INSTRUCTION MANUAL

Sensoray Model 421
ISAbus Analog/Digital I/O Card

10/27/1995

Sensoray Co., Inc.
7313 SW Tech Center Dr., Tigard, Oregon 97223
voice: 503.684.8005, fax: 503.684.8164, e-mail: sales@sensoray.com
www.sensoray.com

M A N U A L 4 2 1

Table of Contents

Limited Warranty

Sensoray Company, Incorporated (Sensoray) warrants the Model 421 hardware to be free from defects in material and workmanship and perform to applicable published Sensoray specifications for one year from the date of shipment to purchaser. Sensoray will, at its option, repair or replace equipment that proves to be defective during the warranty period. This warranty includes parts and labor.

The warranty provided herein does not cover equipment subjected to abuse, misuse, accident, alteration, neglect, or unauthorized repair or installation. Sensoray shall have the right of final determination as to the existance and cause of defect.

As for items repaired or replaced under warranty, the warranty shall continue in effect for the remainder of the original warranty period, or for ninety days following date of shipment by Sensoray of the repaired or replaced part, whichever period is longer.

A Return Material Authorization (RMA) number must be obtained from the factory and clearly marked on the outside of the package before any equipment will be accepted for warranty work. Sensoray will pay the shipping costs of returning to the owner parts which are covered by warranty.

Sensoray believes that the information in this manual is accurate. The document has been carefully reviewed for technical accuracy. In the event that technical or typographical errors exist, Sensoray reserves the right to make changes to subsequent editions of this document without prior notice to holders of this edition. The reader should consult Sensoray if errors are suspected. In no event shall Sensoray be liable for any damages arising out of or related to this document or the information contained in it.

EXCEPT AS SPECIFIED HEREIN, SENSORAY MAKES NO WARRANTIES, EXPRESS OR IMPLIED, AND SPECIFICALLY DISCLAIMS ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. CUSTOMER'S RIGHT TO RECOVER DAMAGES CAUSED BY FAULT OR NEGLIGENCE ON THE PART OF SENSORAY SHALL BE LIMITED TO THE AMOUNT THERETOFORE PAID BY THE CUSTOMER. SENSORAY WILL NOT BE LIABLE FOR DAMAGES RESULTING FROM LOSS OF DATA, PROFITS, USE OF PRODUCTS, OR INCIDENTAL OR CONSEQUENTIAL DAMAGES, EVEN IF ADVISED OF THE POSSIBILITY THEROF.

Special Handling Instructions

The Model 421 circuit board contains CMOS circuitry that is sensitive to Electrostatic Discharge (ESD). Special care should be taken in handling, transporting, and installing the 421 to prevent ESD damage to the board. In particular:

— Do not remove the 421 from its protective antistatic bag until you are ready to configure the board for installation.

— Handle the 421 only at grounded, ESD protected stations.

— Remove power from the ISAbus before installing or removing the 421 circuit board.

Introduction

The Sensoray model 421 is an ISAbus I/O card that performs several functions commonly required in embedded applications. On-board peripherals include:

— Programmable watchdog timer.

— 48 digital I/O channels.

— 12-bit analog-to-digital converter with eight differential input channels and resistor-programmed gain.

— Four 12-bit analog output channels.

— Three 16-bit counter channels optimized for interface to incremental encoders.

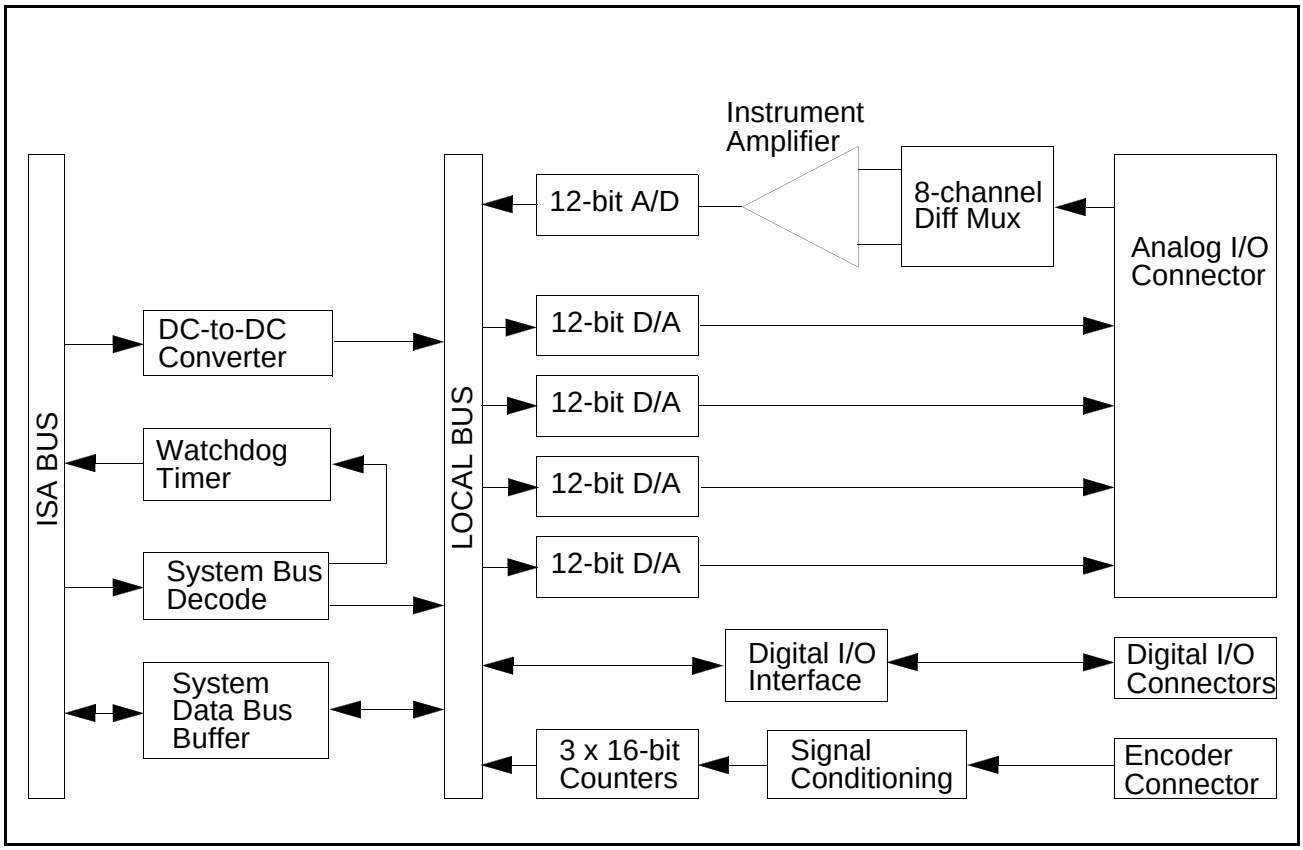
The 421 is powered from the ISAbus 5-volt supply. No auxilliary 12- or 15-volt supplies are required.

Power for linear circuitry is derived from an on-board DC-to-DC converter. Wherever possible, low power CMOS circuitry is used to minimize system power consumption and enhance reliability.

The board occupies a 32-byte block of I/O addresses in the ISAbus 16-bit I/O space. Programming shunts may be installed to map the board to any 32-byte address boundary.

Four headers are provided for connecting on-board peripherals to external circuitry. Right-angle headers are used to satisfy the tight clearances mandated by modern high-density system designs. All headers are supplied with latches to ensure reliable operation in vibrating environments.

Block Diagram



Selecting a Base Address

The 421 board occupies a block of 32 consecutive addresses in the ISAbus 16-bit I/O space. The address range occupied by the board must begin on a 32-byte address boundary. The first address in the block — the "base address" — is assigned by DIP switches S1 and S2.

Care must be exercised to ensure that no other devices use addresses in the address range assigned to the 421. Use the tables shown below to determine the proper address switch settings for your application.

