) the digital expansion card from generating an interrupt request. This bit must be written with a "0" to allow the digital expansion card to generate an interrupt.
13	MSK DSP*	A “1” in this bit masks (prevents) the DSP IO full flag from generating an interrupt request. This bit must be written with a "0" to allow the DSP IO full flag to generate an interrupt.
12	MSK ALM*	A “1” in this bit masks (prevents) the DSP alarm flag from generating an interrupt request. This bit must be written with a "0" to allow the DSP alarm flag to generate an interrupt.
11	MSK SCN*	A "1" in this bit masks (prevents) the completion of a scan from generating an interrupt request. This bit must be written with a "0" to allow the end of scan to generate an interrupt.
10	MSK TTL*	A "1" in this bit masks (prevents) the selected TTL trigger line from generating an interrupt request. This bit must be written with a "0" to allow the TTL trigger to generate an interrupt.
9	MSK EXT*	A "1" in this bit masks (prevents) the external trigger input from generating an interrupt request. This bit must be written with a "0" to allow the external trigger to generate an interrupt.
8	MSK ERR*	A "1" in this bit masks (prevents) the error flag from generating an interrupt request. This bit must be written with a "0" to allow the error flag to generate an interrupt.

7	EN*	A one in this bit is used to disable interrupt generation. This bit prevents <u>any</u> 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 V213 is not capable of interrupt handling, this bit should always be written with a "1".
5-3	IRL2*-IRL0*	This 3-bit field selects the VXIbus interrupt line associated with the interrupt according to the following table:

IRL2* (D05)	IRL1* (D04)	IRL0* (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

2-0	Not Used	These bits are reserved for selecting an interrupt handler line. The V213 does not have interrupt handler capabilities, these bits should always be written with "1s".

All bits in this register are set to "1" on the assertion of SYSRESET* or if the SOFT RESET bit in the Status/Control register is written with a "1". If any condition is to generate an interrupt, the appropriate interrupt masks must be set, IREN* bits must be cleared, and IRL2*-IRL0* must be a value other than 111_2. The mask bits and interrupt enable bits do not effect the current Cause/Status. For example, if an external trigger has occurred, changing the mask bit from a one to a zero will cause the pending interrupt to generate a VXI interrupt. Similarly, if an interrupt request is pending and the interrupt enable bit is changed from a zero to a one, the interrupt request will not be cleared. An interrupt request pending on any particular line is cleared by the assertion of SYSRESET*, putting the module into Soft Reset, changing IRL2*-IRL0*, or if the interrupt is properly acknowledged.

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 V213 is a VXIbus defined Extended Device.
14-0	These bits indicate that this is an Extended Register Based Device.

Suffix High Register

20_{16}	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
(R)	Suffix High Register															
	(?)															

( ) Power Up Value

Suffix Low Register

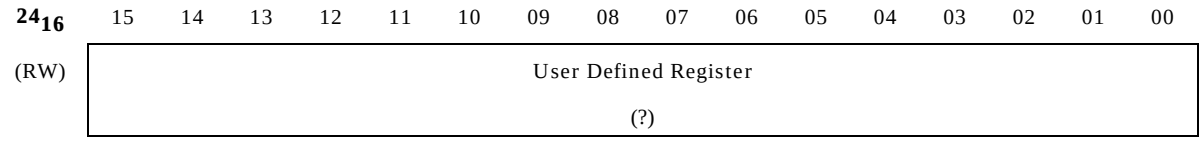
22_{16}	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
(R)	Suffix Low Register															
	(?)															

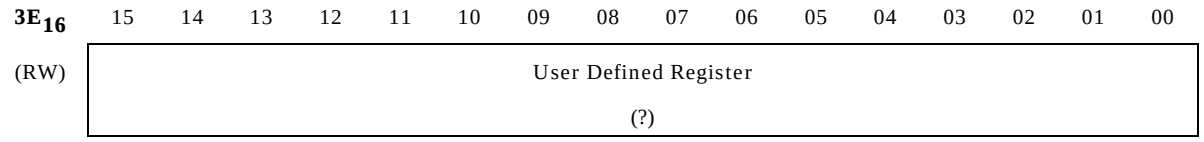
( ) Power Up Value

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 V213-AAA1. The module's suffix is "AAA1". Converting this to ASCII yields 41414131_{16}.

User Defined Registers





( ) 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.

Initial Setup

In general, V213's Operational Registers are organized in increasing order from setup to data collection. The Control Register (Operational Register in A32 space at offset 00_{16}) should be written to select the source of the scan rate. The four selections for the source of the scan rate include:

- Internal continuous scan rate. The Scan Rate Register (Operational Register in A32 space at offset 02_{16}) is used to set the internal scan rate.
- TTL Trigger Line available from the backplane. The trigger line can be selected by the TTL Trigger Register (Operational Register in A32 space at offset 06_{16} page 44). Other TTL trigger functions are also available from this register.
- External front panel source. In this case the scan rate is available from the SMB connector of the front panel.
- Internal single scan. In this case the scan rate is controlled by software. A scan is initiated as soon as the V213 is put into 'run' mode. This is accomplished by a read from the Start Scan Register (Operational Register in A32 space at offset 04_{16} page 47). Once the V213 has reached the end of its scanlist, it will automatically stop scanning. The status of 'run' can be monitored from bit 12 of the Control Register (Operational Register in A32 space at offset 00_{16} page 44). An interrupt may also be generated at the end of scan.

The scan rate may also be sent out a TTL trigger line to synchronize multiple V213s with a single clock.

Table 1 - Scan RAM / Scan Table Example

Scan Table Location (A32 offset)		Value In Scan RAM	Channel Selected
2000_{16}	(Start of scan list)	$0000\ 0000\ 0000\ 0000_2$	Channel 1
2002_{16}		$0000\ 0000\ 0000\ 0001_2$	Channel 2
2004_{16}		$0000\ 0000\ 0000\ 0010_2$	Channel 3
2006_{16}		$0000\ 0000\ 0000\ 0011_2$	Channel 4
2008_{16}		$0000\ 0000\ 0000\ 0100_2$	Channel 5
$200A_{16}$		$0000\ 0000\ 0000\ 0101_2$	Channel 6
$200C_{16}$		$0000\ 0000\ 0000\ 0110_2$	Channel 7
$200E_{16}$		$0000\ 0000\ 0000\ 0111_2$	Channel 8
2010_{16}		$0000\ 0000\ 0000\ 1000_2$	Channel 9
2012_{16}		$0000\ 0000\ 0000\ 1001_2$	Channel 10
2014_{16}		$0000\ 0000\ 0000\ 1010_2$	Channel 11
2016_{16}		$0000\ 0000\ 0000\ 1011_2$	Channel 12
2018_{16}		$0000\ 0000\ 0000\ 1100_2$	Channel 13
$201A_{16}$		$0000\ 0000\ 0000\ 1101_2$	Channel 14
$201C_{16}$		$0000\ 0000\ 0000\ 1110_2$	Channel 15
$201E_{16}$	(End of scan list)	$1000\ 0000\ 0000\ 1111_2$	Channel 16

For this example:

The V213 may be set to scan at 50 kHz (20 uS convert time)

If there are 16 channels in the Scan Table and since the V213 is a scanning ADC, then the scan rate must be less than or equal to 50 kHz / 16 channels = 3125 Hz. The scan rate determines how often the V213 starts its scan list.

Let scan rate = 1 kHz, t_{SCAN} = 1 / 1 kHz = 1 mS.

Every 1 mS:

1.) Scan Table starts at 2000_{16}.
V213 reads 0000_{16}. Channel 1 is converted which takes 20 uS.

2.) Scan Table is now at 2002_{16}.
V213 reads 0001_{16}. Channel 2 is converted which takes 20 uS.

3.) Scan Table is now at 2004_{16}.
V213 reads 0002_{16}. Channel 3 is converted which takes 20 uS.

.
.
.

15.) Scan Table is now at $201C_{16}$.
V213 reads $000E_{16}$. Channel 15 is converted which takes 20 uS.

16.) Scan Table is now at $201E_{16}$.
V213 reads $800F_{16}$. Channel 16 is converted which takes 20 uS. End of list bit is set indicating an end of the scan list. V213 will wait for the next edge of the scan rate clock to restart the scan list.

Setup, Scan RAM

Table 1 (page 36) shows an example of how Scan RAM might be setup in a case using 16 channels of the V213. There is also a programming example related to this topic in a file called "V213.C" in Appendix D.

The Scan RAM is 2048 words long, and the Scan Table is placed in Scan RAM. The V213 scans through its Scan Table sequentially at the ADC conversion rate (50kHz, 20kHz, or 2kHz). At any time, the current data in the list indicates which channel is converted and is stored in the next available location in ping/pong.

The Scan Table consists of a channel number and an end of list bit. The channel is specified by the first 6 bits of Scan RAM for a total of 64 channels. Any channel may be listed multiple times to be converted multiple times. Bit 15 is the end of list bit. This bit must be set on the last location of the scan list.

NOTE: When using multibuffer memory, only an even number of channels is allowed.

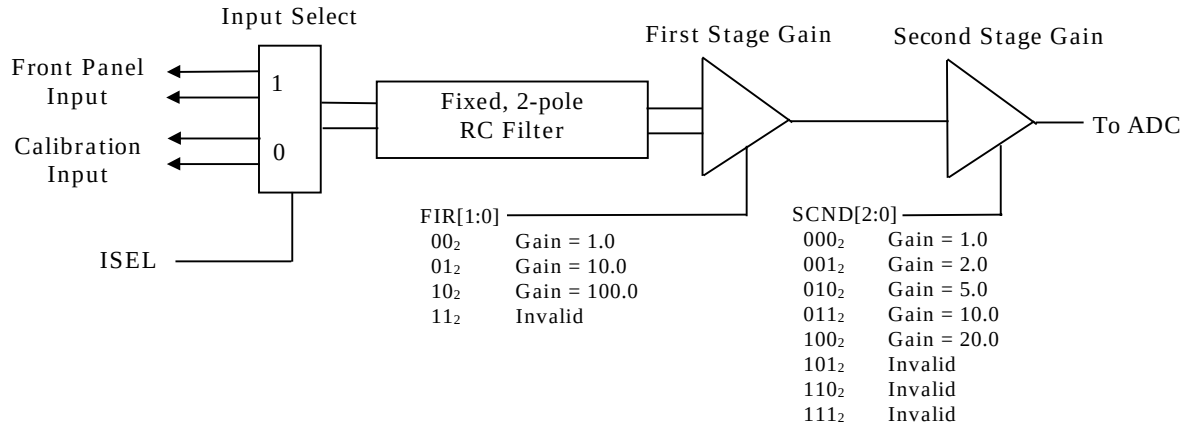


Figure 4 - Channel Setup

Setup Gain

The V213 has two separate gain stages. Both gain stages are applied after the filter and is controllable on a per channel basis which are controlled by Gain RAM (Operational Register in A32 space at offset 300_{16}). First stage gain can be changed by writing bits 5 and 4. First stage gain may be set to 1, 10, or 100. Second stage gain is set by writing bits 2 to 0. 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 connector. It is set by writing a one to the input select registers. 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 A32 space at offset 2_{16}).

Setup Gain For Isothermal Channels

The V213 can have up to two isothermal reference channels, one for each connector. Channel 1 may be switched to the isothermal reference available on the High Density connector, P3. It is controlled by setting bit 9 of the Control Register (Operational Register in A32 space at offset 00_{16} page 44). When the V213 is connected to KineticSystems isothermal panel or assembly, these reference channels will output voltage at 1 mV/K. When switched to the isothermal reference, the gain for channel 1 or 33 should be set to 20 and should be calibrated same as all other channels.

Calibration, Using the DSP

The DSP on the V213 is capable of calibrating all channels automatically using the internal calibration reference. The DSP will return offset and gain coefficients for each channel. These offset and gain error coefficients must be stored in software and be applied to all voltage readings from those channels. The following is a list of steps to follow in performing calibration:

1.) The V213 should be given a minimum of a 30 minute warm-up period. (for highest accuracy)
2.) Setup V213 channels to the chosen gain and filter selection. The chosen gain is referred
to as GAIN SETTING in the following equation. The gain settings available on the V213 is 1, 2, 5, 10, 20, 50, 100, 200, 500, 1000, or 2000.
3.) Request the DSP to perform calibration with current setup. Please note that if the scan list or gain settings are changed, calibration must be repeated.
   a.) Write 0120_{16} to DSP Register to request calibration.
   b.) The DSP will respond by writing a response back through the DSP Register. An interrupt will be asserted and bit 13 will be set in the Control Register (Operation Register in A32 space at offset 00_{16} page 44) when the DSP has responded. The DSP will respond with 0000_{16} to indicate the calibration opcode was accepted.
   c.) Write 0000_{16} to request all channels in the scan list be calibrated.
   d.) The DSP will again respond by writing a response back through the DSP Register. The DSP will respond with 0000_{16} to indicate calibration will proceed on all channels.
   e.) Once calibration is completed, the DSP will send a list of offset and gain coefficients for each channel through the DSP Register starting with the first channel in the scan list. The first number is the signed, 16-bit OFFSET of the channel in ADC counts. The second number is the signed, 16-bit GAIN_ERROR of the channel. The DSP will continue to list channels until it has reached the end of the scan list. If a channel is listed more than once in a scanlist, it will be listed multiple times through the DSP Register.
4.) For any reading from the ping/pong buffer or multibuffer card, the following equations can be used to correct the offset and gain errors of the channel. The reading is referred to as COUNTS in the following equations.

$$GAIN = GAINSETTING \bullet \left(1 + GAINERROR \bullet 10^{-6}\right)$$

$$CALIBRATEDVOLTAGE = \frac{\left(COUNTS - OFFSET\right) \bullet 319.8242188 \mu Volts}{GAIN}$$

If a channel is configured as an isothermal reference, the temperature can be calculated using the following equation:

$$TEMERATURE(CELCIUS) = (CALIBRATEDVOLTAGE - 273.2mV) \bullet 1000$$

Software Calibration

Each channel on the V213 can be calibrated using internal or external calibration sources 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. The following is a list of steps to follow in performing the V213 channel calibration:

Table 2 - Calibration Register Values, Internal Source

Calibration Voltage	Calibration Register Value (Internal Source)
+10.0 Volts	6091_{16}
+5.0 Volts	$60A1_{16}$
+2.0 Volts	$60C1_{16}$
+1.0 Volts	6092_{16}
+0.5 Volts	$60A2_{16}$
+0.2 Volts	$60C2_{16}$
+0.1 Volts	6094_{16}
+0.05 Volts	$60A4_{16}$
+0.02 Volts	$60C4_{16}$
+0.01 Volts	6098_{16}
+0.005Volts	$60A8_{16}$
+0.002 Volts	$60C8_{16}$
-0.002 Volts	6148_{16}
-0.005 Volts	6128_{16}
-0.01 Volts	6118_{16}
-0.02 Volts	6144_{16}
-0.05 Volts	6124_{16}
-0.1 Volts	6114_{16}
-0.2 Volts	6142_{16}
-0.5 Volts	6122_{16}
-1.0 Volts	6112_{16}
-2.0 Volts	6141_{16}
-5.0 Volts	6121_{16}
-10.0 Volts	6111_{16}

1.) The V213 should be given a minimum of a 30 minute warm-up period. (for highest accuracy)
2.) Setup V213 channels to the chosen gain and filter selection. The chosen gain is referred
to as GAIN SETTING in the following equation. The gain settings available on the V213 is 1, 2, 5, 10, 20, 50, 100, 200, 500, 1000, or 2000.
3.) Gain Calibration - for every channel, repeat steps 3a through 3f.
a.) Set channel so it is set to the calibration voltage. This is done by writing to the Input Select Registers (page 52).
b.) Set calibration voltage to that channels positive full scale voltage. The full scale

voltage is dependent on the gain setting for that channel. The expected positive calibration voltage will be referred to as $POSCAL_{EXPECTED}$ in the following equation. The calibration voltage is set by writing to the Calibration Register (Operational register in A32 space at offset $0A_{16}$ page 50). If the internal calibration reference is being used, a table of the register settings is available on page 41. otherwise, the value 4000_{16} must be written to the Calibration Register to set to the reference to the external source. Also, be sure to wait for the voltage to settle to 16 bit accuracy once the calibration voltage is set.

c.) Measure positive voltage (average 100 points, recommended). Use the following equation to convert the COUNTS read from ping/pong or multi-buffer into voltage RTO.

$$VOLTAGE(RTO) = \frac{1}{100}\sum_{1}^{100} COUNTS \bullet 319.8242188 \mu Volts$$

The measured positive calibration voltage is referred to as $POSCAL_{MEASURED}$ in the following equation.

d.) Set calibration voltage to that channels negative full scale voltage. The voltage is dependent on the gain setting for that channel. The expected negative calibration voltage will be referred to as $NEGCAL_{EXPECTED}$ in the following equation. If the internal calibration reference is being used, the calibration voltage is set by writing to the Calibration Register (Operational register in A32 space at offset $0A_{16}$ page 50). A table of the register settings is available on page 41. Also, be sure to wait for the voltage to settle to 16 bit accuracy once the calibration voltage is set.

e.) Measure negative voltage (average 100 points). Use the equation in 3c. The measured negative full scale calibration voltage is referred to as $NEGCAL_{MEASURED}$ in the following equation.

f.) If the internal calibration reference is being used, read gain error for the calibrator voltage from the Correction Table (Operational Register in A32 space starting at offset 400_{16} page 57). The gain error correction term is referred to as GAIN COEF in the following equation. Record gain for channel according to the following equation:

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

If an external calibration reference is being used, the following equation should be used without the gain correction term:

$$GAIN = \frac{POSCAL_{MEASURED} - NEGCAL_{MEASURED}}{\left(POSCAL_{EXPECTED} - NEGCAL_{EXPECTED}\right)}$$

4.) Offset Calibration - for every channel

a.) Set calibration voltage to ground. Write 4000_{16} to the Calibration Register (Operation register in A32 space at offset $0A_{16}$ page 50). Wait for the filter to settle to 16-bit accuracy.

b.) Measure offset in counts (average 100 points). For each channel, read the offset coefficient from the Correction Table (Operational Register in A32 space starting at offset 400_{16} page 57). The offset correction is referred to as OFF COEF in the following equation. Record offset for channel according to the following equation:

$$OFFSET = (\frac{1}{100}\sum_{1}^{100} COUNTS_{OFFSET}) + \frac{(OFFCOEF \bullet 10^{-9}) \bullet GAIN}{319.8242188 \mu Volts}$$

5.) For all other voltage readings - The V213 represents voltage in an twos complement 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 Gain RAM (Operation Registers in A32 space starting at offset 300_{16}). Convert counts to uncalibrated voltage using the following equation:

$$UNCALIBRATEDVOLTAGE = \frac{COUNTS \bullet 319.8242188 \mu Volts}{GAINSETTING}$$

Convert counts to calibrated voltage using the following equation:

$$CALIBRATEDVOLTAGE = \frac{(COUNTS - OFFSET) \bullet 319.8242188 \mu Volts}{GAIN}$$

If a channel is configured as an isothermal reference, the temperature can be calculated using the following equation:

$$TEMPERATURE(CELCIUS) = (CALIBRATEDVOLTAGE - 273.2mV) \bullet 1000$$

V213 Operational Registers, A32 Space

Control Register

00_{16}	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
(Mixed)	ERR		I/O FULL	RUN	*MOT		ISO1	EXT			SRC1	SRC0	CLK3	CLK2	CLK1	CLK0
	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(1)

( ) Power Up Value

Bits(s)	Mnemonic	Meaning
15	ERR	Error - When this READ only flag is set an error occurred during the scan. Either a first stage gain other than 1 was attempted at 50kHz, or the scan rate was too fast for the V213 to complete its scan list. In the second case there were too many channels in the scan list for the V213 to run at the selected scan rate. Should this happen, the V213 will not stop scanning. Rather, it will not start a new scan until it has completed its scan list. Also, if this bit is set, the Failed LED on the front panel will be on.
14	Not used.	This bit is not used and is reserved for future use. This bit will be read back as “0”.
13	I/O FULL	When this READ only flag is set, the DSP has placed valid data in the I/O Register (Operational Register in A32 space at offset 12_{16}). This bit will be cleared by a read from the DSP Register.
12	RUN	When this READ only flag is set, the V213 is ready to or is acquiring data. If this bit is zero after a single scan is initiated, the V213 has completed its scan list and valid data is available in ping/pong.
11	*MOT	This bit selects the manner in which the 16-bit data segments are arranged in a 32-bit transfer. This bit effects only those registers which support D32 transfers (i.e. ping/pong, page 58). When set, the 32-bit word is packed using the Intel format instead of the Motorola format.
10	Not used.	This bit is not used and is reserved for future use and will be read back as “0”.
9	ISO1	When set, channel 1 of the V213 is switched from the front panel to the isothermal reference. This reference is used to measure the temperature of the isothermal panel (V792) or a isothermal termination assembly (V750-ZC11) .
8	EXT	When set, the external trigger available from the front panel SMB is enabled. The trigger is a logical “or” of an enabled TTL trigger line, an external trigger, and a software trigger.

7,6	Not used.	These bits are not used and is reserved for future use and will be read back as "0s".
5,4	SRC[1:0]	These bits selects the source of the scan rate and are defined as follows:

00_2 Internal continuous scan. The rate of this clock is controlled by the Scan Rate Register (Operational Register in A32 space at offset 04_{16}).

01_2 TTL trigger line available from VXIbus. The selection of which of the eight trigger lines is controlled by the TTL Trigger Register (Operational Register in A32 space at offset 06_{16}).

10_2 External scan available from front panel SMB.

11_2 Internal single scan. Scan is initiated by a Read from the Start Scan Register (Operation Register in A32 space at offset 04_{16}). The RUN bit in this register will go to zero when the single scan is complete.

3-0	CLK[3:0]	These bits select the speed at which the ADC scans through the scanlist. At 50 kHz the first stage gain is limited to 1 which limits gain to 1 through 20. If the first stage gain is set to a value other than 1, the error LED on the front panel will light to indicate a problem. At 20 kHz all gains are available. At 2 kHz an additional filter is switched after the second stage gain to reduce noise.

0000_2 50 kHz Gain 1 - 20 available
0001_2 20 kHz Gain 1 - 2000 available
0010_2 2 kHz Gain 1 - 2000 available, Low Noise

Note: This register supports D16 transfers only. Writes are not allowed while the V213 is in "run" mode. A bus error will be generated if a write is attempted while in "run" mode.

Scan Rate Register

02_{16}	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
(RW)	Scan Rate Register (0000_{16})															

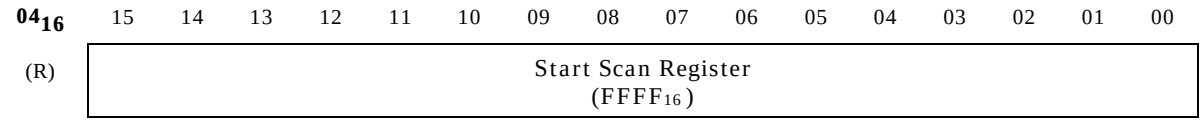
( ) Power Up Value

When the scan rate is selected as an internal continuous rate, this register controls the rate of the clock. A 50 kHz clock is divided by a 16-bit counter controlled by this register. For a given scan rate, the value written to this register is determined by the following equation:

$$SCAN\_RATE\_REGISTER = \frac{50kHz}{SCANRATE(kHz)} - 1$$

Note: This register supports D16 transfers only. Writes are not allowed while the V213 is in “run” mode. A bus error will be generated if a write is attempted while in “run” mode.

Start Scan Register



( ) Power Up Value

This READ only register is used to set the V213 to "run" mode. During this time, WRITE access to the Control, Scan Rate, and TTL Trigger Registers as well as Gain and Scan RAM are not allowed and will generate bus errors.

In cases where a continuous scan rate is to be used, a read from this register will toggle the V213 in and out of "run" mode . While in run mode, the V213 will convert each channel and store the resulting data in ping/pong at the selected scan rate.

In the cases where internal single scan is used, the scan is initiated by a READ from this register. Once the scan is complete, the V213 will automatically come out of "run" mode. The current status of "run" can be monitored from the Control Register (Operational Register in A32 space at offset 00_{16}).

Note: This register supports D16 transfers only.

TTL Trigger Register

06_{16}	15	14 13 12	11	10 09 08	07	06 05 04	03	02 01 00
(RW)	TOUT ENA (0)	TRIG OUT[2:0] (000_2)	TIN ENA (0)	TRIG IN[2:0] (000_2)	SOUT ENA (0)	SCAN OUT[2:0] (000_2)	SIN ENA (0)	SCAN IN[2:0] (000_2)

( ) Power Up Value

Bits(s)	Mnemonic	Meaning
15	TOUT ENA	When set, the V213 will output a trigger out on one of the eight TTL trigger lines available on VXIbus. The selection of the trigger line is made with bits 14 through 12 of this register. The trigger out may be used to trigger the slot 0 or another module in the chassis.
14-12	TRIG OUT[2:0]	Selection of which TTL trigger line is selected for trigger out is as follows: 000_2 TTL trigger line 0 001_2 TTL trigger line 1 010_2 TTL trigger line 2 011_2 TTL trigger line 3 100_2 TTL trigger line 4 101_2 TTL trigger line 5 110_2 TTL trigger line 6 111_2 TTL trigger line 7
11	TIN ENA	When set, the V213 will receive a trigger from one of the eight TTL trigger lines available on VXIbus. The selection of the trigger line is made with bits 10 through 8 of this register. The trigger in may be used by the digital expansion card. Its use is dependent on the type of expansion card installed.
10-8	TRIG IN[2:0]	Selection of which TTL trigger line is selected for trigger in is as follows: 000_2 TTL trigger line 0 001_2 TTL trigger line 1 010_2 TTL trigger line 2 011_2 TTL trigger line 3 100_2 TTL trigger line 4 101_2 TTL trigger line 5 110_2 TTL trigger line 6 111_2 TTL trigger line 7

7	SOUT ENA	When set, the V213 will output the scan rate out on one of the eight TTL trigger lines available on VXIbus. The selection of the trigger line is made with bits 6 through 4 of this register.
6-4	SCAN OUT[2:0]	Selection of which TTL trigger line is selected for scan out is as follows: 000_2 TTL trigger line 0 001_2 TTL trigger line 1 010_2 TTL trigger line 2 011_2 TTL trigger line 3 100_2 TTL trigger line 4 101_2 TTL trigger line 5 110_2 TTL trigger line 6 111_2 TTL trigger line 7
3	SIN ENA	When set, the V213 will receive its scan rate from one of the eight TTL trigger lines available on VXIbus. The selection of the trigger line is made with bits 2 through 0 of this register.
2-0	SCAN IN[2:0]	Selection of which TTL trigger line is selected for scan in is as follows: 000_2 TTL trigger line 0 001_2 TTL trigger line 1 010_2 TTL trigger line 2 011_2 TTL trigger line 3 100_2 TTL trigger line 4 101_2 TTL trigger line 5 110_2 TTL trigger line 6 111_2 TTL trigger line 7

Note: This register supports D16 transfers only. Writes are not allowed while the V213 is in “run” mode. A bus error will be generated if a write is attempted while in “run” mode.

Software Trigger Register

08_{16}	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
(R)	Software Trigger Register ($FFFF_{16}$)															

( ) Power Up Value

This READ only register will initiate a software trigger. This trigger is only used by the digital expansion card. The use of this trigger is dependent on the digital expansion card used.

Note: This register supports D16 transfers only.

Calibration Register

$0A_{16}$	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
(RW)		SRC1	SRC0	GND				INT NEG	INT POS	FIRST 2	FIRST 1	FIRS T 0	SCND 3	SCND 2	SCND 1	SCND 0
	(0)	(1)	(1)	(1)	(0)	(0)	(0)	(1)	(0)	(0)	(0)	(1)	(0)	(0)	(0)	(1)

( ) Power Up Value

The DSP will perform calibration and return offset and gain coefficients for each channel. However, these registers are also available to VXI in cases where software calibration is desirable.

Offset calibration is done by selecting the internal reference with bit 12 (GND) set. In this case the calibration voltage is set to ground to measure the offset error of a channel. If the same internal reference is used with bit 12 (GND) not set, a plus/minus voltage can be generated to perform gain calibration. Bits 8 through 0 of this register are used to define the voltages used to perform a gain calibration. The gain calibration voltage is selected in three groups. The polarity of the calibration voltage is selected with bits 8 and 7. Bit 8 or 7 must be set, but not both. Bits 6 through 4 of this register will scale the reference voltage by 0.5, 0.2, or 1. Only one of these bits may be set at a time. Bits 3 through 0 of this register will scale the reference voltage by an additional 0.001, 0.01, or 1. Only one of these bits may also be set at a time. A table listing all of the available calibration voltages with the appropriate values to be written to this register are on page 41.

Bits(s)	Mnemonic	Meaning
15	Not used.	This bit is not used and is reserved for future use. This bit will be read back as “0”.
14,13	SRC[1:0]	These bits control the 2-pin LEMO on the front panel. This connector can be configured to either input and external calibration source or output its internal calibration source. These bits are defined as follows: 11_2 Internal calibration source selected. Calibration voltage determined by bits 8 through 0 of this register. 10_2 Internal calibration source selected. Calibration voltage determined by bits 8 through 0 of this register and is available out on front panel 2 pin LEMO connector. 01_2 External calibration source selected. Calibration voltage is available in through front panel 2 pin LEMO connector. 00_2 Invalid.
12	GND	This bit should only be set if the internal calibration source is selected. When set, the internal source is ground.
11-9	Not used.	These bits are not used and is reserved for future use and will be read back as “0s”.
8	INT NEG	When set, the internal reference is set to -10.0 Volts
7	INT POS	When set, the internal reference is set to +10.0 Volts
6	FIRST2	When set, the internal reference is multiplied by 0.2
5	FIRST1	When set, the internal reference is multiplied by 0.5
4	FIRST0	When set, the internal reference is multiplied by 1.0
3	SCND3	When set, the internal reference is scaled by 0.001
2	SCND2	When set, the internal reference is scaled by 0.01
1	SCND1	When set, the internal reference is scaled by 0.1
0	SCND0	When set, the internal reference is scaled by 1.0

Note: This register supports D16 transfers only.

Input Select High

$0C_{16}$	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
(RW)	ISEL 32 (0)	ISEL 31 (0)	ISEL 30 (0)	ISEL 29 (0)	ISEL 28 (0)	ISEL 27 (0)	ISEL 26 (0)	ISEL 25 (0)	ISEL 24 (0)	ISEL 23 (0)	ISEL 22 (0)	ISEL 21 (0)	ISEL 20 (0)	ISEL 19 (0)	ISEL 18 (0)	ISEL 17 (0)

( ) Power Up Value

Input Select Low

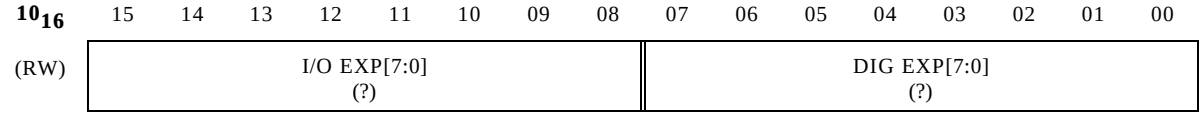
$0E_{16}$	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
(RW)	ISEL 16 (0)	ISEL 15 (0)	ISEL 14 (0)	ISEL 13 (0)	ISEL 12 (0)	ISEL 11 (0)	ISEL 10 (0)	ISEL 9 (0)	ISEL 8 (0)	ISEL 7 (0)	ISEL 6 (0)	ISEL 5 (0)	ISEL 4 (0)	ISEL 3 (0)	ISEL 2 (0)	ISEL 1 (0)

( ) Power Up Value

Mnemonic	Meaning
ISEL <Channel #>	When set, that channels input is switched from the calibration source to the front panel high density connector. The calibration source may be an internally or externally supplied source controlled by the Calibration Register (Operational Register in A32 space at offset $0A_{16}$).

Note: This register supports D16 transfers only.

Interface Option Register



( ) Power Up Value

This READ only register is used to determine if an I/O or digital expansion card is installed and if so, what type.

Bits(s)	Mnemonic	Meaning
15-8	I/O EXP[7:0]	These bits are used to determine the type of I/O expansion card used. These bits store the number in the last three characters of the card's suffix number. For example, the additional 32 channels with 10 Hz filters is a V213-x100. With this card installed, the value in bits 15 through 8 of this register would be 64_{16} or 100_{10}. Current options are as follows: 64_{16} Additional 32 Channels, 10 Hz filter 65_{16} Additional 32 Channels, 20 Hz filter 66_{16} Additional 32 Channels, 50 Hz filter 67_{16} Additional 32 Channels, 100 Hz filter 68_{16} Additional 32 Channels, 200 Hz filter 69_{16} Additional 32 Channels, 500 Hz filter $6A_{16}$ Additional 32 Channels, 1 kHz filter $8C_{16}$ 32 Channel RTD $A0_{16}$ 32 Channel DAC output $B4_{16}$ 16 Ch DAC, 32 Ch TTL Digital I/O $C8_{16}$ 64 Channel TTL Digital I/O FF_{16} No I/O expansion card installed.
7-0	DIG EXP[7:0]	These bits are used to determine the type of digital expansion card used. These bits store the number in the last three characters of the card's suffix number. Current options are as follows: FF_{16} No digital expansion card installed.

Note: This register supports D16 transfers only.

DSP Register

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

( ) Power Up Value

This is a general purpose I/O register for communication with the on-board fixed point DSP. A list of opcodes and responses is on page 61. Should the DSP have something to communicate, once it writes to the DSP Register, a VXI interrupt may be generated. Interrupts are controlled by the Interrupt Control Register (Configuration Register in A16 space at offset $1C_{16}$ page 31). A flag is also available in the Interrupt Status Register (Operation Register in A32 space at offset $1A_{16}$ page 29). This flag is reset by a read from the DSP Register.

Note: This register supports D16 transfers only.

Digital Expansion

100_{16}	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
(RW)	Digital Expansion (?)															

$1FE_{16}$	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
(RW)	Digital Expansion (?)															

( ) Power Up Value

This block of memory is reserved for the digital expansion card. The digital expansion card is available to increase the storage capacity or increase the functionality of the V213.

Note: This block of memory will support D16 and D16 BLK transfers on all digital expansion cards. Depending on the card option, D32 and D32 BLK transfers may also be supported. If no digital expansion card is installed and a read/write is attempted to this address space, a bus error will occur.

I/O Expansion

200_{16}	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
(RW)	I/O Expansion (?)															

$2FE_{16}$	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
(RW)	I/O Expansion (?)															

( ) Power Up Value

This block of memory is reserved for the I/O expansion card. The I/O expansion card is available to provide additional I/O options. Current options include additional 32 channels of analog input, analog output, and TTL I/O.

Note: This block of memory will support D16 and D16 BLK transfers on all I/O expansion cards. Depending on the card option, D32 and D32 BLK transfers may also be supported. If no I/O expansion card is installed and a read/write is attempted to this address space, a bus error will occur.

Gain RAM

300_{16}	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
(RW)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(?)	(?)	FIR1 (?)	FIR0 (?)	(?)	SGN2 (?)	SGN1 (?)	SGN0 (?)

$37E_{16}$	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
(RW)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(?)	(?)	FIR1 (?)	FIR0 (?)	(?)	SGN2 (?)	SGN1 (?)	SGN0 (?)

( ) Power Up Value

The gain can be set on a per channel basis. There is one register for each channel starting at offset 300_{16}. The offset may be computed for a given channel number by the following equation.

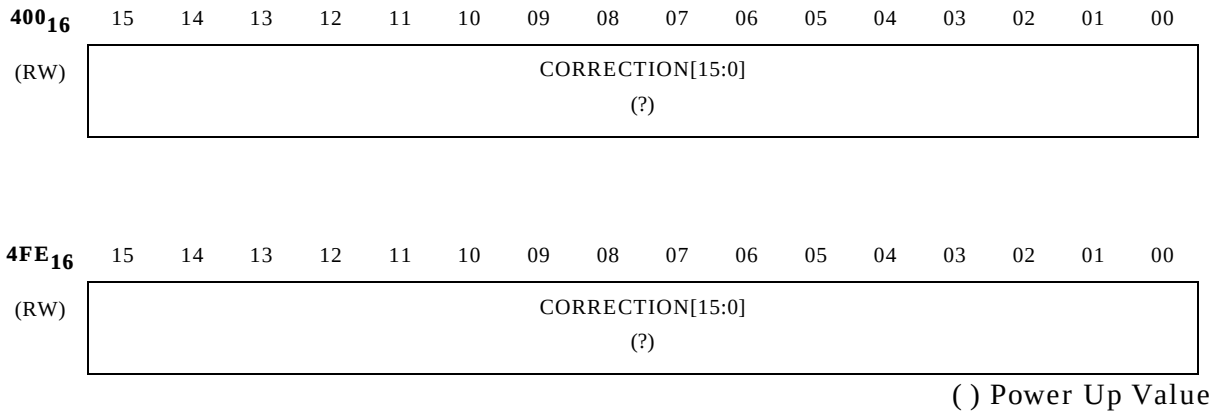
$$A32\_OFFSET = 2FE_{16} + (CHANNEL \bullet 2)$$

If channels 1 or 33 are configured as isothermal channels, their gain should always be set to 20. These channels may be calibrated same as all other channels to improve the accuracy of the reference measurement.