First Address Nibble (MSB)

Adrs	S1-15	S1-14	S1-13	S1-12
0	**On**	**On**	**On**	**On**
1	On	On	On	Off
2	On	On	Off	On
3	On	On	Off	Off
4	On	Off	On	On
5	On	Off	On	Off
6	On	Off	Off	On
7	On	Off	Off	Off
8	Off	On	On	On
9	Off	On	On	Off
A	Off	On	Off	On
B	Off	On	Off	Off
C	Off	Off	On	On
D	Off	Off	On	Off
E	Off	Off	Off	On
F	Off	Off	Off	Off

Second Address Nibble

Adrs	S1-11	S1-10	S1-9	S1-8
0	On	On	On	On
1	On	On	On	Off
2	On	On	Off	On
3	**On**	**On**	**Off**	**Off**
4	On	Off	On	On
5	On	Off	On	Off
6	On	Off	Off	On
7	On	Off	Off	Off
8	Off	On	On	On
9	Off	On	On	Off
A	Off	On	Off	On
B	Off	On	Off	Off
C	Off	Off	On	On
D	Off	Off	On	Off
E	Off	Off	Off	On
F	Off	Off	Off	Off

Third Address Nibble

Adrs	S2-7	S2-6	S2-5
0	On	On	On
2	On	On	Off
4	On	Off	On
6	On	Off	Off
8	Off	On	On
A	**Off**	**On**	**Off**
C	Off	Off	On
E	Off	Off	Off

For example, to program the board base address to 03A0 hex, set the address switches as follows: Turn on S1 switches10-15 and S2 switch 6. Turn off S1 switches 8 and 9. S2 switches 2 and 4.

Note: default switch settings are shown in **bold**

Status Register

A status register, located at the board base address + 11, makes available four status bits for access by the ISAbus host processor. The status register is organized as follows:

STATUS (11):

x	x	x	x	UN	DE	BZ	FT

Status bits are all active high and have the following definitions:

UN — indicates A/D converter is programmed for unipolar operation.

BZ — indicates that the A/D converter is busy digitizing an analog input.

DE — indicates that all four DAC output channels have been enabled via the CHCTRL port.

PD — indicates 421 is in the low-power standby mode, controlled by the RELAY2 port.

Fault Indicator

A red light-emitting diode located near the upper-left corner of the circuit board indicates reset and fault conditions.

In normal operation, the indicator is turned on only during system or 421 local reset. The indicator should be off at all other times.

If the fault indicator remains on for more than one second following a reset, a fault condition may exist on the board that requires servicing.

Before servicing the unit, make sure that your ISAbus host CPU is not invoking repeated soft resets of the 421 (which will also keep the indicator turned on).

Reset

The 421 may be reset by either a “hard reset” from the ISAbus system reset signal or a “soft reset” under software control of the ISAbus master.

Soft Reset

A “soft reset” may be performed at any time. This causes a local reset on only the 421 and will not reset any other devices on the ISAbus. A soft reset is invoked by writing to the RESET port.

RESET (11):

0	0	0	0	0	0	0	0

The value written to the RESET port is ignored, but should be all zeros for compatibility with future product enhancements.

Default State Following a Reset

Both hard and soft resets force the 421 to the following condition:

— The watchdog timer is disabled.

— All DAC outputs are disabled.

— All encoder counters are reset to zero.

— Encoder counters default to Mode 0.

— All relay channels are turned off.

Watchdog Timer

Embedded systems often include a watchdog timer to regain program control following an unplanned loss of control by the ISAbus master. In such systems, the CPU is responsible for periodically refreshing the watchdog timer to prevent a timeout. Should the CPU crash, the watchdog will not be refreshed and will eventually timeout. The resulting timeout will restart the CPU.

The 421 watchdog timer has a guaranteed minimum timeout of 630 milliseconds. Consequently, the interval between any two timer refreshes must not exceed 630 milliseconds.

Some CPU’s are not able to refresh the watchdog at the mandatory minimum rate during system boot or critical I/O operations. To accomodate these situations, the 421 provides a control register for enabling and disabling the watchdog under program control.

The first watchdog refresh must occur no later than 630 milliseconds after enabling the timer. The watchdog is automatically disabled by a soft reset or system-wide hard reset.

Enabling/Disabling the Watchdog

The watchdog is enabled and disabled by writing to the CHCTRL port at board base address + 12. The port is structured as follows:

CHCTRL (12):

0	0	0	0	1	0	D	W

Set W to logic one to enable the watchdog or set to zero to disable the watchdog. Note that D is used to enable and disable DAC outputs. Make sure D is set properly when changing the watchdog enable bit so that you don’t inadvertently enable or disable the DAC outputs.

Refreshing the Watchdog

The watchdog is refreshed by writing to the HITDOG port. The data value written to this port is ignored by the 421, but should be all zeros for compatibility with future product enhancements.

HITDOG (15):

0	0	0	0	0	0	0	0

Routing the Watchdog to the ISAbus

A watchdog timeout may be used to generate a system reset. Two-pin connector P1 may be connected — via user-supplied two-conductor cable — to your ISAbus active-low external reset input.

You may leave connector P1 disconnected if you will not be using the watchdog function.

Encoder Interface

Three counter channels reside on the 421. Each channel is optimized for incremental encoders by providing the following elements:

— Input buffers - interfaces to TTL, CMOS, or RS422 signals.

— Decode logic - detects and converts encoder edges into clock and direction signals.

— 16-bit up/down counter - maintains encoder position without resorting to multiple counter channels.

— Power - 5-volts is accessible to power encoders.

Additional logic is incorporated to implement synchronous transfer of encoder counts into a 16-bit holding register. This feature assures error-free acquisition of encoder counts by the ISAbus master.

Phase Inputs

Each counter channel has two input signals, called the “A” and “B” phases. Depending on the application, one or both of these signals may be connected to an encoder.

If both input phases are used, the phases are assumed to be Quadrature encoded, meaning that they are 90 degrees out of phase with each other. Counter channels will count both up and down by decoding the timing relationship of the two phase inputs.

If only one phase input is used, the input is said to be Single-phase. In this case, counter channels will count either up or down, but not both. This configuration is typically used to count pulses from non-encoder devices that produce a single clock output.

Quadrature encoded inputs have advantages over single-phase inputs. Counters will not accumulate errors when an encoder changes direction or dithers about a state transition boundary. Also, it is possible to increase the effective resolution of an encoder by clocking the counters at a multiple of the single-phase clock rate.

Mode Selection

Counter channels may be configured for any of eight operating modes. A mode register is provided to select the operating mode of the three counter channels. All channels must be configured for the same operating mode.

By selecting a counter mode, you are specifying a combination of counter input type (quadrature encoded or single-phase), clock multiplier (times 1, 2 or 4), and count direction (normal or reverse).

421 Reset

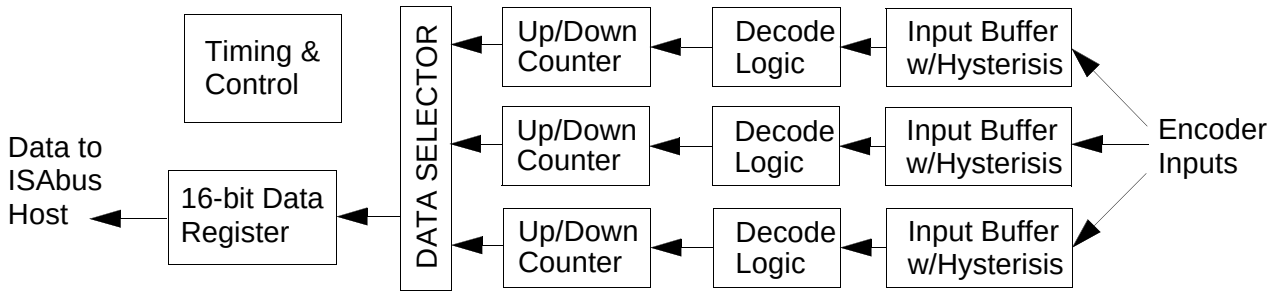
Hardware or software reset of the 421 will asynchronously zero all three encoder counters, the counter mode register (thereby selecting counter operational mode 0) and the 16-bit data holding register. Channel logic is re-enabled for counting upon termination of the reset pulse.

Rollover

All encoder counters will increment from FFFF to 0000 when counting up, and will decrement from 0000 to FFFF when counting down.

No interrupts or status flags are available to notify the ISAbus master of a rollover event. The master should read encoder position data with sufficient frequency to guarantee the validity of position data.

Encoder Interface Block Diagram



Command Functions

The CNTCTL port is used to invoke all counter command and configuration functions. Three counter operations may be accessed through this port: Latch counts, Reset counts, and Mode set. See the box below for details.

The **Latch** command synchronously transfers the specified counter's contents to the 16-bit data holding register. A Latch command must be executed before the contents of a counter may be read. Note that counters may not be read directly — counter contents must first be transferred to the holding register, then read from the holding register.

The **Reset** command resets the specified counter to zero. The affected counter is automatically reenabled for counting within 2 microseconds following a Reset command.

Note that the act of writing a Reset command to the CNTCTL port *triggers* a counter reset operation. After invoking a counter Reset command, no further action is required by the ISAbus host processor to enable counting on the affected channel.

The **Mode** command specifies the operating mode for all three counter channels. Three functional attributes are specified by the selected mode: clock multiplier, number of active phases and count direction.

The clock multiplier determines the number of count events per cycle on the "A" phase input. A *count event* is a phase input state transition that causes either an increment or decrement of the associated encoder counter. For example, a "x2" multiplier means that 2 count events will occur for each cycle of the "A" phase input. Refer to the counter timing diagram for more information.

Either one or two active phases may be selected. One of the 2-phase modes must be selected when using Quadrature-encoded inputs. A 1-phase mode must be selected when using a single-phase clock source.

"Count direction" is specified as either *normal* or *reverse*. Select the counting direction (see timing diagram) best suited for your application.

CNTCTL (14):

0	0	0	0	M_1	M_0	S_1	S_0

M_1M_0 specifies the operation to be performed:

M_1M_0	Function
00	Latch counter S_1S_0 contents
01	Reset counter S_1S_0 to zero
1x	Select operating mode $M_0S_1S_0$

When M_1=0, S_1S_0 selects the counter channel to be operated on:

S_1S_0	Selected Encoder Channel
00	Channel 0
01	Channel 1
10	Channel 2
11	Reserved for future use

When M_1=1, $M_0S_1S_0$ selects the counter mode for all three counter channels:

Mode		Counter Function		
	$M_0S_1S_0$	Clk	Phases	Count Direction
0	000	x2	2	Normal
1	001	x4	2	Normal
2	010	x1	2	Normal
3	011	x1	1	Normal
4	100	x2	2	Reverse
5	101	x4	2	Reverse
6	110	x1	2	Reverse
7	111	x1	1	Reverse

Reading Latched Encoder Counts

After executing a Latch command, the latched data may be read from ports CNTLSB and CNTMSB.

CNTLSB (14): | d_7 | d_6 | d_5 | d_4 | d_3 | d_2 | d_1 | d_0 |

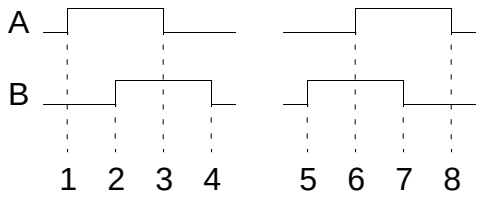
CNTMSB (15): | d_{15} | d_{14} | d_{13} | d_{12} | d_{11} | d_{10} | d_9 | d_8 |

Port CNTLSB contains the least-significant byte of the encoder counts and port CNTMSB contains the most-significant byte.

It makes no difference which byte is read first. The data value remains latched until the next Latch command.

Timing Diagram

Encoder counters change state (count up or down) upon detection of any valid input transition. All possible transitions are shown in the timing diagram below. For example, transition number 1 specifies a rising edge on the "A" phase while "B" is held low.



Valid transitions depend on the operating mode. The table to the right lists valid transitions for each mode. In mode 0, for example, a counter will count up only at transitions 6 and 8, and will count down only at transitions 1 and 3.

Valid count transitions vs. counter operating mode

Counter Configuration		Valid Count Transitions	
Mode	Function	Up	Down
0	Quadrature x2	6, 8	1, 3
1	Quadrature x4	5, 6, 7, 8	1, 2, 3, 4
2	Quadrature x1	8	1
3	Single Phase x1	---	1
4	Quadrature x2	1, 3	6, 8
5	Quadrature x4	1, 2, 3, 4	5, 6, 7, 8
6	Quadrature x1	1	8

Encoder Connections

Connector P3 is used to make all electrical connections to external encoders. Each counter channel is allocated six pins on connector P3: +5V, GND, +A, -A, +B and -B.

Power Connections

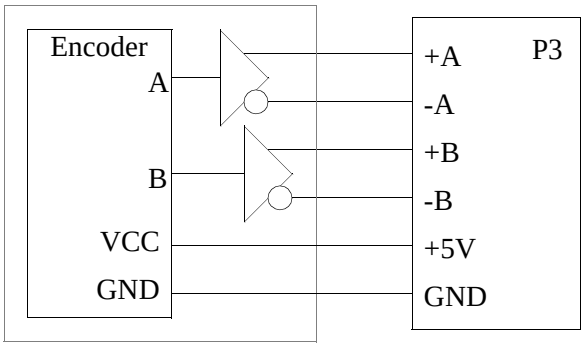
The +5V and GND pins may optionally be used to power an external encoder. If an external encoder power source is used, you must connect GND to the external power supply return. Failure to connect these returns may damage the 421 board.

Clock Connections

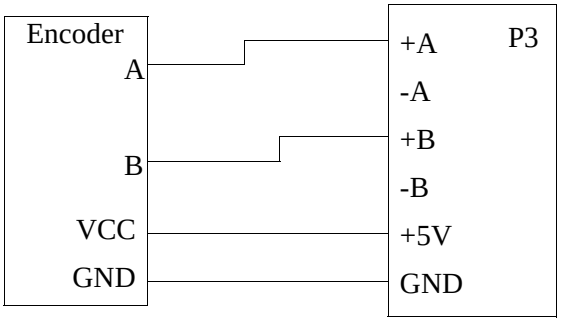
Each counter channel is assigned four pins on connector P3 for connection to the phase inputs. Each phase makes two electrical connections to P3 in the form of a differential RS422 pair. The A phase inputs consist of +A and -A, while the B inputs consist of +B and -B. Connections to these four inputs depend on the type of device to be interfaced.

For a variety of reasons, RS422 termination resistors are not supplied on the 421 board. If your encoder (or other pulse source) employs RS422 drivers, you may need to supply external termination resistors near connector P3 for proper operation. Refer to your encoder manufacturer's documentation for recommended RS422 termination circuitry and practice.

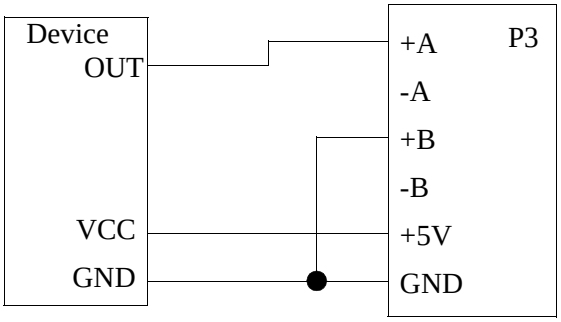
Some encoders have built-in RS422 drivers. If you are using this type of encoder, connect the encoder A outputs to the +A and -A inputs, and connect the encoder B outputs to the +B and -B inputs:



Many encoders utilize single-ended TTL or CMOS compatible outputs. If you are using this type of encoder, connect the encoder A output to the +A input and connect the encoder B output to the +B input. Leave the -A and -B inputs disconnected:



Counter channels may be interfaced to devices that produce a single output phase. In such cases, the counters are configured for operation in either mode 3 or mode 7. Connect the device output to the +A input. Connect the +B input to GND. Leave the -A and -B inputs disconnected:



Connector P3 Pinout:

Pin	Name	Function
1	+5V	Channel 0 +5V power
2	$+A_0$	Channel 0 phase A input (+)
3	GND	Channel 0 5V return
4	$+B_0$	Channel 0 phase B input (+)
5	+5V	Channel 1 +5V power
6	$+A_1$	Channel 1 phase A input (+)
7	GND	Channel 1 5V return
8	$+B_1$	Channel 1 phase B input (+)
9	+5V	Channel 2 +5V power
10	$+A_2$	Channel 2 phase A input (+)
11	GND	Channel 2 5V return
12	$+B_2$	Channel 2 phase B input (+)
13		-- no connect --
14		-- no connect --
15	$-A_0$	Channel 0 phase A input (-)
16	$-A_1$	Channel 1 phase A input (-)
17	$-A_2$	Channel 2 phase A input (-)
18	$-B_0$	Channel 0 phase B input (-)
19	$-B_1$	Channel 1 phase B input (-)
20	$-B_2$	Channel 2 phase B input (-)

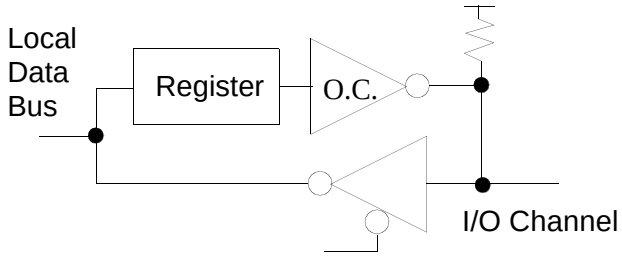
Relay Interface

The 421 provides 48 high-current digital I/O channels for direct interface to industry standard relay racks.

As shown in the diagram to the right, digital I/O channels are physically structured as open-collector buffers (with pullup resistors) looped back to the local data bus by tri-state buffers.

Channel data is active-high on the local bus side, and active-low on the I/O channel side.

One of 48 Digital I/O Channels



RELAYx Ports

The 48 digital I/O channels are organized as two banks of three byte-wide ports. Each port accesses eight relay channels simultaneously, and each bank of three ports addresses 24 relay channels.

RELAY0 (8):	R_7	R_6	R_5	R_4	R_3	R_2	R_1	R_0
RELAY1 (9):	R_{15}	R_{14}	R_{13}	R_{12}	R_{11}	R_{10}	R_9	R_8
RELAY2 (10):	R_{23}	R_{22}	R_{21}	R_{20}	R_{19}	R_{18}	R_{17}	R_{16}
RELAY3 (16):	R_7	R_6	R_5	R_4	R_3	R_2	R_1	R_0
RELAY4 (17):	R_{15}	R_{14}	R_{13}	R_{12}	R_{11}	R_{10}	R_9	R_8
RELAY5 (18):	R_{23}	R_{22}	R_{21}	R_{20}	R_{19}	R_{18}	R_{17}	R_{16}

Each RELAYx port may be read from or written to at any time. Reading from a RELAYx port returns the physical state of the channels associated with that port. Writing to a RELAYx port programs the associated relay channel output registers.

Note: the value read from a relay channel may not be the same as the value stored in that channels output register. Specifically, the values will differ if an output register is programmed to 0 and an external sink pulls the corresponding channel down to 0V.

Configuring I/O Type

Each channel may be independently configured as either an input or output via software. To configure a channel as an input, set the associated channel register to 0.

Connections

All electrical connections are made through 50-pin connectors P5 and P6. On each of these connectors, one pin is assigned to each active-low I/O channel. All even pins are connected to ISAbus five volt return. In addition, pin 49 may be used to supply limited five volt power (<100mA) to an external relay rack.

Connectors P5 and P6: Digital I/O

Pin	Name	Function
1	CH23	I/O Channel 23
3	CH22	I/O Channel 22
5	CH21	I/O Channel 21
7	CH20	I/O Channel 20
9	CH19	I/O Channel 19
11	CH18	I/O Channel 18
13	CH17	I/O Channel 17
15	CH16	I/O Channel 16
17	CH15	I/O Channel 15
19	CH14	I/O Channel 14
21	CH13	I/O Channel 13
23	CH12	I/O Channel 12
25	CH11	I/O Channel 11
27	CH10	I/O Channel 10
29	CH9	I/O Channel 9
31	CH8	I/O Channel 8
33	CH7	I/O Channel 7
35	CH6	I/O Channel 6
37	CH5	I/O Channel 5
39	CH4	I/O Channel 4
41	CH3	I/O Channel 3
43	CH2	I/O Channel 2
45	CH1	I/O Channel 1
47	CH0	I/O Channel 0
49	+5V	+5V power
Even	GND	5V return

D/A Interface

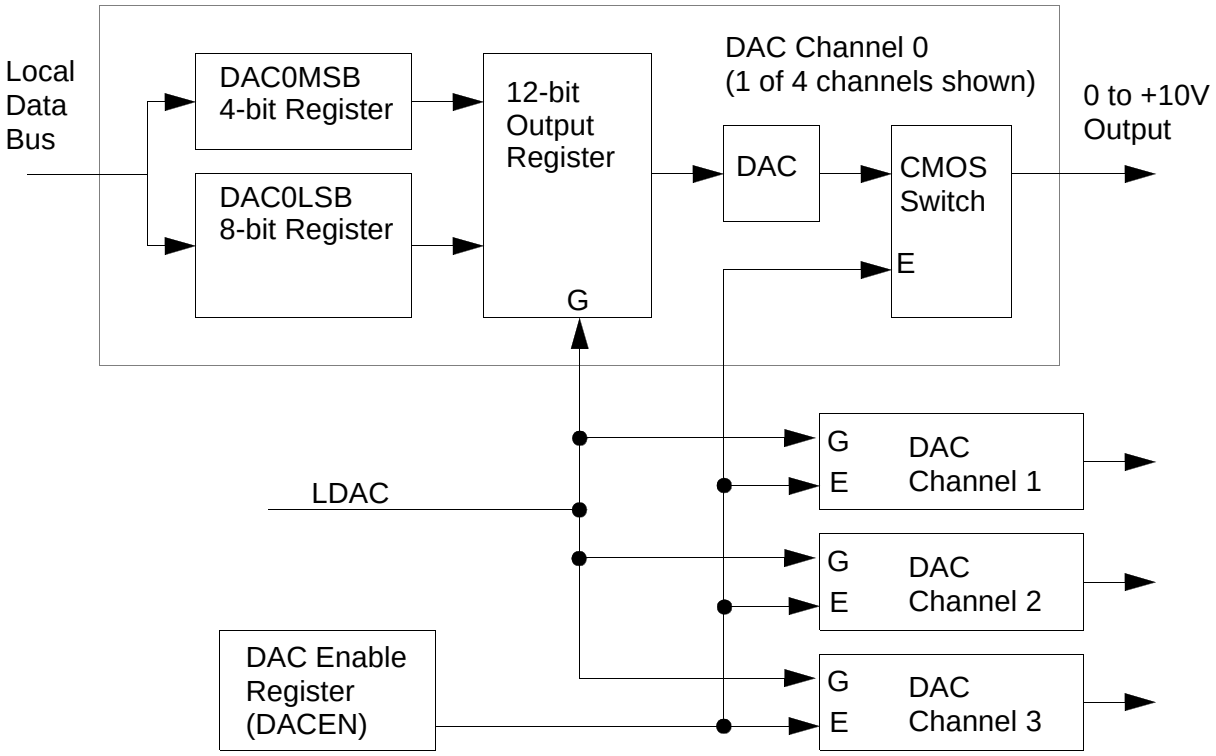
Each DAC channel consists of a low-byte/high-nibble bus register pair, 12-bit output register, 12-bit D/A converter, and CMOS switch.

All four CMOS switches are enabled by the one-bit DAC Enable Register. When enabled, the CMOS switches connect all four DAC outputs to 40-pin header P2. When disabled, all DAC output signals at P2 are pulled down to zero volts.

A control strobe — LDAC — simultaneously transfers data from the four bus register pairs to their corresponding 12-bit output registers. DAC output ranges are fixed at 0 to +10 volts.

.

D/A Subsystem Block Diagram



DAC Enable Register

Following a reset, the four 12-bit DAC output registers contain indeterminate values. To ensure orderly startup, the DAC Enable Register (DACEN) turns off all CMOS switches to prevent random DAC voltages from reaching the analog I/O connector.