Bits(s)	Mnemonic	Meaning
15-6	Not used.	These bits are not used and is reserved for future use and should be written with "0s".
5-4	FIR[1:0]	The first stage gain selections are as follows: 00_2 Gain = 1.0 01_2 Gain = 10.0 10_2 Gain = 100.0 11_2 Invalid
3	Not used.	This bit is not used and is reserved for future use and should be written with "0s".
2-0	SGN[2:0]	The second stage gain selections are as follows: 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: This block of memory will support D16 and D16 BLK transfers. Writes are not allowed while the V213 is in "run" mode. A bus error will be generated if a write is attempted while in "run" mode.

Correction Table



( ) 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
400_{16}	The month of the last calibration date. (1 to 12)
402_{16}	The day of the last calibration date. (1 to 31)
404_{16}	The year of the last calibration date. (example 1995)
406_{16}	User definable.
.	.
$40E_{16}$	User definable.
410_{16}	Gain correction coefficient for +- 10 volts.
412_{16}	Gain correction coefficient for +- 5 volts.
414_{16}	Gain correction coefficient for +- 2 volts.
416_{16}	Gain correction coefficient for +- 1 volt.
418_{16}	Gain correction coefficient for +- 500 mV.
$41A_{16}$	Gain correction coefficient for +- 200 mV.
$41C_{16}$	Gain correction coefficient for +- 100 mV.
$41E_{16}$	Gain correction coefficient for +- 50 mV.
420_{16}	Gain correction coefficient for +- 20 mV.
422_{16}	Gain correction coefficient for +- 10 mV.
424_{16}	Gain correction coefficient for +- 5 mV.
426_{16}	Gain correction coefficient for +- 2 mV.
428_{16}	Reserved.
.	.
$42E_{16}$	Reserved.
430_{16}	Offset value (for I/O expansion options using DAC)
432_{16}	Gain value (for I/O expansion options using DAC)
434_{16}	Reserved.
.	.
$43E_{16}$	Reserved.

440_{16}	Offset correction coefficient for channel 1.
442_{16}	Offset correction coefficient for channel 2.
.	.
$4BC_{16}$	Offset correction coefficient for channel 63.
$4BE_{16}$	Offset correction coefficient for channel 64.
$4C0_{16}$	Reserved.
.	.
$4FE_{16}$	Reserved.

The gain correction coefficient is stored as a signed 16-bit integer. This coefficient need only be used if the internal calibration voltage is selected.

The offset correction coefficient is also stored as a signed 16-bit integer. This coefficient should be used for both internal and external calibration voltages.

Note: This block of memory will support D16 transfers only. Special care must be taken while writing information into these registers. Normally, write access is not allowed to prevent accidental erasure of these values. These gain coefficients are recorded during factory calibration. However, a CAL strap may be moved to allow write access to these registers. The location of this strap is shown in Figure 3, page 17. Once this 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.

Scan RAM

2000_{16}	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
(RW)	EOL										CH 5	CH 4	CH 3	CH 2	CH 1	CH 0
	(?)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(?)	(?)	(?)	(?)	(?)	(?)	(?)

$3FFE_{16}$	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
(RW)	EOL										CH 5	CH 4	CH 3	CH 2	CH 1	CH 0
	(?)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(?)	(?)	(?)	(?)	(?)	(?)	(?)

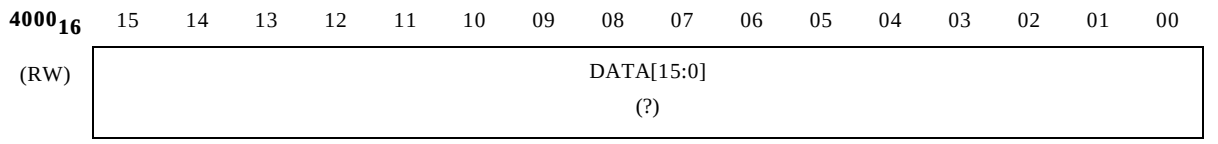
( ) Power Up Value

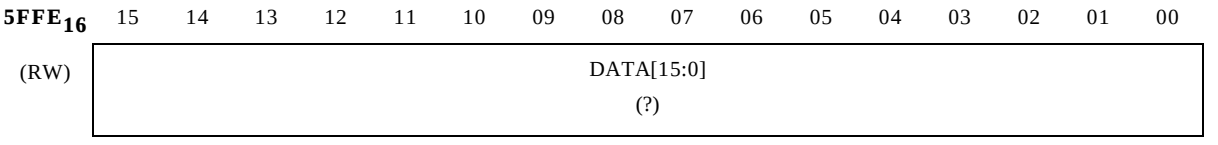
Bits(s)	Mnemonic	Meaning
15	EOL	End of list. The V213 will wait for next scan and resume scanning at the beginning of the list.
14-6	Not used.	These bits are not used and is reserved for future use and should be written as ”0s”.

5-0	CH[5:0]	Selects channel number on the V213 to be converted next. The value is simply the channel number minus 1.

Note: This block of memory will support D16 and D16 BLK transfers. Writes are not allowed while the V213 is in “run” mode. A bus error will be generated if a write is attempted while in “run” mode.

Ping/Pong RAM





( ) Power Up Value

While the V213 is in "run" mode, data is collected in the ping/pong buffer. This memory is organized as a dual ported memory between the ADC and VXIbus. The buffer is split into two halves (ping and pong). The ADC fills one half of the memory while VXI has access to the other half. The location of a channel's data is dependent on the scan list. Expanding on the example with 16 channels in a scan list, Table 3 shows the location of a channel's data in ping/pong as it relates to the scan list.

Table 3 - Scan RAM / Ping Pong Example

Scan Table Location	Value In Scan RAM	Location of Ping/ Pong Data, D16 transfers
2000_{16} (Start of scan list)	0000_{16} (Channel 1)	4000_{16}
2002_{16}	0001_{16} (Channel 2)	4002_{16}
2004_{16}	0002_{16} (Channel 3)	4004_{16}
2006_{16}	0003_{16} (Channel 4)	4006_{16}
2008_{16}	0004_{16} (Channel 5)	4008_{16}
$200A_{16}$	0005_{16} (Channel 6)	$400A_{16}$
$200C_{16}$	0006_{16} (Channel 7)	$400C_{16}$
$200E_{16}$	0007_{16} (Channel 8)	$400E_{16}$
2010_{16}	0008_{16} (Channel 9)	4010_{16}
2012_{16}	0009_{16} (Channel 10)	4012_{16}
2014_{16}	$000A_{16}$ (Channel 11)	4014_{16}
2016_{16}	$000B_{16}$ (Channel 12)	4016_{16}
2018_{16}	$000C_{16}$ (Channel 13)	4018_{16}
$201A_{16}$	$000D_{16}$ (Channel 14)	$401A_{16}$
$201C_{16}$	$000E_{16}$ (Channel 15)	$401C_{16}$
$201E_{16}$ (End of scan list)	$800F_{16}$ (Channel 16)	$401E_{16}$

Given the scanlist from the example in Table 3, data can be packed differently into a 32-bit word. Bit 11 in the Control Register (Operational Register in A32 space at offset 00_{16}, page 44) controls how data is packed in a 32-bit transfer.

Motorola Format, Bit 11 is "0" in Control Register

D31-D16	D15-D0
Channel 1 Data	Channel 2 Data

Intel Format, Bit 11 is "1" in Control Register

D31-D16	D15-D0
Channel 2 Data	Channel 1 Data

Note: This block of memory will support D16, D16 BLK, D32, and D32 BLK transfers.

Digital Memory

400000_{16}	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
(RW)	Digital Memory (4 Meg) (?)															

$7FFFFE_{16}$	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
(RW)	Digital Memory (4 Meg) (?)															

( ) Power Up Value

This second block of memory is reserved for the digital expansion card. The digital expansion card is available to increase the storage capacity or increase the functionality of the V213.

Note: This block of memory will support D16 and D16 BLK transfers on all digital expansion cards. Depending on the card option, D32 and D32 BLK transfers may also be supported. If no digital expansion card is installed and a read/write is attempted to this address space, a bus error will occur.

Accuracy

The accuracy of the V213 can be optimized by following a few simple rules:

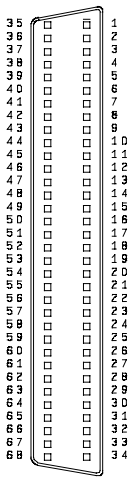
1. If the scanlist includes channels at multiple gains, order the channels in the scanlist with the highest gains first.
2. When possible, reduce the ADC rate from 50 kHz to 20 kHz or 2 kHz. For example, if 32 channels are in the scanlist and the scan rate is set to 10 Hz, the ADC could be slowed to 2 kHz for higher accuracy and lower noise.

The following table shows the initial accuracy of the V213 for all possible gains. Please note that the accuracy of gains listed in the table that are not in the V213 data sheet are guaranteed by design but are not 100 % tested.

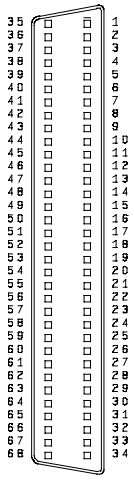
Gain Setting	Offset Error Typical	Offset Error Max	Gain Error Typical	Gain Error Max
2000 **(d)**	3.0 µVolts	5.0 µVolts	0.02 %	0.05 %
1000	3.0 µVolts	5.0 µVolts	0.01 %	0.025 %
500	3.0 µVolts	5.0 µVolts	0.007 %	0.015 %
200	5.0 µVolts	8.0 µVolts	0.007 %	0.015 %
100 ***(d)***	9.0 µVolts	13.0 µVolts	0.007 %	0.015 %
50	12.0 µVolts	25.0 µVolts	0.005 %	0.01 %
20	32.0 µVolts	60.0 µVolts	0.005 %	0.01 %
10 ***(d)***	60.0 µVolts	120.0 µVolts	0.005 %	0.01 %
5	120.0 µVolts	250.0 µVolts	0.005 %	0.01 %
2	320.0 µVolts	600.0 µVolts	0.005 %	0.01 %
1 **(d)**	640.0 µVolts	1200.0 µVolts	0.005 %	0.01 %

(d) = Data Sheet, 100% Tested

Figure 5 - 68 Pin High Density Connector P3 and P4



P3
(Main Card)



P4
(I/O Option)

Table 4 - 68 Pin High Density Connector: P3 Pinout

Pin #	P3 Description	Pin #	P3 Description	Pin #	P3 Description
1	Channel 1 +	13	Channel 13 +	25	Channel 25 +
35	Channel 1 -	47	Channel 13 -	59	Channel 25 -
2	Channel 2 +	14	Channel 14 +	26	Channel 26 +
36	Channel 2 -	48	Channel 14 -	60	Channel 26 -
3	Channel 3 +	15	Channel 15 +	27	Channel 27 +
37	Channel 3 -	49	Channel 15 -	61	Channel 27 -
4	Channel 4 +	16	Channel 16 +	28	Channel 28 +
38	Channel 4 -	50	Channel 16 -	62	Channel 28 -
5	Channel 5 +	17	Channel 17 +	29	Channel 29 +
39	Channel 5 -	51	Channel 17 -	63	Channel 29 -
6	Channel 6 +	18	Channel 18 +	30	Channel 30 +
40	Channel 6 -	52	Channel 18 -	64	Channel 30 -
7	Channel 7 +	19	Channel 19 +	31	Channel 31 +
41	Channel 7 -	53	Channel 19 -	65	Channel 31 -
8	Channel 8 +	20	Channel 20 +	32	Channel 32 +
42	Channel 8 -	54	Channel 20 -	66	Channel 32 -
9	Channel 9 +	21	Channel 21 +	33	Isothermal P3 -
43	Channel 9 -	55	Channel 21 -	67	Isothermal P3 +
10	Channel 10 +	22	Channel 22 +	34	Ground
44	Channel 10 -	56	Channel 22 -	68	Ground
11	Channel 11 +	23	Channel 23 +		
45	Channel 11 -	57	Channel 23 -		
12	Channel 12 +	24	Channel 24 +		
46	Channel 12 -	58	Channel 24 -		

DSP Opcodes

The following is a list of opcodes the fixed point DSP on the base card of the V213 will respond to. For each opcode or value written to the DSP Register (Operational Register in A32 space at offset 12_{16} page 50), the DSP will send a response back through the DSP register. If data is available in the DSP Register, an interrupt will be generated or bit 13 will be set in the Control Register (Operational Register in A32 space at offset 00_{16} page 44). In all cases, the response should be zero, but should a problem occur, the following are other responses the DSP might give.

Response	Meaning
0	Operation Complete.
-1	Invalid Opcode. The DSP was expecting an opcode, but did not receive a valid opcode.
-2	Invalid Data. The DSP did not receive a data value within range. The valid range of the data is dependent on the opcode.
-3	Operation failed. This response indicates the DSP was unable to successfully complete the requested operation. For example, if a selftest (opcode 01_{16}) is requested and fails, the DSP will respond with -3 instead of 0.

General Purpose Opcodes

Reset
Opcode = 00_{16}
Input type = None
Return type = None

This opcode performs a reset of the DSP. All variables for calibration and limit checks are reset to their default values. The DSP will deassert *SYSFAIL line on VXI backplane whether self test passes or not.

Self Test
Opcode = 01_{16}
Input type = None
Return type = None

This opcode will perform self tests. If selftest pass, the DSP will respond with 0. Should they fail, the DSP will respond with a -3. Before selftest are begun, limit checking is disabled as not to cause unexpected limit violations during the test.

Self Test with Report
Opcode = 02_{16}
Input type = None
Return type = ASCII

This opcode will perform self tests. While performing the test, the DSP will generate a report as to exactly what parts of the V213 passed and did not pass self tests. The DSP will also give information as to the current firmware version being used and what I/O and digital expansion cards it supports. The report is generated as an ASCII character stream placed in the DSP Register. The report is completed by the NULL (00_{16}) character. Before selftest are begun, limit checking is disabled as not to cause unexpected limit violations during the test.

Return Version
Opcode = 03_{16}
Input type = None
Return type = BCD

This opcode instructs the DSP to return the version of the DSP firmware. The DSP will return the version as binary coded decimal number (BCD).

Calibration Opcodes

The DSP will calculate offset and gain coefficients using the internal calibration reference. The DSP will not support the use of an external calibration source. In this case, calibration must be done in software. A step by step list of instructions for software calibration is discussed on page 39.

Set Settling Time
Opcode = 100_{16} <value>
Input type = Unsigned 16-bit integer
Return type = None

This opcode is used to set the amount of settling time during calibration. The <value> indicates the amount of time in mili-seconds. The default value of 2500 mS is used to guarantee a 10 Hz filter will settle to 16-bit accuracy using the internal calibration voltages. This number may be modified if filters with higher cutoffs are used to decrease the calibration time. The value for settling time cannot equal zero. If a value of zero is attempted, the DSP will respond with Invalid Data (-2).

Return Settling Time
Opcode = 101_{16}
Input type = None
Return type = Unsigned 16-bit integer

This opcode instructs the DSP to return the current value for settling time in mili-seconds. The default is 2500 mS.

Set Average Number
Opcode = 102_{16} <value>
Input type = Unsigned 16-bit integer
Return type = None

This opcode is used to set the number of samples averaged during calibration. The default value is 100 points. The value for the number of averages cannot equal zero. If a value of zero is attempted, the DSP will respond with Invalid Data (-2).

Return Average Number
Opcode = 103_{16}
Input type = None
Return type = Unsigned 16-bit integer

This opcode instructs the DSP to return the current value for the number of samples averaged during calibration. The default value is 100 points.

Calibrate
Opcode = 120_{16} <channel>
Input type = Unsigned 16-bit integer
Return type = Signed 16-bit integers (list)

This opcode is used to indicate what channels to calibrate. If the value for <channel> equals zero, all channels in the scan list will be calibrated. If the value is any other number, only that channel will be calibrated. Limit checking is disabled as not to cause unexpected limit violations during calibration.

The scan list and gain for all channels must be complete before calibration. Once calibration is begun, the DSP will temporarily put the V213 into "setup" mode so it may look up the scan list and read gain RAM. Once the V213 is placed back into "run" mode, the DSP will list offset and gain error for the selected channels. If all channels are being calibrated, the DSP will continue the list of offsets and gain coefficients in the order that they appear in the scanlist until the end of the scan list is reached.

For each channel, the DSP will first list the offset error for the channel indicated. The offset error is simply the average value in counts with the input to the channel grounded. This value is referred to as OFFSET in the following equation. The gain error for the channel is followed by the offset error. This value is referred to as GAIN_ERROR in the following equation. The variable GAIN_SETTING is the gain of the channel selected by Gain RAM (Operational Register in A32 space at offset 300_{16}). The calibrated gain for a channel can be computed as follows:

$$GAIN = GAINSETTING \bullet \left(1 + GAINERROR \bullet 10^{-6}\right)$$

Once a reading is taken from ping/pong, the following equation will remove offset and gain errors for that channel to provide the calibrated reading (RTI).

$$CALIBRATEDVOLTAGE = \frac{\left(COUNTS - OFFSET\right) \bullet 319.8242188 \mu Volts}{GAIN}$$

Calibrate DAC
Opcode = 121_{16}
Input type = None
Return type = None

This opcode is used to calibrate the DAC for the V213-x180 and -x160 I/O Expansion Options. If the expansion card is not installed, the DSP will respond with -3 to indicate it was unable to perform the calibration. Otherwise, the DSP will use the ADC on the main card to calibrate the DAC.

On the next power-up or reset, the V213 will revert back to the original offset and gain values stored in the Correction Table. To use the new offset and gain values, copy the values stored in the Offset and Gain Registers (see sections VI and VII) to the Correction Table.

Limit checking is disabled as not to cause unexpected limit violations during calibration.

Limit Checking Opcodes

The V213 supports one of two types of limit checking. Upper / Lower Bound is used in applications where a maximum and minimum setpoint is necessary for each channel. The upper and lower bound can be specified for each channel. The V213 will generate an interrupt or output on a specified TTL trigger line if a channel input is above the upper bound or less than the lower bound. Once the V213 is triggered, it will remain triggered until the input falls back in range plus a "dead band" (256 counts). This band is intended to remove multiple triggers caused by input noise.

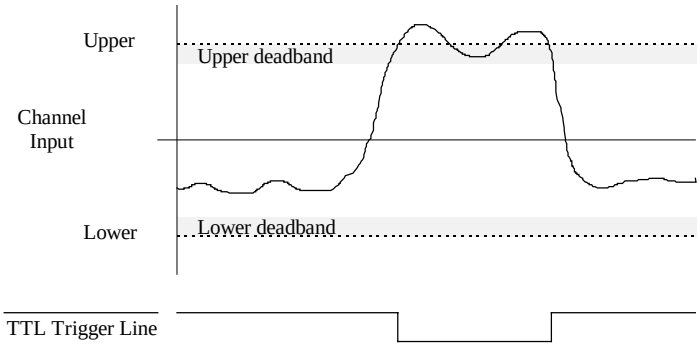


Figure 6 - Upper / Lower Bound Example

Conversely, the V213 can be configured to trigger on a threshold and slope. The V213 will generate an interrupt or output a pulse on a specified TTL trigger line if a channel crosses a voltage level (threshold) on a positive or negative slope. The threshold and slope can be specified on a per channel basis.

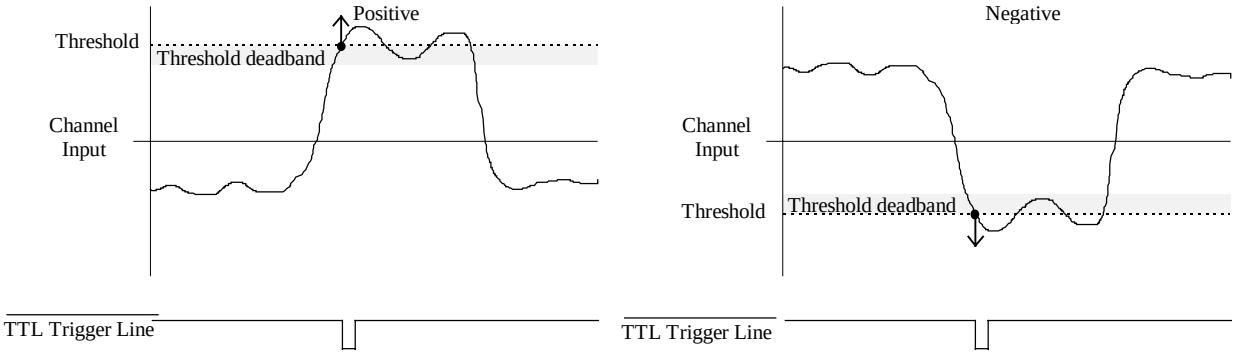


Figure 7 - Threshold / Slope Example

Please note: Limit checking can only be enabled if the ADC is converting at 20kHz or less.

Limit Type
Opcode = 200_{16} <value>
Input type = Unsigned 16-bit integer
Return type = None

This opcode selects between two basic types of limit checking. The DSP will support checking for upper/lower bounds or a threshold with slope. If <value> is set to zero (default), the DSP will check for limits using an upper and lower bound. In this case, the V213 is triggered if the current value of a channel is greater than an upper bound or less than a lower bound. The bounds are set on a per channel basis using opcodes 220_{16} and 222_{16}. If the <value> is set to a one, the DSP uses a threshold point and slope to check for a limit violation. In this case, if the current value of a channel crosses the threshold point, the V213 is triggered. The threshold and slope is specified on a per-channel basis with opcodes 224_{16} and 226_{16}. Because a parameter affecting limit checking is changed, limit checking is disabled as not to cause expected limit violations.

Return Limit Type
Opcode = 201_{16}
Input type = None
Return type = Unsigned 16-bit integer

This opcode instructs the DSP to return the current status of the limit type. The DSP will return a zero if the limit checking is set for upper/lower bound (default) or a one if it is set for a threshold and slope.

Limit Function (And / Or)
Opcode = 202_{16} <value>
Input type = Unsigned 16-bit integer
Return type = None

This opcode selects the trigger function when upper/lower bound limit checking is selected. If <value> is to zero, limit checking is done as an AND function. In this case, the V213 will trigger only if all channels meet the limit requirements. The limit checking may also be performed as an OR function (default). In this case, the V213 will trigger if any of the channels meet the limit requirements. Because a parameter affecting limit checking is changed, limit checking is disabled as not to cause expected limit violations.

Return Limit Function
Opcode = 203_{16}
Input type = None
Return type = Unsigned 16-bit integer

This opcode instructs the DSP to return the current status of the limit function. The DSP will return a zero if the limit function is as an AND of all channels or a one if it is set as an OR of any channel.

Set Upper Bound
Opcode 220_{16} <channel> <value>
Input type = Unsigned 16-bit integer, signed 16-bit integer
Return type = None

If limit checking is set to check for upper/lower bound, this opcode is used to set the upper bound. If the <channel> equals zero, the upper bound will be set on all channels. If the value is any other number, only that channel will be affected. The values for the upper and lower bounds are compared directly with the values from the ADC. When the V213 starts converting, if any value for a channel is greater than the upper bound or less than the lessor bound, a limit violation will occur. The following equation may be used to set the upper bound as a percentage of the full scale voltage:

$$VALUE = \left(\frac{PERCENTAGE}{100} \right) \bullet 32678 \qquad \text{where} \quad -100\% \le PERCENTAGE < 100\%$$

The default value for the upper bound is set to positive full scale of the ADC and will therefor not cause a trigger. Because a parameter affecting limit checking is changed, limit checking is disabled as not to cause unexpected limit violations.

Return Upper Bound
Opcode = 221_{16} <channel>
Input type = Unsigned 16-bit integer
Return type = Signed 16-bit integer

This opcode instructs the DSP to return the value for the given channel's upper bound. If the value for <channel> is between 1 and 64, the DSP will return the upper bound for that channel.

Set Lower Bound
Opcode 222_{16} <channel> <value>
Input type = Unsigned 16-bit integer, signed 16-bit integer
Return type = None

If limit checking is to check for upper/lower bound, this opcode is used to set the lower bound. If the <channel> equals zero, the lower bound will be set on all channels. If the value is any other number, only that channel will be affected. The values for the upper and lower bounds are compared directly with the values from the ADC. When the V213 starts converting, if any value for a channel is greater than the upper bound or less than the lessor bound, a limit violation will occur. The value for the lower bound may be calculated using the same equation:

$$VALUE = \left(\frac{PERCENTAGE}{100} \right) \bullet 32678 \qquad \text{where} \quad -100\% \le PERCENTAGE < 100\%$$

The default value for the lower bound is set to negative full scale of the ADC and will therefor not cause a trigger. Because a parameter affecting limit checking is changed, limit checking is disabled as not to cause unexpected limit violations.

Return Lower Bound
Opcode = 223_{16} <channel>
Input type = Unsigned 16-bit integer
Return type = Signed 16-bit integer

This opcode instructs the DSP to return the value for the given channel's lower bound. If the value for <channel> is between 1 and 64, the DSP will return the lower bound for the specified channel.

Set Threshold
Opcode 224_{16} <channel> <value>
Input type = Unsigned 16-bit integer, signed 16-bit integer
Return type = None

If limit checking is set to check for a threshold crossing, this opcode is used to control the threshold voltage. If the <channel> equals zero, the threshold will be set on all channels. If the value is any other number, only that channel will be affected. The lower 8-bits of <value> are zeroed to create a 256 sample “dead band” to prevent multiple triggers occurring due to noise. The value of the threshold is compared with the values from the ADC. The default value for the threshold is set to positive full scale of the ADC and will therefor not cause a trigger. Because a parameter affecting limit checking is changed, limit checking is disabled as not to cause unexpected limit violations.

Return Threshold
Opcode = 225_{16} <channel>
Input type = Unsigned 16-bit integer
Return type = Signed 16-bit integer

This opcode instructs the DSP to return the value for the given channel's threshold. If the value for <channel> is between 1 and 64, the DSP will return the threshold for the specified channel.

Set Polarity
Opcode 226_{16} <channel> <value>
Input type = Two unsigned 16-bit integers
Return type = None

If limit checking is set to check for a threshold crossing, this opcode is used to control the direction of the crossing. If the <channel> equals zero, the polarity will be set on all channels. If the value is any other number, only that channel will be affected. If the <value> is set to a one (positive = default), the V213 will be triggered if the current reading from the ADC has crossed the threshold on a positive edge. That means that the voltage for the given channel has changed from a voltage less than the threshold to a current value that is greater. In the case where the <value> is set to zero (negative), the V213 will be triggered on a negative crossing of the threshold voltage. Because a parameter affecting limit checking is changed, limit checking is disabled as not to cause unexpected limit violations.

Return Polarity
Opcode = 227_{16} <channel>
Input type = Unsigned 16-bit integer
Return type = Unsigned 16-bit integer

This opcode instructs the DSP to return the polarity for threshold limit check. If the value for <channel> is between 1 and 64, the DSP will return the polarity for the specified channel.

TTL Trigger On Limit Violation
Opcode = 240_{16} <value>
Input type = Signed 16-bit integer
Return type = None

This opcode is used to indicate which TTL trigger line will be asserted should a limit violation occur. If the value is -1, no TTL trigger line will be asserted (default). Values 0 through 7 indicate the particular trigger line to be asserted.

If the limit is checked using upper and lower bounds and a violation occurs, the TTL trigger line will be asserted for a long as the current value is out of bounds. However, if a violation occurs from a threshold crossing, the TTL trigger line will be asserted as a pulse.

Return TTL Trigger On Limit Violation
Opcode = 241_{16}
Input type = None
Return type = Signed 16-bit integer

This opcode instructs the DSP to return the current status for the TTL trigger.

Set Total Trigger Count
Opcode = 260_{16} <value>
Input type = Unsigned 16-bit integer
Return type = None

This opcode is used to indicate the number of triggers allowed. If the <value> is set to $FFFF_{16}$ (65535), then the V213 will allow an infinite number of triggers. If <value> is any other number, the post trigger count will decrement by one for each trigger until it reaches zero disabling limit checking. The default value for the Trigger Counter is set to one. The trigger count for each individual channel is reset to zero.

Return Total Trigger Count
Opcode = 261_{16}
Input type = None
Return type = Unsigned 16-bit integer

This opcode instructs the DSP to return the current number of triggers allowed.

Return Trigger Count
Opcode = 262_{16} <channel>
Input type = Unsigned 16-bit integer
Return type = Unsigned 16-bit integer

This opcode instructs the DSP to return the number of triggers caused by the specified channel. Should the count reach its maximum (65535), the count will no longer be incremented with each trigger.

Enable/Disable Limit Check
Opcode = 280_{16} <value>
Input type = Unsigned 16-bit integer
Return type = None

This opcode will enable or disable limit checking. If <value> is set to one and the ADC conversion rate is ≤20kHz, limit checking is enabled. As a default, limit checking is disabled (zero).

The scan list and gain for all channels must be complete before limit checking is enabled. Once enabled, the DSP will temporarily put the V213 into "setup" mode so it may look up the scan list and read Gain RAM.

While the ADC is converting (ADC LED on front panel is on) the DSP LED will be on if the DSP is checking for limit violations. If limit checking is disabled by opcode 280_{16} or the Trigger Counter decrements to zero., the DSP LED will turn off.

Return Limit Enable Status
Opcode = 281_{16}
Input type = None
Return type = Unsigned 16-bit integer

This opcode instructs the DSP to return the current status of limit checking. If the number returned is a one, limit checking is enabled. Otherwise, the number returned will be a zero indicating limit checking is disabled.

Section IV. Analog Input Option

Analog Input Control Register

200_{16}	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
(Mixed)																ISO33
	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(0)	(0)	(0)	(0)

( ) Power Up Value

Bits(s)	Mnemonic	Meaning
15-1	Not Used	These bits are reserved for future use.
0	ISO 33	When set, channel 33 of the V213 is switched from the front panel to the isothermal reference. This reference is used to measure the temperature of the isothermal panel (V792) or a isothermal termination assembly (V750-ZC11).

Note: This register supports D16 transfers only.

Analog Input Select High

202_{16}	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
(RW)	ISEL 64	ISEL 63	ISEL 62	ISEL 61	ISEL 60	ISEL 59	ISEL 58	ISEL 57	ISEL 56	ISEL 55	ISEL 54	ISEL 53	ISEL 52	ISEL 51	ISEL 50	ISEL 49
	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)

( ) Power Up

Analog Input Select Low

204_{16}	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
(RW)	ISEL 48	ISEL 47	ISEL 46	ISEL 45	ISEL 44	ISEL 43	ISEL 42	ISEL 41	ISEL 40	ISEL 39	ISEL 38	ISEL 37	ISEL 36	ISEL 35	ISEL 34	ISEL 33
	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)

( ) Power Up

The input select registers are used by the DSP to control the Analog Input daughter card. Although these registers are accessible from VXI-bus, the DSP functions will normally provide sufficient control of these registers.

Mnemonic	Meaning
ISEL <Channel #>	When set, that channels input is switched from the calibration source to the front panel high density connector. The calibration source may be an internally or externally supplied source controlled by the Calibration Register (Operational Register in A32 space at offset $0A_{16}$).

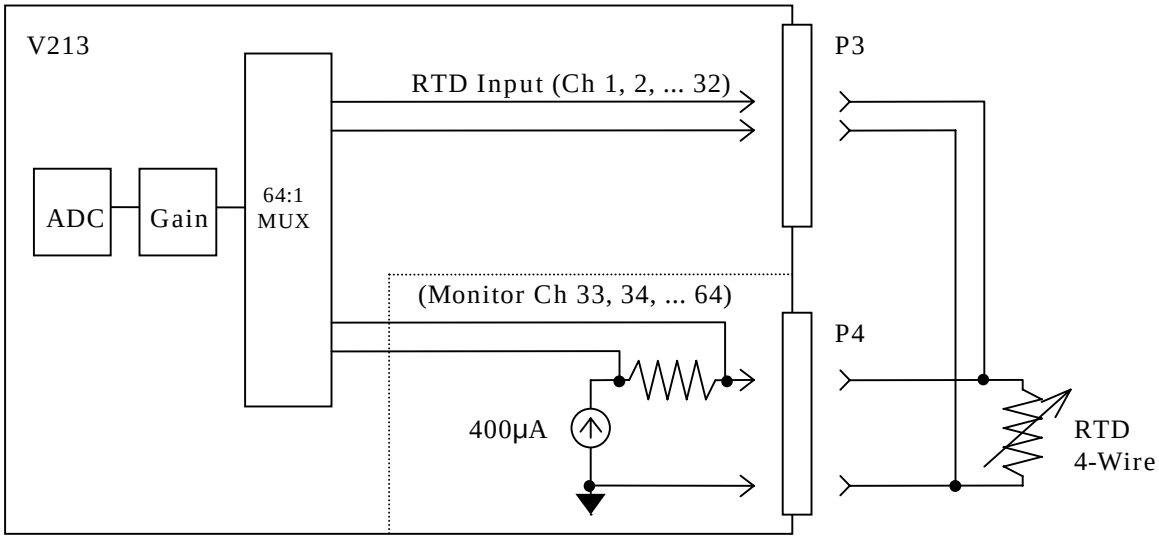
Note: This register supports D16 transfers only.

Table 5 - 68 Pin High Density Connector: P4 Pinout (Analog Input Option)

Pin #	P4 Description	Pin #	P4 Description	Pin #	P4 Description
1	Channel 33 +	13	Channel 45 +	25	Channel 57 +
35	Channel 33 -	47	Channel 45 -	59	Channel 57 -
2	Channel 34 +	14	Channel 46 +	26	Channel 58 +
36	Channel 34 -	48	Channel 46 -	60	Channel 58 -
3	Channel 35 +	15	Channel 47 +	27	Channel 59 +
37	Channel 35 -	49	Channel 47 -	61	Channel 59 -
4	Channel 36 +	16	Channel 48 +	28	Channel 60 +
38	Channel 36 -	50	Channel 48 -	62	Channel 60 -
5	Channel 37 +	17	Channel 49 +	29	Channel 61 +
39	Channel 37 -	51	Channel 49 -	63	Channel 61 -
6	Channel 38 +	18	Channel 50 +	30	Channel 62 +
40	Channel 38 -	52	Channel 50 -	64	Channel 62 -
7	Channel 39 +	19	Channel 51 +	31	Channel 63 +
41	Channel 39 -	53	Channel 51 -	65	Channel 63 -
8	Channel 40 +	20	Channel 52 +	32	Channel 64 +
42	Channel 40 -	54	Channel 52 -	66	Channel 64 -
9	Channel 41 +	21	Channel 53 +	33	Isothermal P4 -
43	Channel 41 -	55	Channel 53 -	67	Isothermal P4 +
10	Channel 42 +	22	Channel 54 +	34	Ground
44	Channel 42 -	56	Channel 54 -	68	Ground
11	Channel 43 +	23	Channel 55 +		
45	Channel 43 -	57	Channel 55 -		
12	Channel 44 +	24	Channel 56 +		
46	Channel 44 -	58	Channel 56 -		

Section V. RTD Option

The V213-x140 daughter card provides 32 current sources and a monitor for each source. These current sources can be used with the channels located on the main card to be used with 32 RTDs.



The RTD input channels are programmed in the scanlist as channels 1 through 32. The gain may be set according to the type of RTD and the temperature range of interest. The monitor channels are programmed in the scanlist as channels 33 through 64. The gain should always be set to 200 for the monitor channels.

After the completion of the power-up selftest, the scanlist will have all 64 channels in the list with all input channels set to a gain of 100 and the monitor channels set to a gain of 200. All that is required is to set the scan rate and calibrate before taking data.

Once calibration is complete, the current as measured by the monitor channel can be calculated as follows:

$$CURRENT = \frac{CALIBRATEDVOLTAGE}{100\Omega}$$

Refer to sections involving Software Calibration (page 41) or Calibration Using the DSP (page 39) for equations to calculate CALIBRATED VOLTAGE.

V213-x140 RTD Option

The following registers are used by the DSP to control the RTD daughter card. Although these registers are accessible from VXI-bus, the DSP functions will normally provide sufficient control of the daughter card. The RTD Control Register is only required during selftests and the monitor is automatically switched to the monitor resistor after calibration.