DACEN — which is automatically cleared by a reset — may be manipulated by the ISAbus master. The DACEN register is accessed through the write-only CHCTRL port.

CHCTRL (12):

0	0	0	0	1	0	D	W

The D bit enables DAC outputs when set to 1 and disables DAC outputs when set to 0.

Note: the W bit, which enables and disables the watchdog timer, is also controlled by this port. Make sure you don't inadvertently change the watchdog enable bit when enabling or disabling the DAC outputs.

DAC Data Ports

Each DAC is allocated two output ports —DACxLSB and DACxMSB (where x is the DAC identifier 0, 1, 2, or 3) — called bus registers. All DAC setpoint data are written into these registers.

DACxLSB is the D/A converter least-significant data byte register, and DACxMSB is the most-significant data nibble register. The data nibble is right-justified in the DACxMSB registers.

Data may be written to the DACxLSB or DACxMSB registers in any order.

Note: DAC outputs do not change when the DACxLSB and DACxMSB registers are written to. DAC outputs change only when the LDAC port is read (see next section).

.

Base Address Offset	Function
0	DAC0LSB
1	DAC0MSB
2	DAC1LSB
3	DAC1MSB
4	DAC2LSB
5	DAC2MSB
6	DAC3LSB
7	DAC3MSB

LDAC Port

DAC outputs change when data is transferred from the bus registers into the corresponding DAC output registers.

The LDAC port is used to transfer data from bus registers to output registers. Reading from the LDAC port transfers data to all four DAC output registers simultaneously.

LDAC (0):

x	x	x	x	x	x	x	x

The data value returned from the LDAC port is indeterminate and has no meaning.

Transfer Function

The relationship between DAC input code and output voltage is shown in the following table. Input coding for DAC channels is unsigned binary.:

Data Value (hexadecimal)	Output Level (volts)
000	0.0000
001	0.0024
7FF	4.9976
800	5.0000
801	5.0024
FFF	9.9976

DAC Initialization

Before setting the DAC Enable Register to enable DAC outputs, the ISAbus master should first zero all DAC output registers. Execute this event sequence to achieve orderly startup following a reset:

— Write zeros to all DACxLSB and DACxMSB registers.

— Read from the LDAC port to change all DAC outputs to 0 volts.

— Enable DAC outputs by setting the CHCTRL port D bit.

Channel Differences

DAC channel 0 is identical to the other channels except for the addition of a remote sense function.

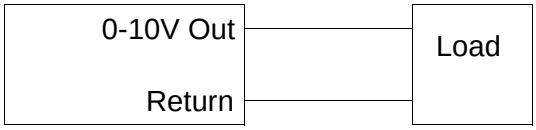
Each channel has an output source impedance specified at 85 ohms, maximum. Because channel 0 senses the DAC output after its CMOS switch, however, its effective source impedance is zero for output currents up to the specified maximum.

Because of their 85 ohm source impedances, channels 1, 2, and 3 will tend to exhibit “gain error” as a function of load current. This is not a problem as long as the load current is constant, and hence, DAC output voltage is monotonic.

Use DAC channel 0 if load impedance varies significantly or absolute accuracy is important for your application. Alternately, any of the channel 1, 2, or 3 outputs may be buffered by external amplifiers.

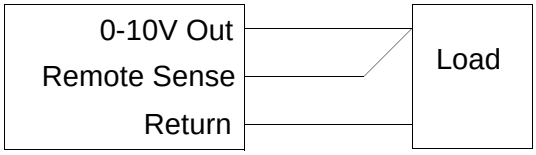
Connections: Channel 1, 2, and 3

Each DAC channel makes two connections to the 421: 0 to +10V output, and output return.



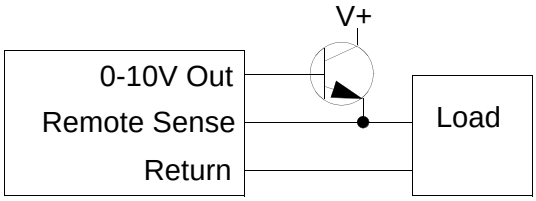
Connections: Channel 0

In addition to output and return, channel 0 has a connection for remote voltage sensing. The remote sense signal must be connected to the output signal. Connect the output and sense signals near to the load for best accuracy.



Connections: Channel 0 with External Buffer

In applications that use an external buffer for DAC channel 0, it is best to sense the voltage at the buffer output instead of the DAC output.



Connector Pinout

All external circuitry connects to the DAC channels through 40-pin header P2. Connector P2 is shared by the D/A and A/D converter circuits.

The table below defines that portion of connector P2 used for the DAC channels.

Pin on Conn P2	Function
1	Channel 0 output
19	Channel 0 sense
3	Channel 0 return
5	Channel 1 output
7	Channel 1 return
9	Channel 2 output
11	Channel 2 return
13	Channel 3 output
15	Channel 3 return

A/D Interface

The 421 digitizer section consists of an eight-input differential CMOS analog multiplexer, resistor programmed instrumentation amplifier, and high speed 12-bit analog-to-digital converter.

A/D conversions are accomplished by executing the following event sequence:

— Select analog input channel to be digitized.

— Start an A/D conversion.

— Wait for conversion to finish.

— Read conversion result.

Selecting an Input Channel

Analog input channels are selected by writing the desired channel number to the CHCTRL port. Valid channel numbers range from 0 to 7, inclusive.

CHCTRL (12):

0	0	0	0	0	N_2	N_1	N_0

Channel numbers may be written to CHCTRL at any time, but in general should not be done during an A/D conversion.

Sufficient settling time must be allowed between writing a new value to CHCTRL and starting a conversion. 9 microseconds settling time is adequate for low impedance analog sources. If you are digitizing high impedance sensors or using high gain ratios, longer settling delays may be required to meet specified performance.

It is not necessary to repeatedly write to CHCTRL if your application dwells on one analog input channel. No settling time is required in such cases.

Starting a Conversion

A/D conversions are initiated by writing to the ADSTART port. The data value written to this port is ignored, but should be all zeros for compatibility with future product enhancements.

ADSTART (13):

0	0	0	0	0	0	0	0

Writing to this port starts a conversion and sets the BZ flag (A/D busy flag) in the status register. When the conversion ends — approximately 9 microseconds after starting the conversion — the BZ flag is reset to zero and the binary result may be accessed.

Reading the Conversion Result

Binary A/D conversion results may be read from ports ADLSB and ADMSB. Port ADLSB contains the least-significant data byte. Port ADMSB contains the most-significant nibble, right justified with leading zeros.

ADLSB (12):

d_7	d_6	d_5	d_4	d_3	d_2	d_1	d_0

ADMSB (13):

0	0	0	0	d_{11}	d_{10}	d_9	d_8

Data from port ADLSB must be read first, followed by data from ADMSB. These I/O ports must be read once and only once at the end of each A/D conversion. Digitized data may be read any time after a conversion is finished.

A short delay is required between ISAbus host reads from the ADLSB and ADMSB ports. There is no maximum delay between reading ADLSB and ADMSB ports. See the A/D timing diagram for details.

Configuring Unipolar/Bipolar Operation

All analog input channels may be collectively programmed for either unipolar or bipolar operation. DIP switch S2-1 is used to configure the operating mode:

S2-1	Mode
On	Unipolar
Off	Bipolar

Install a shunt at UNI to configure all channels for unipolar operation. Remove the shunt from UNI to configure all channels for bipolar operation.

Data Format

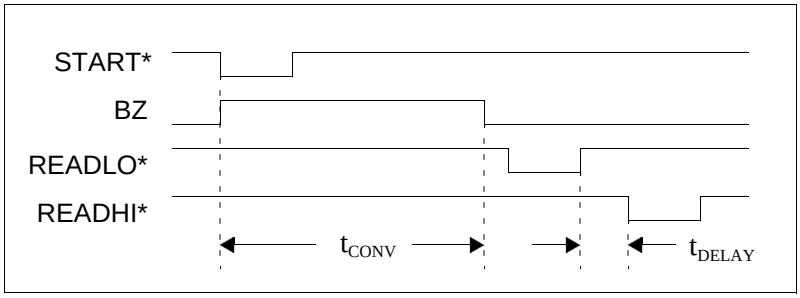
A/D output data format is a function of the digitizer operating mode.