RTD Control Register

200_{16}	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
(Mixed)																TEST
	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(0)	(0)	(0)	(0)

( ) Power Up Value

Bits(s)	Mnemonic	Meaning
15-1	Not Used	These bits are reserved for future use.
0	TEST	When set, a resistor is switched into the current source for all channels. This bit must be set to zero during normal operation.

Note: This register supports D16 transfers only.

Monitor Source High

202_{16}	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
(RW)	MON 32 (0)	MON 31 (0)	MON 30 (0)	MON 29 (0)	MON 28 (0)	MON 27 (0)	MON 26 (0)	MON 25 (0)	MON 24 (0)	MON 23 (0)	MON 22 (0)	MON 21 (0)	MON 20 (0)	MON 19 (0)	MON 18 (0)	MON 17 (0)

( ) Power Up

Monitor Source Low

204_{16}	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
(RW)	MON 16 (0)	MON 15 (0)	MON 14 (0)	MON 13 (0)	MON 12 (0)	MON 11 (0)	MON 10 (0)	MON 9 (0)	MON 8 (0)	MON 7 (0)	MON 6 (0)	MON 5 (0)	MON 4 (0)	MON 3 (0)	MON 2 (0)	MON 1 (0)

( ) Power Up

Mnemonic	Meaning
MON <Channel #>	When set, that channels input is switched from the calibration source to the 100Ω monitor resistor. The calibration source may be an internally or externally supplied source controlled by the Calibration Register (Operational Register in A32 space at offset $0A_{16}$).

Note: This register supports D16 transfers only.

Table 6 - 68 Pin High Density Connector: P4 Pinout (RTD Option)

Pin #	P4 Description	Pin #	P4 Description	Pin #	P4 Description
1	Source 1 +	13	Source 13 +	25	Source 25 +
35	Source 1 -	47	Source 13 -	59	Source 25 -
2	Source 2 +	14	Source 14 +	26	Source 26 +
36	Source 2 -	48	Source 14 -	60	Source 26 -
3	Source 3 +	15	Source 15 +	27	Source 27 +
37	Source 3 -	49	Source 15 -	61	Source 27 -
4	Source 4 +	16	Source 16 +	28	Source 28 +
38	Source 4 -	50	Source 16 -	62	Source 28 -
5	Source 5 +	17	Source 17 +	29	Source 29 +
39	Source 5 -	51	Source 17 -	63	Source 29 -
6	Source 6 +	18	Source 18 +	30	Source 30 +
40	Source 6 -	52	Source 18 -	64	Source 30 -
7	Source 7 +	19	Source 19 +	31	Source 31 +
41	Source 7 -	53	Source 19 -	65	Source 31 -
8	Source 8 +	20	Source 20 +	32	Source 32 +
42	Source 8 -	54	Source 20 -	66	Source 32 -
9	Source 9 +	21	Source 21 +	33	Ground
43	Source 9 -	55	Source 21 -	67	Ground
10	Source 10 +	22	Source 22 +	34	Ground
44	Source 10 -	56	Source 22 -	68	Ground
11	Source 11 +	23	Source 23 +		
45	Source 11 -	57	Source 23 -		
12	Source 12 +	24	Source 24 +		
46	Source 12 -	58	Source 24 -		

Section VI. DAC Option

The V213-x160 daughter card provides 32 channels of DAC (Digital to Analog Conversion). A single, 16-bit DAC is used with a 32 channel sample hold. On power-up, the DSP will perform a selftest on all 32 channels as well as calibrate the DAC. The DSP uses offset and gain adjustments stored in the Correction Table (page 57) to remove offset and gain errors from the DAC. The DAC may be recalibrated at any time using opcode 121_{16} (see page 70 for details).

DAC Control Register

200_{16}	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
(RW)	CAL														MON	DAC
	(0)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)

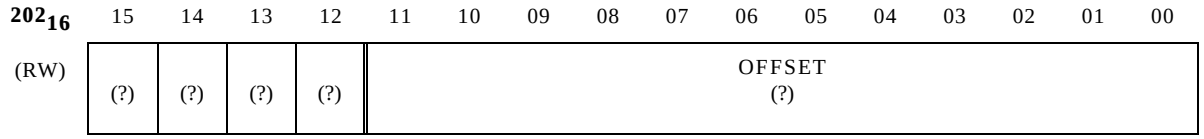
( ) Power Up

This register is used by the DSP to control the DAC daughter card during selftest and calibration. Although this register is accessible from VXI-bus, the DSP will normally provide sufficient control of the daughter card.

Value Bits(s)	Mnemonic	Meaning
15	CAL	When set, write access is granted to the Offset and Gain Registers.
14-2	Not Used	These bits are reserved for future use.
1	MON	This bit is used by the DSP to calibrate the monitor for channel 1. 0_2 Channel 1 connected to the calibration source 1_2 Channel 1 connected to the DAC
0	DAC	When set, all 32 DAC output channels are connected to the DAC. When reset, all 32 DAC outputs are connected to ground through a relay.

Note: This register supports D16 transfers only.

DAC Offset Register



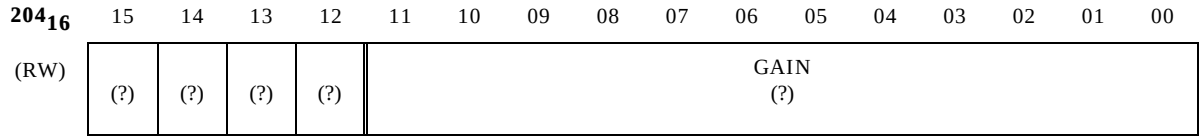
( ) Power Up

A 12-bit integer can be used to adjust the offset error of the DAC. The offset can be adjusted by at least ±30 mV. After power-up, the number stored in the Correction Table is copied to this register by the DSP. The DSP can perform a new calibration by using opcode 121_{16}.

The Offset Register cannot be written unless bit 15 of the DAC Control Register is set to one. This register is used by the DSP to calibrate the DAC on the DAC daughter card. Although this register is accessible from VXI-bus, the DSP will normally provide capabilities for this offset adjustment.

Note: This register supports D16 transfers only.

DAC Gain Register



( ) Power Up

A 12-bit integer can be used to adjust the gain error of the DAC. The gain can be adjusted by at least ±30 mV. After power-up, the number stored in the Correction Table is copied to this register by the DSP. The DSP can perform a new calibration by using opcode 121_{16}.

The Gain Register cannot be written unless bit 15 of the DAC Control Register is set to one. This register is used by the DSP to calibrate the DAC on the DAC daughter card. Although this register is accessible from VXI-bus, the DSP will normally provide capabilities for this gain adjustment.

Note: This register supports D16 transfers only.

DAC Register

240_{16}	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
(RW)	Channel 1 DAC (0000_{16})															

$27E_{16}$	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
(RW)	Channel 32 DAC (0000_{16})															

( ) Power Up

A 16-bit integer can be written into each channel register to change the voltage at the selected channel. The voltage range of the DAC is +9.999695 volts to -10.0 volts. The channel and voltage can be selected using the following equations:

$$A32\_OFFSET = 23E_{16} + (CHANNEL \bullet 2)$$

$$DAC\_REGISTER = \frac{VOLTAGE}{305.1757813\mu V}$$

The following table explains how data is packed in a 32-bit transfer.

Motorola Format, Bit 11 is “0” in Control Register
(Operational Register in A32 space at offset 00_{16})

D31-D16	D15-D0
Channel 1 Data	Channel 2 Data

Intel Format, Bit 11 is “1” in Control Register
(Operational Register in A32 space at offset 00_{16})

D31-D16	D15-D0
Channel 2 Data	Channel 1 Data

Note: This register supports D16, D16 BLK, D32 and D32 BLK transfers.

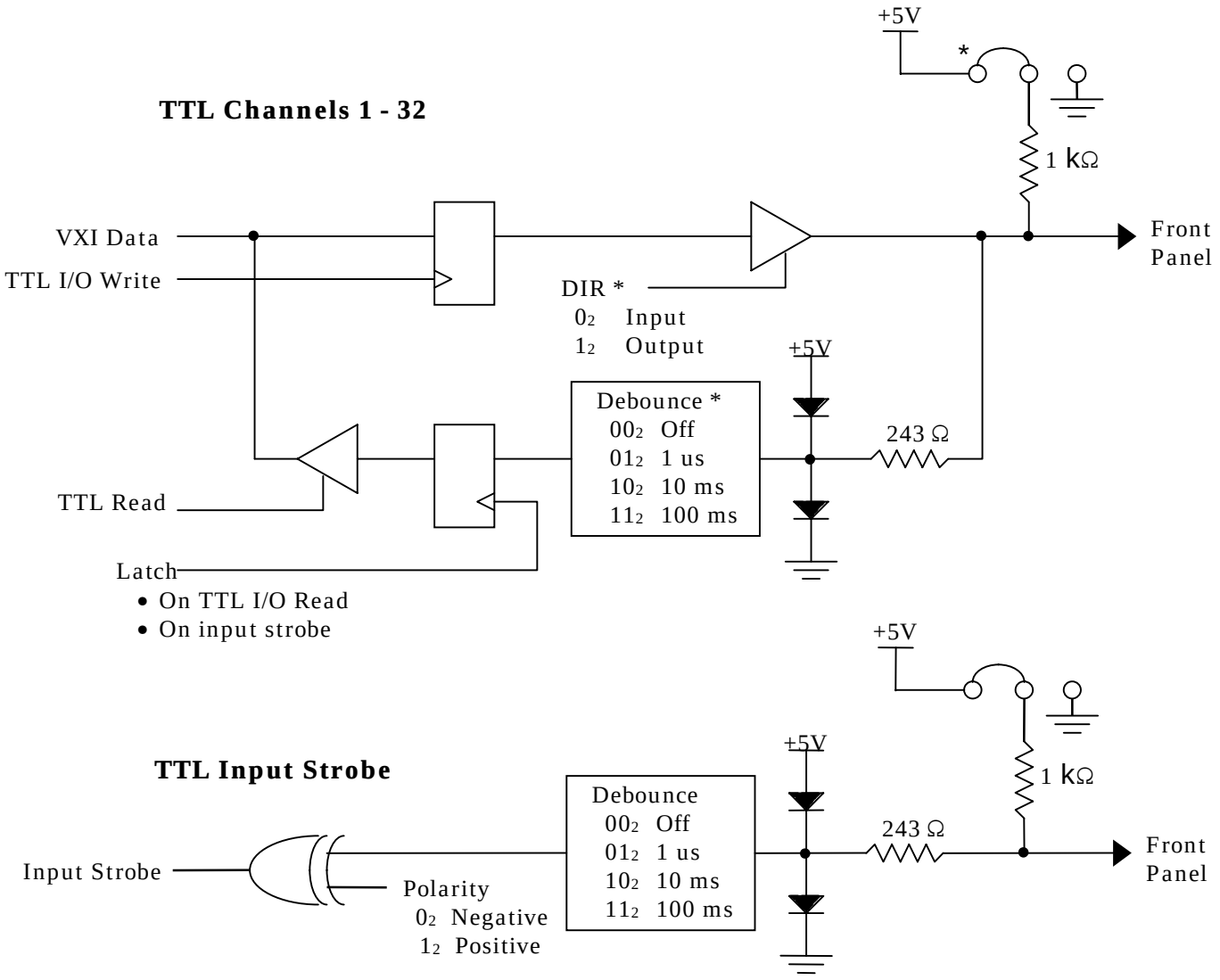
Table 6 - 68 Pin High Density Connector: P4 Pinout (DAC Option)

Pin #	P4 Description	Pin #	P4 Description	Pin #	P4 Description
1	Channel 1 +	13	Channel 13 +	25	Channel 25 +
35	Channel 1 -	47	Channel 13 -	59	Channel 25 -
2	Channel 2 +	14	Channel 14 +	26	Channel 26 +
36	Channel 2 -	48	Channel 14 -	60	Channel 26 -
3	Channel 3 +	15	Channel 15 +	27	Channel 27 +
37	Channel 3 -	49	Channel 15 -	61	Channel 27 -
4	Channel 4 +	16	Channel 16 +	28	Channel 28 +
38	Channel 4 -	50	Channel 16 -	62	Channel 28 -
5	Channel 5 +	17	Channel 17 +	29	Channel 29 +
39	Channel 5 -	51	Channel 17 -	63	Channel 29 -
6	Channel 6 +	18	Channel 18 +	30	Channel 30 +
40	Channel 6 -	52	Channel 18 -	64	Channel 30 -
7	Channel 7 +	19	Channel 19 +	31	Channel 31 +
41	Channel 7 -	53	Channel 19 -	65	Channel 31 -
8	Channel 8 +	20	Channel 20 +	32	Channel 32 +
42	Channel 8 -	54	Channel 20 -	66	Channel 32 -
9	Channel 9 +	21	Channel 21 +	33	Ground
43	Channel 9 -	55	Channel 21 -	67	Ground
10	Channel 10 +	22	Channel 22 +	34	Ground
44	Channel 10 -	56	Channel 22 -	68	Ground
11	Channel 11 +	23	Channel 23 +		
45	Channel 11 -	57	Channel 23 -		
12	Channel 12 +	24	Channel 24 +		
46	Channel 12 -	58	Channel 24 -		

Section VII. DAC / TTL Option

The V213-x180 daughter card provides 32 TTL I/O channels. All channels can be controlled in groups of 8 channels with the following features:

- Input or output under software control.
- Hardware strapped with a 1 kΩ pull-up.
- Debounce setting under software control.



* Controlled in groups of 8 channels

V213-x180 DAC / TTL Option

An input strobe is available to latch input data and generate an interrupt. Bit 15 of the Interrupt Status and Interrupt Control Registers are used to configure the interrupt coming from the I/O expansion card (see page 31). The I/O status bit will be cleared after the interrupt acknowledge cycle.

The V213-x180 daughter card also provides 16 channels of DAC (Digital to Analog Conversion). A single, 16-bit DAC is used with a 16 channel sample hold. On power-up, the DSP will perform a selftest on all 16 channels as well as calibrate the DAC. The DSP uses offset and gain adjustments stored in the Correction Table (page 57) to remove offset and gain errors from the DAC. The DAC may be recalibrated at any time using opcode 121_{16} (see page 70 for details).

DAC / TTL Control Register

200_{16}	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
(RW)	CAL													TTL	MON	DAC
	(0)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)

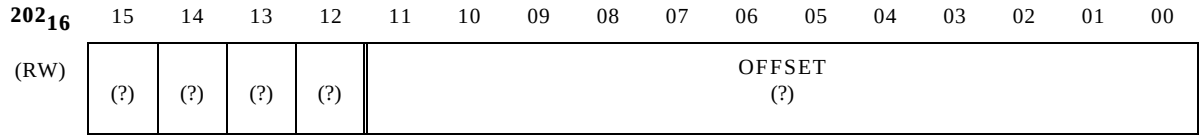
( ) Power Up

This register is used by the DSP to control the DAC / TTL daughter card during selftest and calibration. Although this register is accessible from VXI-bus, the DSP will normally provide sufficient control of the daughter card.

Value Bits(s)	Mnemonic	Meaning
15	CAL	When set, write access is granted to the Offset and Gain Registers.
14-3	Not Used	These bits are reserved for future use.
2	TTL	When set to zero, all TTL channels inputs are disconnected from the front panel and rerouted to the output register for diagnostic purposes.
1	MON	This bit is used by the DSP to calibrate the monitor for channel 1. 0_2 Channel 1 connected to the calibration source 1_2 Channel 1 connected to the DAC
0	DAC	When set, all 16 DAC outputs channels are connected to the DAC. When reset, all 16 DAC outputs are connected to ground through a relay.

Note: This register supports D16 transfers only.

DAC / TTL Offset Register



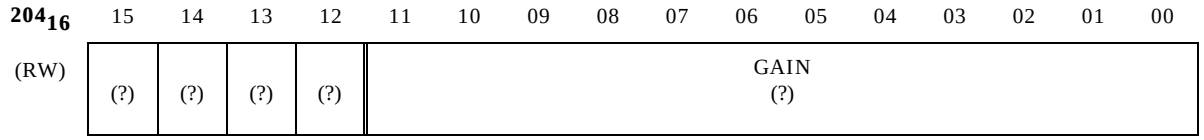
( ) Power Up

A 12-bit integer can be used to adjust the offset error of the DAC. The offset can be adjusted by at least ±30 mV. After power-up, the number stored in the Correction Table is copied to this register by the DSP. The DSP can perform a new calibration by using opcode 121_{16}.

The Offset Register cannot be written unless bit 15 of the DAC / TTL Control Register is set to one. This register is used by the DSP to calibrate the DAC on the DAC / TTL daughter card. Although this register is accessible from VXI-bus, the DSP will normally provide capabilities for this offset adjustment.

Note: This register supports D16 transfers only.

DAC / TTL Gain Register



( ) Power Up

A 12-bit integer can be used to adjust the gain error of the DAC. The gain can be adjusted by at least ±30 mV. After power-up, the number stored in the Correction Table is copied to this register by the DSP. The DSP can perform a new calibration by using opcode 121_{16}.

The Gain Register cannot be written unless bit 15 of the DAC / TTL Control Register is set to one. This register is used by the DSP to calibrate the DAC on the DAC / TTL daughter card. Although this register is accessible from VXI-bus, the DSP will normally provide capabilities for this gain adjustment.

Note: This register supports D16 transfers only.

DAC / TTL I/O Register

206_{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)	DIR 25-32 (0)	DIR 17-24 (0)	DIR 9-16 (0)	DIR 1-8 (0)

( ) Power Up

Value Bits(s)	Mnemonic	Meaning
15-4	Not Used	These bits are reserved for future use.
3	DIR 25-32	This bit sets the direction of channels 25 through 32. Should the input strobes be enabled on any channel, the direction for channel 32 will be configured as an input regardless of this bit. 0_2 Input 1_2 Output
2	DIR 17-24	This bit sets the direction of channels 17 through 24. 0_2 Input 1_2 Output
1	DIR 9-16	This bit sets the direction of channels 9 through 16. 0_2 Input 1_2 Output
0	DIR 1-8	This bit sets the direction of channels 1 through 8. 0_2 Input 1_2 Output

Note: This register supports D16 transfers only.

DAC / TTL Debounce Register

208_{16}	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
(RW)	Debounce Strobe (00_2)		(1)	(1)	(1)	(1)	(1)	(1)	Debounce 25-32 (00_2)		Debounce 17-24 (00_2)		Debounce 9-16 (00_2)		Debounce 1-8 (00_2)	

( ) Power Up

This register is used to set the debounce timing of the TTL input. Channels 1 through 32 are controlled in groups of 8 channels. The input strobe has independent control of the debounce setting.

Debounce

00_2	Bypass
01_2	1 us
10_2	10 ms
11_2	100 ms

Note: This register supports D16 transfers only.

DAC / TTL Input Strobe Register

$20A_{16}$	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
(RW)	IN POL												STB 25-32	STB 17-24	STB 9-16	STB 1-8
	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(0)	(0)	(0)	(0)

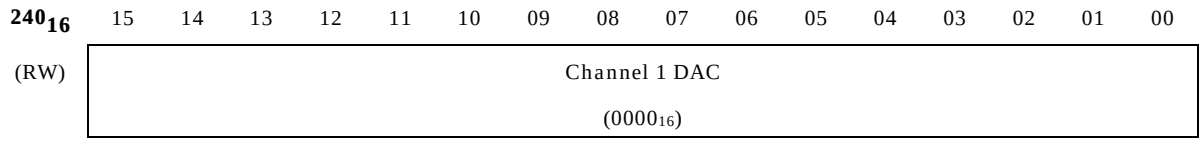
( ) Power Up

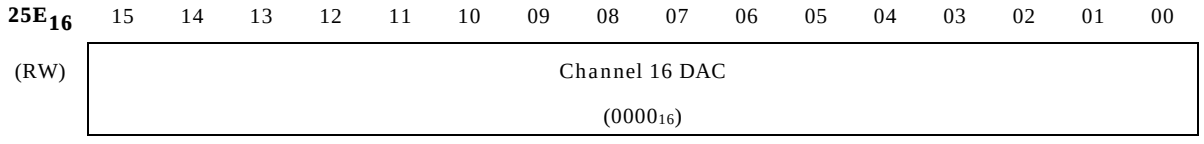
Value Bits(s)	Mnemonic	Meaning
15	IN POL	This bit can be used to set the input polarity of the strobe. 0_2 Negative Edge 1_2 Positive Edge
14-4	Not Used	These bits are reserved for future use.
3	STB 25-32	When set, channels 25 through 32 input will be latched by the input strobe.
2	STB 17-24	When set, channels 17 through 24 input will be latched by the input strobe.
1	STB 9-16	When set, channels 9 through 16 input will be latched by the input strobe.
0	STB 1-8	When set, channels 1 through 8 input will be latched by the input strobe.

An interrupt can be generated by the input strobe. Bit 15 of the Interrupt Status and Interrupt Control Registers are used to configure the interrupt coming from the I/O expansion card (see page 29). The I/O status bit will be cleared after the interrupt acknowledge cycle.

Note: This register supports D16 transfers only.

DAC Register





( ) Power Up

A 16-bit integer can be written into each channels register to change the voltage at the selected channel. The voltage range of the DAC is +9.999695 volts to -10.0 volts. The channel and voltage can be selected using the following equations:

$$A32\_OFFSET = 23E_{16} + (CHANNEL \bullet 2)$$

$$DAC\_REGISTER = \frac{VOLTAGE}{305.1757813\mu V}$$

The following table explains how data is packed in a 32-bit transfer.

Motorola Format, Bit 11 is “0” in Control Register
(Operational Register in A32 space at offset 00_{16})

D31-D16	D15-D0
Channel 1 Data	Channel 2 Data

Intel Format, Bit 11 is “1” in Control Register
(Operational Register in A32 space at offset 00_{16})

D31-D16	D15-D0
Channel 2 Data	Channel 1 Data

Note: This register supports D16, D16 BLK, D32 and D32 BLK transfers.

TTL I/O Register

280_{16}	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
(RW)	TTL 16	TTL 15	TTL 14	TTL 13	TTL 12	TTL 11	TTL 10	TTL 9	TTL 8	TTL 7	TTL 6	TTL 5	TTL 4	TTL 3	TTL 2	TTL 1

282_{16}	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
(RW)	TTL 32	TTL 31	TTL 30	TTL 29	TTL 28	TTL 27	TTL 26	TTL 25	TTL 24	TTL 23	TTL 22	TTL 21	TTL 20	TTL 19	TTL 18	TTL 17

On power-up the current value of the TTL input is stored in the TTL Register.

The following table explains how data is packed in a 32-bit transfer.

Motorola Format, Bit 11 is “0” in Control Register
(Operational Register in A32 space at offset 00_{16})

D31-D16	D15-D0
TTL Channels 16 - 1	TTL Channels 32 - 17

Intel Format, Bit 11 is “1” in Control Register
(Operational Register in A32 space at offset 00_{16})

D31-D16	D15-D0
TTL Channels 32 - 17	TTL Channels 16- 1

These read / write registers provide access to all 32 TTL channels. If the direction is set for an output, a write to the registers will effect output. Regardless of the direction of the TTL channel, a read will return the current value of the TTL channel.

Note: This register supports D16 and D32 transfers.

V213-x180 DAC / TTL Option

Table 7 - 68 Pin High Density Connector: P4 Pinout (DAC / TTL Option)

Pin #	P4 Description	Pin #	P4 Description	Pin #	P4 Description
1	Channel 1 +	13	Channel 13 +	25	TTL 17
35	Channel 1 -	47	Channel 13 -	59	TTL 18
2	Channel 2 +	14	Channel 14 +	26	TTL 19
36	Channel 2 -	48	Channel 14 -	60	TTL 20
3	Channel 3 +	15	Channel 15 +	27	TTL 21
37	Channel 3 -	49	Channel 15 -	61	TTL 22
4	Channel 4 +	16	Channel 16 +	28	TTL 23
38	Channel 4 -	50	Channel 16 -	62	TTL 24
5	Channel 5 +	17	TTL 1	29	TTL 25
39	Channel 5 -	51	TTL 2	63	TTL 26
6	Channel 6 +	18	TTL 3	30	TTL 27
40	Channel 6 -	52	TTL 4	64	TTL 28
7	Channel 7 +	19	TTL 5	31	TTL 29
41	Channel 7 -	53	TTL 6	65	TTL 30
8	Channel 8 +	20	TTL 7	32	TTL 31
42	Channel 8 -	54	TTL 8	66	TTL 32
9	Channel 9 +	21	TTL 9	33	Strobe
43	Channel 9 -	55	TTL 10	67	Ground
10	Channel 10 +	22	TTL 11	34	Ground
44	Channel 10 -	56	TTL 12	68	Ground
11	Channel 11 +	23	TTL 13		
45	Channel 11 -	57	TTL 14		
12	Channel 12 +	24	TTL 15		
46	Channel 12 -	58	TTL 16		

MULTIBUFFER END ADDRESS REGISTER

APPENDIX

APPENDIX A - Register Layout for A16 and A32 Space

APPENDIX B - Procedure for Module Calibration

APPENDIX C - Driving Balanced Analog Inputs From Unbalanced Sources by Robert T. Cleary, September, 1987

APPENDIX D - C/C++ Programming Examples Using National Instruments Drivers

APPENDIX A - Register Layout for A16 and A32 Space

V213 Configuration Registers, A16 Space

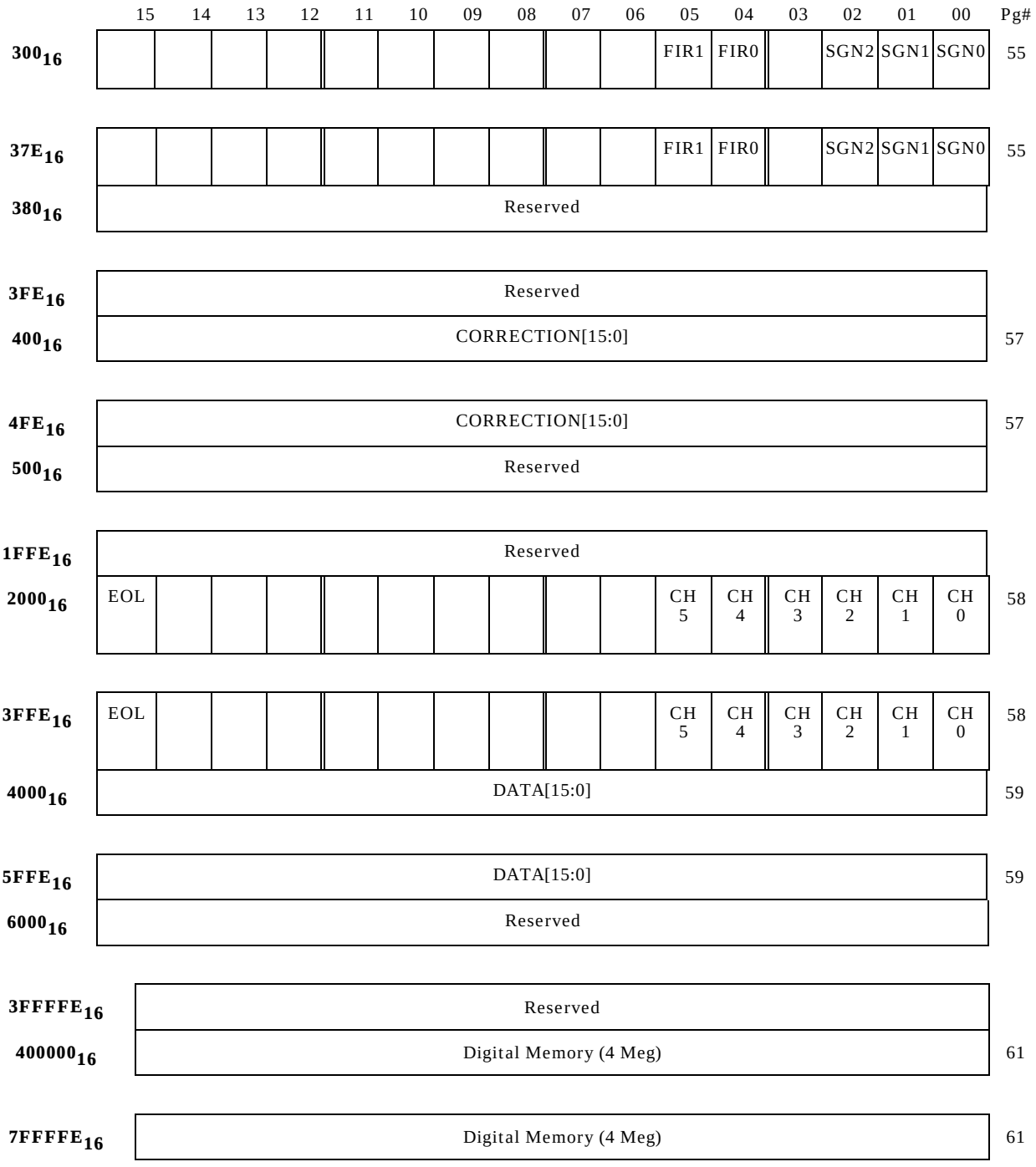
	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00	Pg#
00_{16}	(0)	(1)	(0)	(1)	(0)	(1)	(0)	(0)	(1)	(1)	(1)	(1)	(1)	(0)	(0)	(1)	24
(w)									LA7	LA6	LA5	LA4	LA3	LA2	LA1	LA0	24
02_{16}	(0)	(1)	(1)	(1)	(0)	(0)	(1)	(0)	(0)	(0)	(0)	(1)	(0)	(0)	(1)	(1)	24
04_{16}	A32 ENA	Modid											Ready	Pass	Sys Inb.	Soft Reset	25
06_{16}	OFF 15	OFF 14	OFF 13	OFF 12	OFF 11	OFF 10	OFF 9	OFF 8									26
08_{16}														IR*	IH*	IC*	26
$0A_{16}$	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	27
$0C_{16}$	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	27
$0E_{16}$	Firmware Version #				Firmware Revision #				Hardware Version #				Hardware Revision #				28
10_{16}	Reserved																

18_{16}	Reserved																
$1A_{16}$	IO EXP	DIG EXP	DSP IO	DSP ALM	END SCAN	TTL TRIG	EXT TRIG	ERR	Logical Address								29
$1C_{16}$	MSK I/O*	MSK DIG*	MSK DSP*	MSK ALM*	MSK SCN*	MSK TTL*	MSK EXT*	MSK ERR*	IREN *		IRQ2	IRQ1	IRL0				31
$1E_{16}$	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(0)	33
20_{16}	Suffix High Register																33
22_{16}	Suffix Low Register																33
24_{16}	User Defined Register																34

$3E_{16}$	User Defined Register	34

V213 Operational Registers, A32 Space

	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00	Pg#
00_{16}	ERR		I/O FULL	RUN	*MOT		ISO1	EXT			SRC1	SRC0	CLK3	CLK2	CLK1	CLK0	44
02_{16}	Scan Rate Register																45
04_{16}	Start Scan Register																47
06_{16}	TOUT ENA	TRIG OUT[2:0]			TIN ENA	TRIG IN[2:0]			SOUT ENA	SCAN OUT[2:0]			SIN ENA	SCAN IN[2:0]			48
08_{16}	Software Trigger Register																50
$0A_{16}$		SRC1	SRC0	GND				INT POS	INT NEG	FIRS T 2	FIRS T 1	FIRS T 0	SCND 3	SCND 2	SCND 1	SCND 0	50
$0C_{16}$	ISEL 16	ISEL 15	ISEL 14	ISEL 13	ISEL 12	ISEL 11	ISEL 10	ISEL 9	ISEL 8	ISEL 7	ISEL 6	ISEL 5	ISEL 4	ISEL 3	ISEL 2	ISEL 1	52
$0E_{16}$	ISEL 32	ISEL 31	ISEL 30	ISEL 29	ISEL 28	ISEL 27	ISEL 26	ISEL 25	ISEL 24	ISEL 23	ISEL 22	ISEL 21	ISEL 20	ISEL 19	ISEL 18	ISEL 17	52
10_{16}	I/O EXP[7:0]								DIG EXP[7:0]								53
12_{16}	DSP 15	DSP 14	DSP 13	DSP 12	DSP 11	DSP 10	DSP 9	DSP 8	DSP 7	DSP 6	DSP 5	DSP 4	DSP 3	DSP 2	DSP 1	DSP 0	54
14_{6}	Reserved																
FE_{16}	Reserved																
100_{16}	Digital Expansion																54
$1FE_{16}$	Digital Expansion																54
200_{16}	I/O Expansion																55
$2FE_{16}$	I/O Expansion																55


15 14 13 12 11 10 09 08 07 06 05 04 03 02 01 00 Pg#
300₁₆
FIR1 FIR0 SGN2 SGN1 SGN0
55
37E₁₆
FIR1 FIR0 SGN2 SGN1 SGN0
55
380₁₆
Reserved
3FE₁₆
Reserved
400₁₆
CORRECTION[15:0]
57
4FE₁₆
CORRECTION[15:0]
57
500₁₆
Reserved
1FFE₁₆
Reserved
2000₁₆
EOL CH 5 CH 4 CH 3 CH 2 CH 1 CH 0
58
3FFE₁₆
EOL CH 5 CH 4 CH 3 CH 2 CH 1 CH 0
58
4000₁₆
DATA[15:0]
59
5FFE₁₆
DATA[15:0]
59
6000₁₆
Reserved
3FFFFE₁₆
Reserved
400000₁₆
Digital Memory (4 Meg)
61
7FFFFE₁₆
Digital Memory (4 Meg)
61

APPENDIX B - Procedure for Module Calibration

In order to achieve the high degree of accuracy the V213, error coefficients for both offset and gain are stored in the Correction Table (non-volatile RAM) on the V213. Coefficients are stored for the calibrator to compensate for any gain error. These coefficients are applied during channel calibration to compute the true voltage applied to a channel. These coefficients are stored in the Correction Table starting at A32 offset 410_{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 A32 offset 440_{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 A32 offset 400_{16}. The following is a list of steps to follow in performing the V213 module calibration. There is also a programming example in Appendix D which performs the module calibration called "V213CAL.C". The documented code example will refer directly to the following steps:

1.) The V213 should be a minimum of a 30 minute warm-up period.
2.) Connect CAL to a precision voltage meter.
3.) Gain Calibration - repeat steps 3a to 3c for all 12 differential, calibrator voltages. In the following equation, these expected calibrator voltages are referred to as CALIBRATOR ranging from 10 volts to 2.0 mV. The calibrator voltages are set by writing to the Calibration Register (Operational register in A32 space at offset $0A_{16}$). A table of the precise register setting is available on page 41. Make sure to also set bit 13 of the Calibration Register to enable calibration voltage out of the front panel 2-pin LEMO connector. Also, make sure to wait for the calibrator voltage to settle before any measurements are taken.
   a.) Use precision meter to measure the positive calibration voltage (average 20 points). The measured positive calibration voltage is referred to as PCAL in the following equation.
   b.) Use precision meter to measure the negative calibration voltage (average 20 points). The measured negative calibration voltage is referred to as NCAL in the following equation.
   c.) Store error coefficient into the Correction Table 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.
4.) If offset calibration is unnecessary, skip to step #6.
5.) Offset Calibration
   a.) Write a zero to the Correction Table that contains the offset coefficients (A32 offsets 440_{16} through $4BE_{16}$).
   b.) Make sure a shorting connector is connected to all channels.
   c.) Create a scanlist with all channels set to a gain 2000. The mux rate should be set to

2 kHz for best accuracy. The scan rate is not critical, but it should be set to a valid rate (30 Hz is recommended).

d.) Have the DSP calibrate all channels using opcode 120_{16}.

e.) Measure each channel's offset error. Use the following equation to the measurement:

$$PER\_CHANNEL\_OFFSET = \frac{(COUNTS - OFFSET) \bullet 319.8242188 \mu Volts}{GAIN}$$

f.) Store offset error coefficient into the Correction Table as a 16-bit signed integer according to the following equation:

$$OFFSETCOEFFICIENT = PER\_CHANNEL\_OFFSET \bullet 10^{9}$$

6.) Store calibration date into the Correction Table.

Please note: Must wait a minimum of 3 mS after writing each part of the date before writing again.