In the unipolar mode, output data is formatted as 12-bit unsigned binary integers with leading zeros in the most-significant nibble.

In the bipolar mode, output data is formatted as 12-bit signed (two's complement) integers with leading zeros in the most-significant nibble.

A/D Timing Diagram

Timing Diagram Signals	
Name	Description
START*	A/D start strobe. Coincident with host CPU write strobe to port ADSTART.
BZ	"A/D busy" flag -- BZ bit as read by host from STATUS port.
READLO*	A/D LSB read strobe. Coincident with host read strobe from port ADLSB.
READHI*	A/D MSB read strobe. Coincident with host read strobe from port ADMSB.



Parameter	Symbol	Minimum	Maximum
Conversion time	t_{CONV}		10 us
READLO to READHI delay	t_{DELAY}	1100 ns	

Gain Programming

A/D front-end gain may be set to any desired value greater than or equal to unity via selection of a gain programming resistor. Note that all analog input channels have the same gain. Changing the front-end gain affects all channels. As shipped from the factory, the front-end gain is set to unity.

The front-end gain may be programmed by installing a resistor at position R1. For any arbitrary gain R1 can be calculated using the formula

$$R1 = \frac{49400}{Gain - 1}$$

Note that for Gain = 1, R1 must be an open circuit (default as shipped from the factory). To minimize gain drift, R1 should have a low temperature coefficient — 50 ppm/C or better for the best performance.

The table below shows required values of R1 for various gains using standard 1% resistor values. The listed gain values are nominal and may be trimmed to exact values as described in the next section.

R1 (ohms)	Gain	Input Range	
		Unipolar	Bipolar
Open	1	0 to +10V	-5 to +5V
49.9K	2	0 to +5V	-2.5 to +2.5V
12.4K	5	0 to +2V	-1V to +1V
5.49K	10	0 to +1V	-500 to +500mV
2.61K	20	0 to +500mV	-250 to +250mV
1.00K	50	0 to +200mV	-100 to +100mV
499	100	0 to +100mV	-50 to +50mV
249	200	0 to +50mV	-25 to +25mV
100	500	0 to +20mV	-10 to +10mV
49.9	1000	0 to +10mV	-5 to +5mV

Gain Trim

Since it is unlikely that the gain setting resistor R1 will produce the exact gain value desired, a gain trim adjustment is provided on the 421

A/D front-end gain may be trimmed by adjusting potentiometer R2. The trimmer provides an adjustment range of approximately plus and minus seven percent.

Offset Trim

A/D input offset may be trimmed by adjusting potentiometer R3. This trimmer provides an adjustment range of approximately +/- 6 millivolts.

This adjustment may alter circuit gain. Consequently, offset should be trimmed before adjusting gain.

Unipolar-mode Transfer Function

As discussed earlier, A/D output codes are represented as 12-bit unsigned integers in the unipolar mode. Output values range from 0 at zero volts input to 4095 at the positive full-scale input voltage.

The following table shows the relationship between input voltage and A/D output codes. A 0 to +10 volt gain range (unity gain) is assumed for this illustration.

Input Voltage	Output Code
0.000V	000
2.441mV	001
4.998V	7FF
5.000V	800
5.002V	801
+9.997V	FFF

Bipolar-mode Transfer Function

A/D output codes are formatted as 12-bit signed two's complement integers in the bipolar mode. Output values range from -2048 at negative full-scale input to +2047 at positive full-scale input voltage.

The following table shows the relationship between input voltage and A/D output codes. A -5 to +5 volt gain range (unity gain) is assumed for this illustration.

Input Voltage	Output Code
-5.000V	800
-4.997V	801
-2.441mV	FFF
0.000V	000
+2.441mV	001
+4.997V	7FF

Input Connections

Each analog input channel makes two connections to the 421 board. These two connections constitute a differential input pair. The digitizer measures the difference in voltage between the two inputs.

Note that the input common-mode voltage — the voltage at either input relative to ISAbus five volt return (GND) — should not exceed plus or minus 10 volts. Minor excursions beyond this limit will cause inaccurate measurements. Significant excursions may result in damage to digitizer circuitry. Refer to the 421 specifications section for further details.

All analog input sources should be referenced to ISAbus GND. If you are connecting an isolated source (a source not referenced to GND), you should tie one side of the source to GND. Since analog input channels are true differential, it doesn't matter which side is connected to GND.

Connector Pinout

All differential analog inputs connect to 40-pin header connector P2. This connector is shared by 421 analog input and analog output functions. The table below describes that portion of P2 related to analog input functions. Note that the GND signals are related to both analog input and output functions.

Connector P2: Analog I/O

Pin	Name	Function
2	ADC0+	A/D channel 0 positive input
4	ADC0-	A/D channel 0 negative input
6	ADC1+	A/D channel 1 positive input
8	ADC1-	A/D channel 1 negative input
10	ADC2+	A/D channel 2 positive input
12	ADC2-	A/D channel 2 negative input
14	ADC3+	A/D channel 3 positive input
16	ADC3-	A/D channel 3 negative input
18	ADC4+	A/D channel 4 positive input
20	ADC4-	A/D channel 4 negative input
22	ADC5+	A/D channel 5 positive input
24	ADC5-	A/D channel 5 negative input
26	ADC6+	A/D channel 6 positive input
28	ADC6-	A/D channel 6 negative input
30	ADC7+	A/D channel 7 positive input
32	ADC7-	A/D channel 7 negative input
33,34, 38,39, 40	GND	Analog common

Appendix A: Specifications

Bus Interface	Type	ISA, I/O slave, 16-bit address, 8-bit data	
	Address requirements	32-byte block starting on any 32-byte boundary	
Watchdog Timer	Timeout	630 milliseconds, min.	770 milliseconds, max.
Encoder Interface	Type	Quadrature encoded, single-ended	
	Input characteristics	RS422 differential, single-ended TTL/CMOS compatible	
	Pulse rate	0 Hz, min.	375 KHz, max.
	5V current, per channel		100 mA, max.
Relay Interface	Input characteristics	TTL/CMOS compatible, 10Kohm pullup to +5V	
	Output sink current	60 mA, min.	
A/D Interface	A/D type	12-bits, successive approximation	
	Differential Nonlinearity		+/- 1 LSB, max.
	Integral Nonlinearity		+/- 1 LSB, max.
	Settling time (G = 1)		9 microseconds, max.
	Conversion time		10 microseconds, max.
	CMRR (G = 1)	73 dB min.	90 dB typical
	Analog input	-25 Volts, absolute min.	+25 Volts, absolute max.
	Offset error		+/- 1 LSB, max.
	Input impedance	10 megohms, min.	
D/A Interface	D/A type	12 bits, voltage output, output range fixed at 0 to +10V	
	Gain error		+/- 3 LSB, max.
	Output offset		+/- 7.5 millivolts, max.
	Output Z (chan 1, 2 & 3)		85 ohms, max.
Power	Operating range	+4.75 Volts, min.	+5.25 Volts, max.
	Quiescent current	200 mA, typical	

Appendix B: I/O Map Summary

Adrs	Write Function	Read Function
0	**DAC0LSB** d_7 d_6 d_5 d_4 d_3 d_2 d_1 d_0 Loads DAC 0 LSB register without updating the DAC output.	**LDAC** x x x x x x x x Transfers all DACxMSB and DACxLSB data to the DAC output registers, updating all DAC outputs simultaneously. The returned data value is random and has no significant meaning.
1	**DAC0MSB** x x x x d_{11} d_{10} d_9 d_8 Loads DAC 0 MSB register without updating the DAC output.	
2	**DAC1LSB** d_7 d_6 d_5 d_4 d_3 d_2 d_1 d_0 Loads DAC 1 LSB register without updating the DAC output.	
3	**DAC1MSB** x x x x d_{11} d_{10} d_9 d_8 Loads DAC 1 MSB register without updating the DAC output.	
4	**DAC2LSB** d_7 d_6 d_5 d_4 d_3 d_2 d_1 d_0 Loads DAC 2 LSB register without updating the DAC output.	
5	**DAC2MSB** x x x x d_{11} d_{10} d_9 d_8 Loads DAC 2 MSB register without updating the DAC output.	
6	**DAC3LSB** d_7 d_6 d_5 d_4 d_3 d_2 d_1 d_0 Loads DAC 3 LSB register without updating the DAC output.	
7	**DAC3MSB** x x x x d_{11} d_{10} d_9 d_8 Loads DAC 3 MSB register without updating the DAC output.	
8	**RELAY0** R_7 R_6 R_5 R_4 R_3 R_2 R_1 R_0 Sets relay channels 0-7 simultaneously. All bits are active high.	**RELAY0** R_7 R_6 R_5 R_4 R_3 R_2 R_1 R_0 Returns the states of relay channels 0-7. All bits are active high.
9	**RELAY1** R_{15} R_{14} R_{13} R_{12} R_{11} R_{10} R_9 R_8 Sets relay channels 8-15 simultaneously. All bits are active high.	**RELAY1** R_{15} R_{14} R_{13} R_{12} R_{11} R_{10} R_9 R_8 Returns the states of relay channels 8-15. All bits are active high.
A	**RELAY2** R_{23} R_{22} R_{21} R_{20} R_{19} R_{18} R_{17} R_{16} Sets relay channels 16-23 simultaneously. All bits are active high.	**RELAY2** R_{23} R_{22} R_{21} R_{20} R_{19} R_{18} R_{17} R_{16} Returns the states of relay channels 16-23. All bits are active high.
B	**RESET** 0 0 0 0 0 0 0 0 Force a local reset on the 421, returning it to its default power-up state. All encoder counters are reset, and the watchdog timer and DAC outputs are disabled. Relay channels are not affected.	**STATUS** x x x x UN DE BZ FT Returns 421 status. All status bits are active high. Status bits have the following meanings: UN=A/D unipolar mode, DE=DAC channels enabled, BZ=A/D subsystem busy, FT=421 fault.
C	**CHCTRL** 0 0 0 0 M N_2 N_1 N_0 Select A/D channel or set hardware enables. M specifies operation: 0=select channel $N_2N_1N_0$, 1=set enables: N_0 enables watchdog timer and N_1 enables DAC outputs.	**ADLSB** d_7 d_6 d_5 d_4 d_3 d_2 d_1 d_0 Returns the least-significant data byte (lsb) from the last A/D conversion. The lsb must be read before the msb to ensure proper operation.
D	**ADSTART** 0 0 0 0 0 0 0 0 Start an A/D conversion.	**ADMSB** 0 0 0 0 d_{11} d_{10} d_9 d_8 Returns the most-significant data nibble from the last A/D conversion.
E	**CNTCTL** 0 0 0 0 M_1 M_0 S_1 S_0 M_1M_0 specifies counter operation: 00=latch counter S_1S_0, 01=reset counter S_1S_0, 1X=set mode $M_0S_1S_0$. When S_1S_0 selects counter channel: 00=chan0, 01=chan1, 10=chan2, 11=not valid.	**CNTLSB** c_7 c_6 c_5 c_4 c_3 c_2 c_1 c_0 Returns the least-significant byte from the encoder counter latch register.
F	**HITDOG** 0 0 0 0 0 0 0 0 Restart the watchdog timer.	**CNTMSB** c_{15} c_{14} c_{13} c_{12} c_{11} c_{10} c_9 c_8 Returns the most-significant byte from the encoder counter latch register.

Adrs	Write Function	Read Function
0	**RELAY4** R_7 R_6 R_5 R_4 R_3 R_2 R_1 R_0 Sets relay channels 0-7 simultaneously. All bits are active high.	**RELAY4** R_7 R_6 R_5 R_4 R_3 R_2 R_1 R_0 Returns the states of relay channels 0-7. All bits are active high.
1	**RELAY5** R_{15} R_{14} R_{13} R_{12} R_{11} R_{10} R_9 R_8 Sets relay channels 8-15 simultaneously. All bits are active high.	**RELAY5** R_{15} R_{14} R_{13} R_{12} R_{11} R_{10} R_9 R_8 Returns the states of relay channels 8-15. All bits are active high.
2	**RELAY6** R_{23} R_{22} R_{21} R_{20} R_{19} R_{18} R_{17} R_{16} Sets relay channels 16-23 simultaneously. All bits are active high.	**RELAY6** R_{23} R_{22} R_{21} R_{20} R_{19} R_{18} R_{17} R_{16} Returns the states of relay channels 16-23. All bits are active high.
3	Reserved for future use	
4		
5		
6		
7		
8		
9		
A		
B		
C		
D		
E		
F		

Appendix C: Connector Pinouts

Connector P5 and P6: Digital I/O

Pin	Name	Function
1	CH23	I/O Channel 23
3	CH22	I/O Channel 22
5	CH21	I/O Channel 21
7	CH20	I/O Channel 20
9	CH19	I/O Channel 19
11	CH18	I/O Channel 18
13	CH17	I/O Channel 17
15	CH16	I/O Channel 16
17	CH15	I/O Channel 15
19	CH14	I/O Channel 14
21	CH13	I/O Channel 13
23	CH12	I/O Channel 12
25	CH11	I/O Channel 11
27	CH10	I/O Channel 10
29	CH9	I/O Channel 9
31	CH8	I/O Channel 8
33	CH7	I/O Channel 7
35	CH6	I/O Channel 6
37	CH5	I/O Channel 5
39	CH4	I/O Channel 4
41	CH3	I/O Channel 3
43	CH2	I/O Channel 2
45	CH1	I/O Channel 1
47	CH0	I/O Channel 0
49	+5V	+5V power
Even	GND	5V return

Connector P3: Encoder Inputs

Pin	Name	Function
1	+5V	Channel 0 +5V power
2	$+A_0$	Channel 0 phase A input (+)
3	GND	Channel 0 5V return
4	$+B_0$	Channel 0 phase B input (+)
5	+5V	Channel 1 +5V power
6	$+A_1$	Channel 1 phase A input (+)
7	GND	Channel 1 5V return
8	$+B_1$	Channel 1 phase B input (+)
9	+5V	Channel 2 +5V power
10	$+A_2$	Channel 2 phase A input (+)
11	GND	Channel 2 5V return
12	$+B_2$	Channel 2 phase B input (+)
13		-- no connect --
14		-- no connect --
15	$-A_0$	Channel 0 phase A input (-)
16	$-A_1$	Channel 1 phase A input (-)
17	$-A_2$	Channel 2 phase A input (-)
18	$-B_0$	Channel 0 phase B input (-)
19	$-B_1$	Channel 1 phase B input (-)
20	$-B_2$	Channel 2 phase B input (-)

Connector P2: Analog I/O

Pin	Name	Function
1	DAC0+	DAC channel 0 output
2	ADC0+	A/D channel 0 positive input
3	DAC0-	DAC channel 0 return
4	ADC0-	A/D channel 0 negative input
5	DAC1+	DAC channel 1 output
6	ADC1+	A/D channel 1 positive input
7	DAC1-	DAC channel 1 return
8	ADC1-	A/D channel 1 negative input
9	DAC2+	DAC channel 2 output
10	ADC2+	A/D channel 2 positive input
11	DAC2-	DAC channel 2 return
12	ADC2-	A/D channel 2 negative input
13	DAC3+	DAC channel 3 output
14	ADC3+	A/D channel 3 positive input
15	DAC3-	DAC channel 3 return
16	ADC3-	A/D channel 3 input
17	—	no connect
18	ADC4+	A/D channel 4 positive input
19	DAS0+	DAC channel 0 output sense
20	ADC4-	A/D channel 4 negative input
21	—	no connect
22	ADC5+	A/D channel 5 positive input
23	—	no connect
24	ADC5-	A/D channel 5 negative input
25	—	no connect
26	ADC6+	A/D channel 6 positive input
27	—	no connect
28	ADC6-	A/D channel 6 negative input
29	—	no connect
30	ADC7+	A/D channel 7 positive input
31	—	no connect
32	ADC7-	A/D channel 7 negative input
33	GND	Analog common
34	GND	Analog common
35	—	no connect
36	+15V	+15 volts
37	—	no connect
38	GND	Analog common
39	GND	Analog common
40	GND	Analog common

Connector P1: Watchdog Timer Output

Pin	Name	Function
1	-Reset	Output to ISAbus reset conn.
2	GND	ISAbus 5V return

Appendix D: Programming

Sample QuickBasic code segments are listed below to illustrate programming interface techniques for the 421 board. If you are coding your application in a language other than QuickBasic, you may find these listings useful as a framework for writing your own drivers.