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

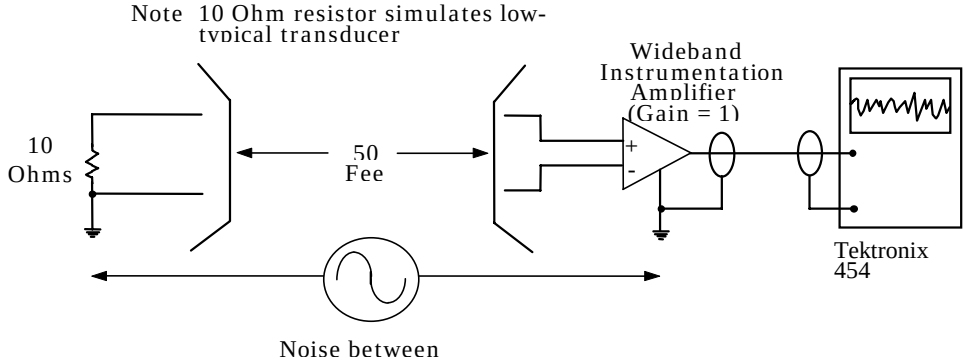
The level of noise, even when using balanced-pair cable, is affected by the exposure and the relative noise potential between the transmitting and receiving ends of the analog path. When analog signals are transmitted between racks that are in close proximity, noise performance is generally improved 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

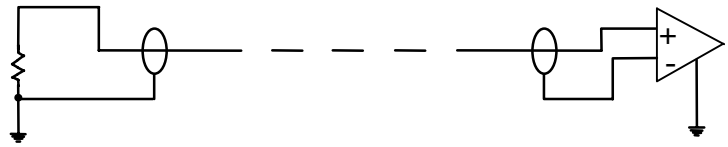
When receiving balanced analog signals, the following conclusions can be drawn:

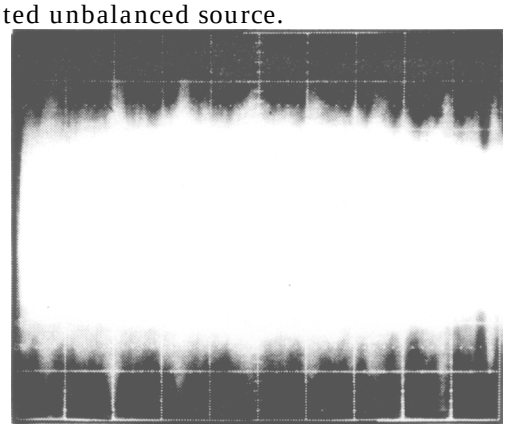
1. Shielded balanced-pair cable should be used with both ends grounded when the source and load are ground-referenced [Figure 4(a)].
2. If the source is isolated, then the source-end shield should be connected to the "common" of the source circuit [Figure 4(b)].
3. If the load is isolated, then the load-end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



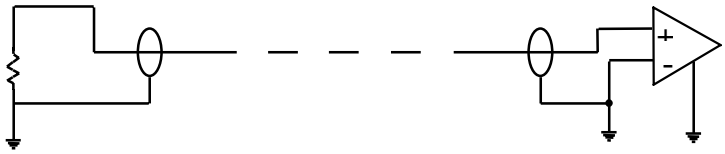
+

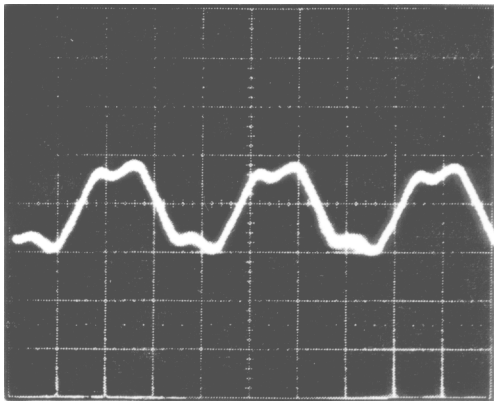
Figure 1. The noise test set-up with a simulated unbalanced source.



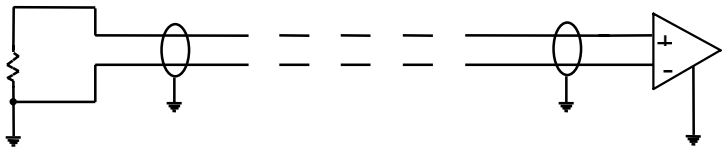


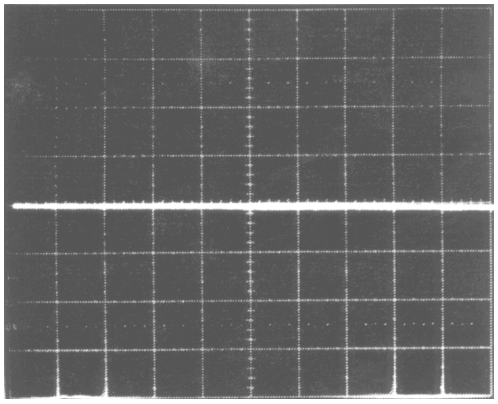
a) Coaxial cable, grounded at source only. 100mV/Div, 5 mS/Div





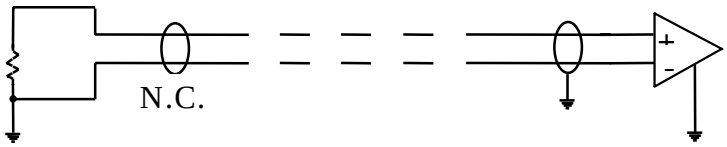
b) Coaxial cable, grounded at source and load. 100mV/Div, 5 mS/Div

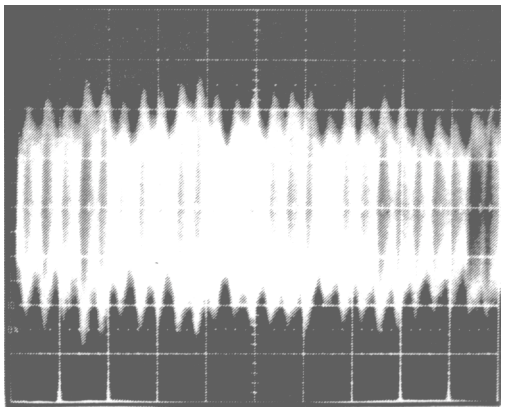




c) Shielded balanced pair, grounded at both ends 100mV/Div, 5 mS/Div

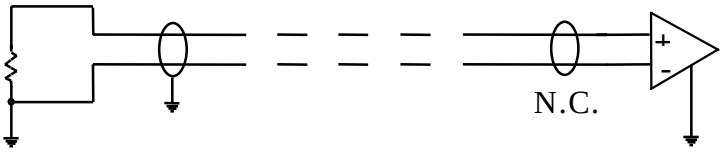
Figure 2. Noise comparison between coaxial cable and shielded balanced pair.

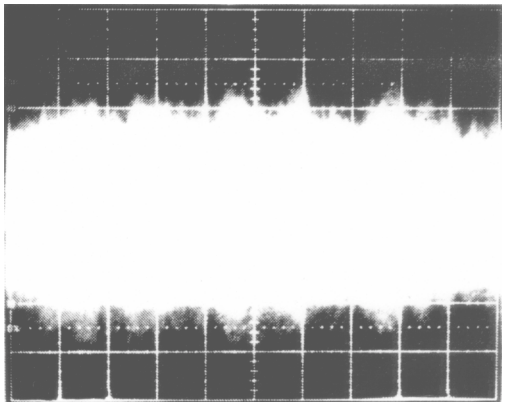




a) No shield connection at source

100 mV/Div, 5 mS/Div

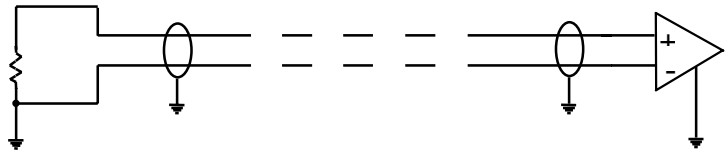


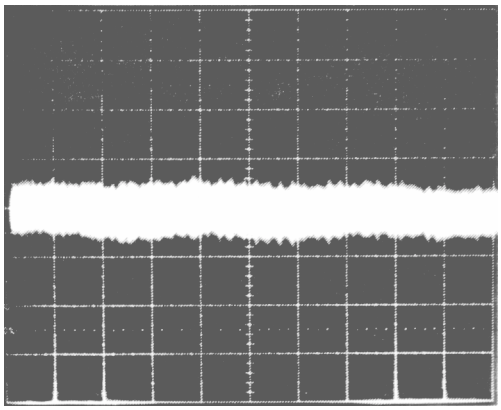


b) No shield connection at load.

100 mV/Div, 5 mS/Div

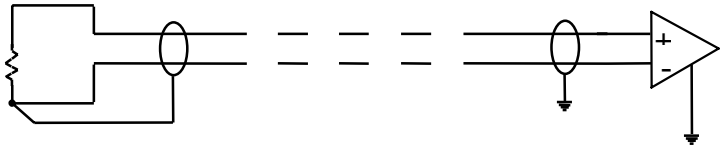
Figure 3. Shielded balanced pair with a ground connection at one end only

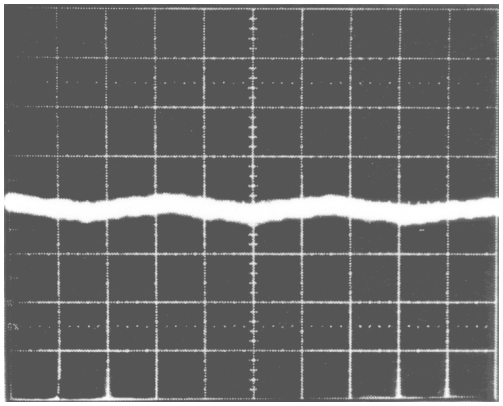




a) Grounded source

5 mV/Div, 5mS/Div

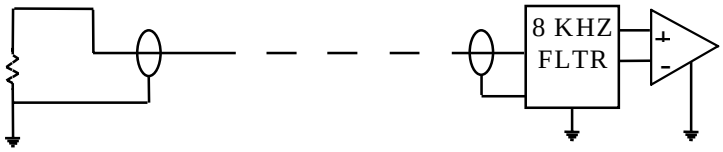


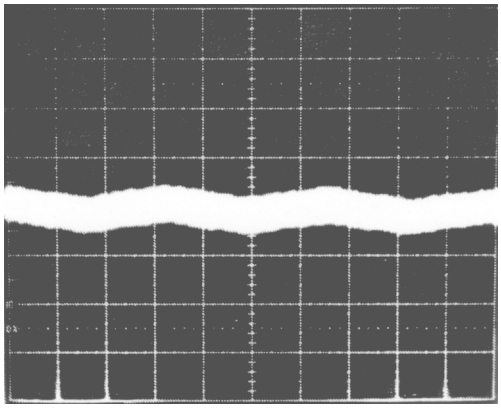


b) Floating source

5 mV/Div, 5 mS/Div

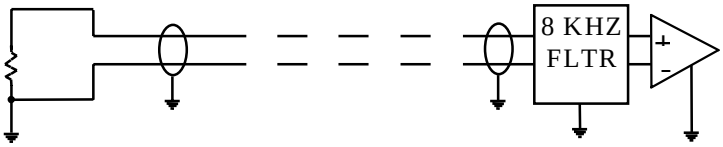
Figure 4. Noise comparison with a grounded and a floating source

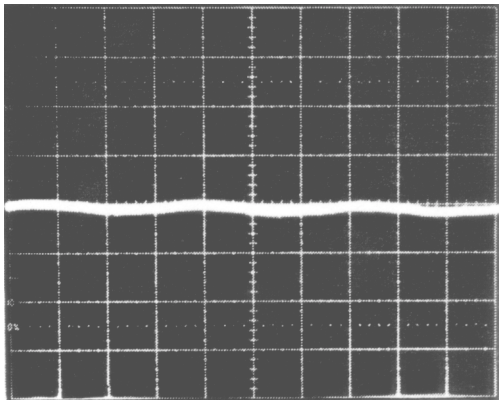




a) Coaxial cable, grounded at source only.

5 mV/Div, 5 mS/Div

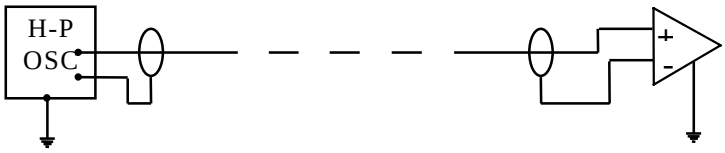


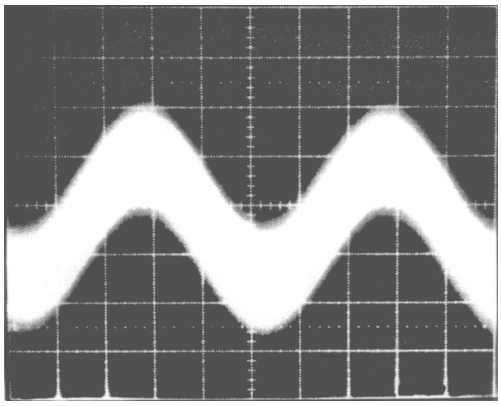


b) Shielded balanced pair, ground at both ends

5 mV/Div, 5mS/Div

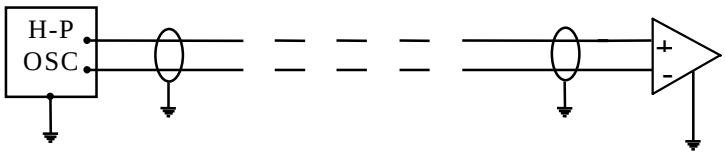
Figure 5. The offset of an 8 kHz filter on noise response.

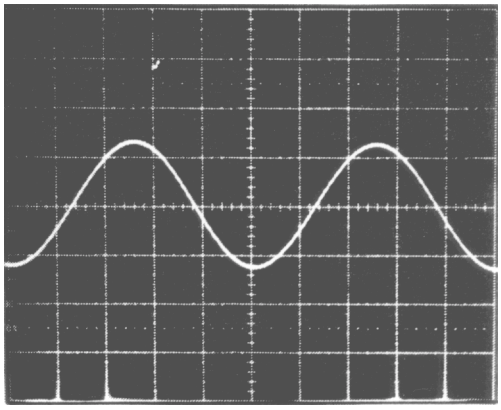




a) Coaxial cable, ground at source only

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.

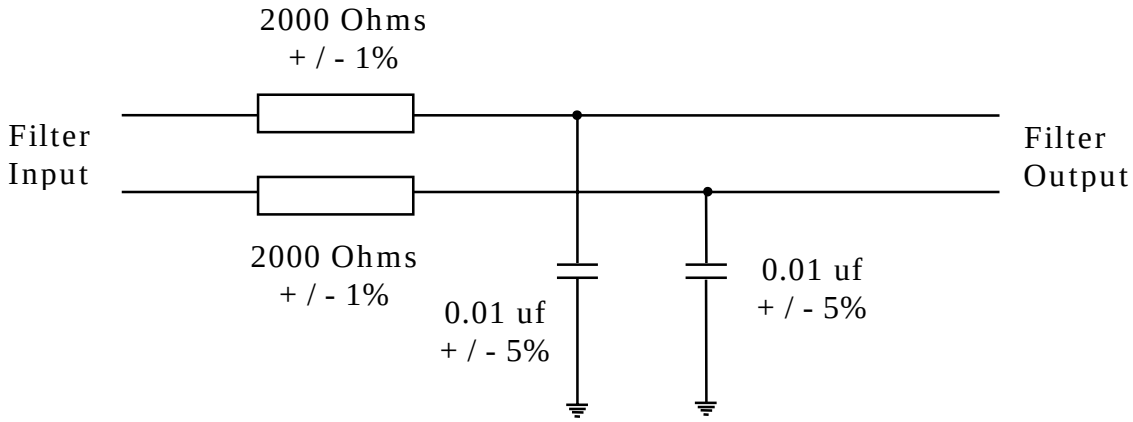


Figure 7. Details of the 8 kilohertz lowpass filter.