Although these listings are intended principally as programming examples, they may be used as is (or with minor modification) as a basic set of interface drivers for the 421.

Feel free to modify, plagiarize and adapt these listings to suit your application requirements.

Module Level Code

```
'*************************************************************************
' SAMPLE QUICKBASIC INTERFACE CODE FOR THE 421
'*************************************************************************

CONST BASEPORT = &H3A0                  '421 I/O base address

' Write-only I/O ports:
CONST DAC0LSB = BASEPORT + 0            'DAC data registers
CONST RESETPORT = BASEPORT + 11         'board reset
CONST CHCTRL = BASEPORT + 12            'ADC channel reg / watchdog, DAC enables
CONST ADSTART = BASEPORT + 13           'start A/D
CONST CNTCTL = BASEPORT + 14            'encoder control port
CONST HITDOG = BASEPORT + 15            'watchdog refresh

' Read-only I/O ports:
CONST LDAC = BASEPORT + 0               'DAC update port
CONST STATUSPORT = BASEPORT + 11        'board status
CONST ADLSB = BASEPORT + 12             'A/D least-significant byte
CONST ADMSB = BASEPORT + 13             'A/D most-significant nibble
CONST CNTLSB = BASEPORT + 14            'encoder least-significant byte
CONST CNTMSB = BASEPORT + 15            'encoder most-significant byte

' Read/Write I/O ports:
CONST RELAY0 = BASEPORT + 8             'relay channel low bank access ports
CONST RELAY2 = BASEPORT + 16            'relay channel high bank access ports

' CHCTRL port bit constants:
CONST DOGBIT = 1                        'watchdog enable/disable control bit
CONST DACBIT = 2                        'DAC output enable/disable control bit

' Boolean constants:
CONST FALSE = 0
CONST TRUE = NOT FALSE

' Variables that are visible throughout this module:
DIM SHARED relay%(0 To 5)               'image of relay channel output registers
DIM SHARED chctrlImage%                 'image of CHCTRL output port
```

D/A Interface Procedures

```
SUB DacEnable (Value%)
'****************************************************************************
' Collectively enable or disable all DAC outputs.
'
' Imports:
'     Value% (boolean): TRUE enables outputs, FALSE disables outputs.
''****************************************************************************

      SELECT CASE Value%
            CASE TRUE: chctrlImage% = chctrlImage% OR DACBIT
            CASE FALSE: chctrlImage% = chctrlImage% AND (DACBIT XOR -1)
      END SELECT

      OUT CHCTRL, 8 OR chctrlImage%

END SUB



SUB WriteDAC (DACid%, Value%)
'****************************************************************************
' Write data value to DAC output register and refresh DAC output.
'
' Imports:
'     DACid% = DAC channel in range 0:3.
'     Value% = data value to be written to DAC in range 0:4095.
'****************************************************************************

      ' Copy data value to DAC input register
      OUT DAC0LSB + DACid% * 2, Value% AND 255
      OUT DAC0LSB + DACid% * 2 + 1, Value% \ 256

      ' Update DAC outputs
      junk% = INP(LDAC)

END SUB
```

A/D Interface Procedure

```
FUNCTION Digitize% (Chan%)
'*****************************************************************************
' Digitize one 421 analog input channel.
'
' Imports:
'     Chan% = A/D input channel in the range 0:7.
'
' Exports:
'     Digitize% = digitized value in the range 0:4095.
'*****************************************************************************

    CONST BZ = 2                                    ' Status register mask: "A/D busy" flag

    OUT CHCTRL, Chan% AND 7                         ' Select analog input channel

' NOTE: you may need to insert a settling time delay here, depending on your programming
' language and processor speed.

    OUT ADSTART, 0                                  ' Digitize the input

    DO: LOOP WHILE INP(STATUSPORT) AND BZ           ' Wait for A/D to finish

    loResult% = INP(ADLSB)                          ' Read the digitized result
    hiResult% = INP(ADMSB)

    Digitize% = loResult% + 256 * (hiResult% AND &HF)   ' Package result for function return

END FUNCTION
```

Board Reset Procedure

```
SUB reset421 ()
'*****************************************************************************
' Invoke 421 soft reset.
'*****************************************************************************

    OUT RESETPORT, 0            ' Reset 421 board

    chctrlImage% = 0            ' Initialize CHCTRL output port image

    For i% = 0 To 2             ' Initialize relay images
        relay%(i%) = 0
    Next i%

END SUB
```

Relay Interface Procedures

```
FUNCTION ReadRelay% (bankAdrs%, chan%)
'*************************************************************************
' Read relay channel input state.
'
' Imports:
'     chan% = relay channel in range 0:23.
'     bankAdrs% = address of relay group (RELAY0 or RELAY2).
'
' Exports:
'     ReadRelay% = relay channel input state (TRUE or FALSE).
'*************************************************************************

    adrs% = bankAdrs% + chan% \ 8                   ' Compute relay channel port address

    bitMask% = 2 ^ (chan% AND 7)                    ' Compute relay channel bit mask

    readRelay% = ((INP(adrs%) AND bitMask%) <> 0)   ' Read relay input state

END FUNCTION



SUB writeRelay (bankAdrs%, chan%, state%)
'*************************************************************************
' Program relay output.
'
' Imports:
'     chan% = relay channel in range 0:23.
'     bankAdrs% = address of relay group (RELAY0 or RELAY2).
'     state% (boolean): TRUE = relay on, FALSE = relay off.
'*************************************************************************

    Index% = chan% \ 8                      ' Compute relay%() index and port address

    bitMask% = 2 ^ (chan% AND 7)            ' Compute relay channel bit set mask
    invMask% = bitMask% XOR 255             ' Compute relay channel bit reset mask

    ' Update relay port output image
    relay%(Index%) = (relay%(Index%) AND invMask%) OR (state% AND bitMask%)

    OUT bankAdrs% + Index%, relay%(Index%)' Write new state to relay channel output register

END SUB
```

Encoder Interface Procedures

```
FUNCTION ReadEncoder& (EncoderID%)
'*************************************************************************
' Read and return encoder position.
'
' Imports:
'     EncoderID% = encoder channel number in range 0:2.
'
' Exports:
'     ReadEncoder& = encoder position in range 0:65535.
'*************************************************************************

     ' Transfer encoder counter to holding register
     OUT CNTCTL, EncoderID%

     ' Read holding register as 2 bytes & concatenate into long
     counts& = INP(CNTMSB)
     counts& = counts& * 256 + INP(CNTLSB)

     ReadEncoder& = counts&

END FUNCTION



SUB ResetEncoder (EncoderID%)
'*************************************************************************
' Reset specified encoder counter to zero.
'
' Imports:
'     EncoderID% = encoder channel in range 0:2.
'*************************************************************************

     OUT CNTCTL, 4 Or EncoderID%          ' Reset encoder counter to zero

END SUB
```

Watchdog Timer Procedures

```
SUB WatchdogEnable (Value%)
'*************************************************************************
' Enable or disable watchdog timer.
'
' Imports:
'     Value% (boolean): TRUE = enable timer, FALSE = disable timer.
'*************************************************************************

    SELECT CASE Value%
        CASE TRUE: chctrlImage% = chctrlImage% OR DOGBIT
        CASE FALSE: chctrlImage% = chctrlImage% AND (DOGBIT XOR -1)
    END SELECT

    OUT CHCTRL, 8 OR chctrlImage%

END SUB




SUB WatchdogBump ()
'*************************************************************************
' Refresh watchdog timer.
'*************************************************************************

    OUT HITDOG, 0

END SUB
```