APPENDIX D - C/C++ Programming Examples Using National Instruments Drivers

V213SELF.C	This program example will perform a selftest on the V213 and report the results to the screen. In the report, the DSP on the V213 will indicate the I/O and digital expansion cards it supports, the type of expansion cards installed in the V213, as well as detailed results of the selftest.
V213CAL.C	This program example will calibrate the internal reference on the V213 using a precision multimeter connected to GPIB.
V213.C	This program example will setup a 32 channel scanlist, set the gain of all channels to 1, calibrate, and make a calibrated measurement from all channels.
V213SUBS.C	These subroutines are used by the previous programming examples.

```
/*****  Copyright (c) 1996, Kinetic Systems Corporation ********************/
/*                                                                          */
/* File:      V213SELF.C                                                    */
/*                                                                          */
/* Descr:   This program example will request the V213 to perform           */
/*            a selftest with a report.  The report is printed to the       */
/*            screen.                                                       */
/*                                                                  */
/****************************************************************************/
#include <stdio.h>

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

main()
{
  int16        error;
  uint16        data;

  error = InitVXIlibrary();
  if (error != 0) {
    printf("Error: InitVXIlibrary(), error code = %d \n",error);
    CloseVXIlibrary();
    exit(1);
    }

  /* Request selftest with report   */
  dsp_register_write(0x2);

  do {
    data = dsp_register_read();
    if (data != 0)
+       printf("%c",(char)data);
    }
  while (data != 0);

  error = CloseVXIlibrary();
  if (error != 0) {
    printf("Error: CloseVXIlibrary(), error code = %d \n",error);
    exit(1);
    }

  }
```

```
/***** Copyright (c) 1996, Kinetic Systems Corporation ******************/
/*                                                                      */
/* File:     V213CAL.C                                                  */
/*                                                                      */
/* Descr:    This program example will use an GPIB digital voltmeter    */
/*           to calibrate the V213.  The cal 2-pin LEMO located on       */
/*           the front panel must be connected to the dvm.              */
/*                                                                      */
/*           Please note that the Correction Table is normally          */
/*           protected from accidental erasure by a strap located near  */
/*           the logical address switch.  This "Cal" strap must be      */
/*           moved in order to gain access to the table.                */
/*                                                                      */
/************************************************************************/
#include<stdlib.h>
#include<stdio.h>

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

/* Setup V213 for 32 channels   */
#define NUMBER_CHAN     32

/* Number of averages during calibration        */
int16   const   AVERAGES = 20;

/* Set scan rate to 30 Hz       */
#define SCAN            30.0

/* LSB of the V213 is 20.96 Volts / 2^16        */
#define LSB             319.8242188E-6

uint16 dsp_register_read();
void dsp_register_write(uint16);

main() {
 char    offset_option;
 int16   i,j,error,V213,sdata;
 uint32  address,A32_V213,step;
 uint16  data,expdata;
 double  voltage,expect,range,high,low;
 int16   offset_error[NUMBER_CHAN+1];
 float   gain[NUMBER_CHAN+1];

 /* Array stores values for calibrator register        */
 uint16  cal[12] = { 0x0011, 0x0021, 0x0041, 0x0012, 0x0022, 0x0042,
                     0x0014, 0x0024, 0x0044, 0x0018, 0x0028, 0x0048 };
 float   volt[12] = { 10.0,5.0,2.0,1.0,0.5,0.2,0.1,0.05,0.02,0.01,0.005,0.002 };
```

```
error = InitVXIlibrary();
if (error != 0)
  {
  printf("Error: InitVXIlibrary(), error code = %d \n",error);
  CloseVXIlibrary();
  exit(1);
  }

/* Find KineticSystems module V213    */
error = FindDevLA("",0xF29,0x213,-1,-1,-1,-1,&V213);
if (error != 0)
  {
  printf("Error: FindDevLA(), error code = %d\n",error);
  CloseVXIlibrary();
  exit(1);
  }

error = GetDevInfo(V213,12,&A32_V213);
if (error != 0)
  {
  printf("Error: GetDevInfo(), error code = %d\n",error);
  CloseVXIlibrary();
  exit(1);
  }

/* Take V213 out of run mode  */
address = A32_V213;
error = VXIin(0x3,address,2,&data);
if (error != 0)
  {
  printf("Error: VXIin(), error code = %d\n",error);
  CloseVXIlibrary();
  exit(1);
  }
if ((data&0x1000) == 0x1000)
  {
  address = A32_V213 + 0x4;
  error = VXIin(0x3,address,2,&data);
  if (error != 0) {
    printf("Error: VXIin(), error code = %d\n",error);
    CloseVXIlibrary();
    exit(1);
    }
  }

/* Set for internal continuous scan, 2 kHz for highest accuracy */
data = 2;
address = A32_V213;
```

```
error = VXIout(0x3,address,2,data);
if (error != 0)
  {
  printf("Error: VXIout(), error code = %d\n",error);
  CloseVXIlibrary();
  exit(1);
  }

/* Set SCAN rate     */
data = (uint16) ((50000.0 / SCAN) - 0.5);
address = A32_V213 + 0x2;
error = VXIout(0x3,address,2,data);
if (error != 0)
  {
  printf("Error: VXIout(), error code = %d\n",error);
  CloseVXIlibrary();
  exit(1);
  }

/* Output sample clock on TTL Trigger Line 1  */
data = 0x80;
address = A32_V213 + 0x6;
error = VXIout(0x3,address,2,data);
if (error != 0)
  {
  printf("Error: VXIout(), error code = %d\n",error);
  CloseVXIlibrary();
  exit(1);
  }

/* Set number of averages for calibration      */
dsp_register_write(0x102);
dsp_register_write(AVERAGES);

/* Setup Scan List */
address = A32_V213 + 0x2000;
for (data=0; data<NUMBER_CHAN-1; data++)
  {
  error = VXIout(0x3,address,2,data);
  if (error != 0)
    {
    printf("Error: VXIout(), error code = %d\n",error);
    CloseVXIlibrary();
    exit(1);
    }
  address += 2;
  }
```

```
data |= 0x8000;
error = VXIout(0x3,address,2,data);
if (error != 0)
  {
  printf("Error: VXIout(), error code = %d\n",error);
  CloseVXIlibrary();
  exit(1);
  }

/* Setup Gain, all channels to 2000  */
data = 0x24;
for (address=A32_V213+0x300; address<=A32_V213+0x37E; address += 2)
  {
  error = VXIout(0x3,address,2,data);
  if (error != 0)
    {
    printf("Error: VXIout(), error code = %d\n",error);
    CloseVXIlibrary();
    exit(1);
    }
  }

/* V213 in run mode   */
address = A32_V213 + 0x4;
error = VXIin(0x3,address,2,&data);
if (error != 0) {
  printf("Error: VXIin(), error code = %d\n",error);
  CloseVXIlibrary();
  exit(1);
  }

printf("Calibrating V213\n");
for (i=0; i<12; i++)
  {
  printf("  Calibration voltage +- %f Volts\n",volt[i]);

  /* Set positive internal cal source, enable 2-pin LEMO      */
  address = A32_V213 + 0xA;
  data = cal[i] | 0x4080;
  error = VXIout(0x3,address,2,data);
  if (error != 0)
    {
    printf("Error: VXIout(), error code = %d \n",error);
    CloseVXIlibrary();
    exit(1);
    }

  /* Wait 1 second for relay to switch and voltage to settle  */
  wait(1000);
```

```
high = 0.0;
for (j=1; j<=AVERAGES; j++)
  {
  dvm_read(&voltage);
  high += voltage;
  }
high /= (double)AVERAGES;

/* Set negative internal cal source, enable 2-pin LEMO      */
address = A32_V213 + 0x000A;
data = cal[i] | 0x4100;
error = VXIout(0x3,address,2,data);
if (error != 0)
  {
  printf("Error: VXIout(), error code = %d \n",error);
  CloseVXIlibrary();
  exit(1);
  }

/* Wait 1 second for relay to switch and voltage to settle  */
wait(1000);

low = 0.0;
for (j=1; j<=AVERAGES; j++)
  {
  dvm_read(&voltage);
  low += voltage;
  }
low /= (double)AVERAGES;

/* Correction factor can be stored if error less than 3% */
voltage = high - low;
expect = volt[i] * 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",volt[i]);
  printf("       Voltage(+CAL) = %2.8f volts,  Voltage(-CAL) = %2.8f volts\n",high,low);
  printf("       DE = %2.8f volts +- %f, DR = %2.8f volts\n",expect,range,voltage);
  CloseVXIlibrary();
  exit(1);
  }
```

```
 address = A32_V213 + 0x410 + (i*2);
 data = (uint16)( ((voltage / expect) - 1.0) * 1000000.0);
 error = VXIout(0x3,address,2,data);
 if (error != 0)
  {
  printf("Error: VXIout(), error code = %d \n",error);
  CloseVXIlibrary();
  exit(1);
  }
 /* Must wait a minimum of 3 mS after writing correction table      */
 /* before the table can be re-accessed                             */
 wait(10);

 /* Verify data was written correctly in case "Cal" strap    */
 /* is in the incorrect position.                            */
 expdata = data;
 error = VXIin(0x3,address,2,&data);
 if (error != 0)
  {
  printf("Error: VXIin(), error code = %d \n",error);
  CloseVXIlibrary();
  exit(1);
  }
 if (data != expdata)
  {
  printf("Error: Data was not written to Correction Table.\n");
  printf("       Please check the cal strap \n");
  CloseVXIlibrary();
  exit(1);
  }
 }

printf("\nCalibrate Offset Error ?\n");
printf("  (Need shorting connector on all channels)\n");
scanf("%c",&offset_option);
```

```
if (offset_option == 'y' || offset_option == 'Y')
  {
  /* Remove old offset coefficients before calibration       */
  data = 0;
  for (address=A32_V213+0x440; address<=A32_V213+0x4BE; address+=2)
    {
    error = VXIout(0x3,address,2,data);
    if (error != 0)
        {
        printf("Error: VXIout(), error code = %d \n",error);
        CloseVXIlibrary();
        exit(1);
        }
    wait(10);
    }

  /* Request calibration for all channels      */
  dsp_register_write(0x120);
  dsp_register_write(0x0);

  /* Collect offset and gain errors for each channel    */
  for (i=1; i<=NUMBER_CHAN; i++)
    {
    offset_error[i] = dsp_register_read();
    gain[i] = 2000.0 * (1.0 + ((float)dsp_register_read() * 1.0E-6));
    }

  step = A32_V213 + 0x4000;
  for (i=1; i<=NUMBER_CHAN; i++)
    {
    voltage = 0.0;
    for (j=1; j<=AVERAGES; j++)
        {
        error = VXIin(0x3,step,2,&sdata);
        if (error != 0)
          {
          printf("Error: VXIin(), error code = %d\n",error);
          CloseVXIlibrary();
          exit(1);
          }
        /* remove offset and gain errors      */
        voltage += ((sdata - offset_error[i]) * LSB) / gain[i];
```

```
    /* look for falling edge of sample clock    */
    do
     {
     error = GetVXIbusStatusInd(-1,7,&data);
     if (error != 0)
       {
       printf("Error: GetVXIbusStatusInd(), error code = %d\n",error);
       CloseVXIlibrary();
       exit(1);
       }
     }
    while ( (data&1)==0 );
    do
     {
     error = GetVXIbusStatusInd(-1,7,&data);
     if (error != 0)
       {
       printf("Error: GetVXIbusStatusInd(), error code = %d\n",error);
       CloseVXIlibrary();
       exit(1);
       }
     }
    while ( (data&1)==1 );

    }
voltage /= (float)AVERAGES;

if ( voltage>10.0E-6 || voltage<-10.0E-6 )
    {
    printf("Error: Channel %d Offset error out of expected range (+- 10 uVolts)\n",i);
    printf("      Offset = %2.4f uVolts \n",voltage*1.0E6);
    printf("      Verify all channels are grounded\n");
    CloseVXIlibrary();
    exit(1);
    }
else
    {
    address = A32_V213 + 0x43E + (i*2);
    data = (uint16)( voltage * 1.0E9 );
    error = VXIout(0x3,address,2,data);
    if (error != 0)
     {
     printf("Error: VXIout(), error code = %d \n",error);
     CloseVXIlibrary();
     exit(1);
     }
```

```
        /* Must wait a minimum of 3 mS after writing correction table     */
        /* before the table can be re-accessed                            */
        wait(10);
        }
    step += 2;
    }
  }

/* Store month    */
printf("\nCurrent month (1-12) ?\n");
scanf("%d",&data);
address = A32_V213 + 0x400;
error = VXIout(0x3,address,2,data);
if (error != 0) {
  printf("Error: VXIout(), error code = %d \n",error);
  CloseVXIlibrary();
  exit(1);
  }
wait(10);

/* Store day    */
printf("\nCurrent day (1-31) ?\n");
scanf("%d",&data);
address = A32_V213 + 0x402;
error = VXIout(0x3,address,2,data);
if (error != 0) {
  printf("Error: VXIout(), error code = %d \n",error);
  CloseVXIlibrary();
  exit(1);
  }
wait(10);

/* Store year */
printf("\nCurrent year (example 1996) ?\n");
scanf("%d",&data);
address = A32_V213 + 0x404;
error = VXIout(0x3,address,2,data);
if (error != 0) {
  printf("Error: VXIout(), error code = %d \n",error);
  CloseVXIlibrary();
  exit(1);
  }

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

```
/***** Copyright (c) 1996, Kinetic Systems Corporation ******************/
/*                                                                      */
/* File:      V213.C                                                    */
/*                                                                      */
/* Descr:     This program example will set up the V213 for a           */
/*            particular gain, calibrate, and record calibrated         */
/*            measurements to the screen.                               */
/*                                                                      */
/************************************************************************/
#include<stdlib.h>
#include<stdio.h>

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

/* Setup V213 for 32 channels   */
#define NUMBER_CHAN     32

/* 2 kHz mux rate selected, for highest accuracy       */
#define MUX_RATE        2

/* Set scan rate to 30 Hz      */
#define SCAN            30.0

/* Gain for all channels are set to 1   */
#define GAINSETTING     1.0
#define GAIN            0

/* LSB of the V213 is 20.96 Volts / 2^16       */
#define LSB             319.8242188E-6

/* Number of averages requested during calibration      */
/* and while taking data                        */
#define AVERAGES        100

void main() {
uint16 data;
uint32 address,A32_V213;
int16  error,i,j,sdata,V213,offset_error[NUMBER_CHAN+1];
float  gain[NUMBER_CHAN+1],voltage;

  error = InitVXIlibrary();
  if (error != 0) {
    printf("Error: InitVXIlibrary(), error code = %d \n",error);
    CloseVXIlibrary();
    exit(1);
    }
```

```
/* Find KineticSystems module V213    */
error = FindDevLA("",0xF29,0x213,-1,-1,-1,-1,&V213);
if (error != 0) {
  printf("Error: FindDevLA(), error code = %d\n",error);
  CloseVXIlibrary();
  exit(1);
  }

error = GetDevInfo(V213,12,&A32_V213);
if (error != 0) {
  printf("Error: GetDevInfo(), error code = %d\n",error);
  CloseVXIlibrary();
  exit(1);
  }

printf("V213 - Setup\n");

/* Take V213 out of run mode  */
address = A32_V213;
error = VXIin(0x3,address,2,&data);
if (error != 0) {
  printf("Error: VXIin(), error code = %d\n",error);
  CloseVXIlibrary();
  exit(1);
  }
if ((data&0x1000) == 0x1000) {
  address = A32_V213 + 0x4;
  error = VXIin(0x3,address,2,&data);
  if (error != 0) {
    printf("Error: VXIin(), error code = %d\n",error);
    CloseVXIlibrary();
    exit(1);
    }
  }

/* Set for internal continuous scan */
data = MUX_RATE;
address = A32_V213;
error = VXIout(0x3,address,2,data);
if (error != 0) {
  printf("Error: VXIout(), error code = %d\n",error);
  CloseVXIlibrary();
  exit(1);
  }
```

```
/* Set SCAN rate     */
data = (uint16) ((50000.0 / SCAN) - 0.5);
address = A32_V213 + 0x2;
error = VXIout(0x3,address,2,data);
if (error != 0) {
  printf("Error: VXIout(), error code = %d\n",error);
  CloseVXIlibrary();
  exit(1);
  }

/* Output sample clock on TTL Trigger Line 1  */
data = 0x80;
address = A32_V213 + 0x6;
error = VXIout(0x3,address,2,data);
if (error != 0) {
  printf("Error: VXIout(), error code = %d\n",error);
  CloseVXIlibrary();
  exit(1);
  }

printf("V213 - Calibrating\n");

/* Set number of averages for calibration      */
dsp_register_write(0x102);
dsp_register_write(AVERAGES);

/* Setup Scan List */
address = A32_V213 + 0x2000;
for (data=0; data<NUMBER_CHAN-1; data++)
  {
  error = VXIout(0x3,address,2,data);
  if (error != 0) {
    printf("Error: VXIout(), error code = %d\n",error);
    CloseVXIlibrary();
    exit(1);
    }
  address += 2;
  }
data |= 0x8000;
error = VXIout(0x3,address,2,data);
if (error != 0) {
  printf("Error: VXIout(), error code = %d\n",error);
  CloseVXIlibrary();
  exit(1);
  }
```

```
/* Setup Gain, all channels   */
data = GAIN;
for (address=A32_V213+0x300; address<=A32_V213+0x37E; address += 2)
  {
  error = VXIout(0x3,address,2,data);
  if (error != 0) {
    printf("Error: VXIout(), error code = %d\n",error);
    CloseVXIlibrary();
    exit(1);
    }
  }

/* Request calibration for all channels       */
dsp_register_write(0x120);
dsp_register_write(0x0);

/* Collect offset and gain errors for each channel    */
for (i=1; i<=NUMBER_CHAN; i++)
  {
  offset_error[i] = dsp_register_read();
  gain[i] = GAINSETTING * (1.0 + ((float)dsp_register_read() * 1.0E-6));
  }

address = A32_V213 + 0x4000;
for (i=1; i<=NUMBER_CHAN; i++)
  {
  voltage = 0.0;
  for (j=1; j<=AVERAGES; j++)
    {
    error = VXIin(0x3,address,2,&sdata);
    if (error != 0) {
        printf("Error: VXIin(), error code = %d\n",error);
        CloseVXIlibrary();
        exit(1);
        }
    /* remove offset and gain errors      */
    voltage += ((sdata - offset_error[i]) * LSB) / gain[i];

    /* look for falling edge of sample clock    */
    do
        {
        error = GetVXIbusStatusInd(-1,7,&data);
        if (error != 0) {
          printf("Error: GetVXIbusStatusInd(), error code = %d\n",error);
          CloseVXIlibrary();
          exit(1);
          }
        }
    while ( (data&1)==0 );
```

```
    do
        {
        error = GetVXIbusStatusInd(-1,7,&data);
        if (error != 0) {
          printf("Error: GetVXIbusStatusInd(), error code = %d\n",error);
          CloseVXIlibrary();
          exit(1);
          }
        }
    while ( (data&1)==1 );

    }
  voltage /= (float)AVERAGES;
  printf("Channel %d  %f Volts\n",i,voltage);
  address += 2;
  }
}
```

```
/***** Copyright (c) 1996, Kinetic Systems Corporation ******************/
/*                                                                      */
/* File:      V213SUBS.C                                                */
/*                                                                      */
/* Descr:     Subroutines used by V213 program examples                 */
/*                                                                      */
/************************************************************************/
#include <stdio.h>
#include <stdlib.h>

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

uint16 dsp_register_read() {
 uint32    address,A32_V213;
 uint16    data;
 int16     error,V213;

 /* Find KineticSystems module V213    */
 error = FindDevLA("",0xF29,0x213,-1,-1,-1,-1,&V213);
 if (error != 0) {
   printf("Error: FindDevLA(), error code = %d\n",error);
   CloseVXIlibrary();
   exit(1);
   }

 error = GetDevInfo(V213,12,&A32_V213);
 if (error != 0) {
   printf("Error: GetDevInfo(), error code = %d\n",error);
   CloseVXIlibrary();
   exit(1);
   }
```

```
  /* Wait I/O full flag to go true            */
  /* For simplicity, no timeout is provided.   */
  address = A32_V213;
  do
    {
    error = VXIin(0x3,address,2,&data);
    if (error != 0) {
      printf("Error: VXIin(), error code = %d\n",error);
      CloseVXIlibrary();
      exit(1);
      }
    }
  while ( ((data&0x2000)==0) );

  address = A32_V213 + 0x12;
  error = VXIin(0x3,address,2,&data);
  if (error != 0) {
    printf("Error: VXIin(), error code = %d\n",error);
    CloseVXIlibrary();
    exit(1);
    }

  return data;
  }

void dsp_register_write(uint16 data) {
  uint32   address,A32_V213;
  int16    error,V213;

  /* Find KineticSystems module V213   */
  error = FindDevLA("",0xF29,0x213,-1,-1,-1,-1,&V213);
  if (error != 0) {
    printf("Error: FindDevLA(), error code = %d\n",error);
    CloseVXIlibrary();
    exit(1);
    }

  error = GetDevInfo(V213,12,&A32_V213);
  if (error != 0) {
    printf("Error: GetDevInfo(), error code = %d\n",error);
    CloseVXIlibrary();
    exit(1);
    }
```

```
address = A32_V213 + 0x12;
error = VXIout(0x3,address,2,data);
if (error != 0) {
  printf("Error: dsp_register_write(), error code = %d\n",error);
  CloseVXIlibrary();
  exit(1);
  }

error = dsp_register_read();
if (error != 0) {
  printf("Error: Incorrect response from DSP\n");
  printf("       Expect = 0 (Operation Complete), Received = %d\n",error);
  CloseVXIlibrary();
  exit(1);
  }
}
```