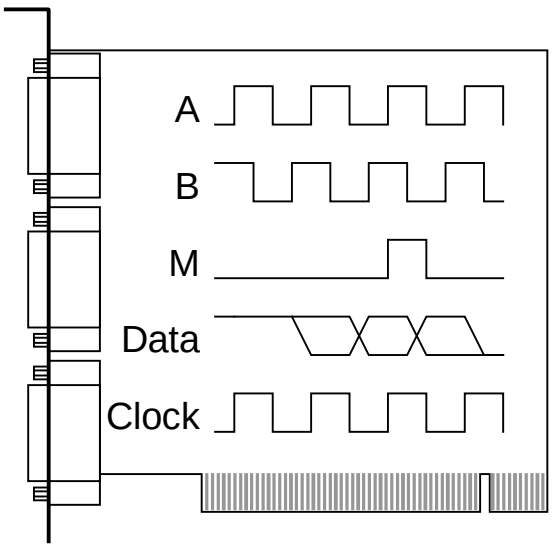
DEVA001 PCI/PCIE encoder interface card

User's Manual



52 Woodside Business Park
Birkenhead
Wirral CH41 1EL
United Kingdom

Tel +44 (0)151 647 3222
Fax +44 (0)151 647 4511

Email: support@deva.co.uk

Web: www.deva.co.uk

1 Overview

1.1 Product Features

The DEVA001 range of 3 and 4 axis incremental and absolute encoder interface cards have been designed to enable simultaneous reading of 3 or 4 encoders using a PC based system. They can be used for a wide range of measurement and motion control applications, such as CNC machines, robotics and co-ordinate measuring machines.

The range includes support for PCIe, PCI and ISA bus interfaces. All three types share many common features; however the PCI / PCIe variants have several additional facilities.

1.1.1 Common Incremental interface features

- Three encoder inputs for differential or single - ended input
- 32 bit counters for each encoder channel
- Marker input (freeze / capture) for each encoder channel
- Zero input for each encoder channel
- +5v and +12v Power supplies available on encoder connectors
- Digital filters on all channels
- Timer / Event driven interrupt logic
- Optional Renishaw probe input

1.1.2 Common Absolute interface features

- Three encoder inputs
- 32 bit registers for each encoder channel
- Programmable data and scan rates
- Programmable bit length and Gray to binary conversion
- Power fail and parity detection
- +5v and +12v power supplies available on encoder connectors
- Timer / Event driven interrupt logic

1.1.3 PCI specific features

PCI interface cards from issue 4.2 onwards have the following additional set of facilities.

- 16+32 TTL level digital IO
- Flexible event system
- Axis Compare
- Pulse generator
- Time stamper
- Data acquisition buffer
- -12v Supply

1.1.4 PCIe specific features

PCIe interface cards from issue 5.0 onwards have the following additional set of facilities.

- 4th encoder channel
- 16+32 TTL level digital IO
- Flexible event system
- Axis Compare
- Pulse generator
- Time stamper
- Data acquisition buffer

1.2 Typical applications

A few examples are given to illustrate how the DEVA001 may be effectively used within a target application.

1.2.1 Co-ordinate measuring machine

In this application the linear incremental encoders from the X, Y and Z axes are connected to the X, Y and Z channels of the DEVA001. Each encoder has a marker or index signal which is connected to the M input. This will allow a reference cycle to be performed when the equipment is switched on. A measurement probe is

connected to the Renishaw touch probe input. This will allow the X, Y and Z axes positions to be captured when the probe is deflected.

1.2.2 Synchronised measurement of encoders

The PCI / PCIe variants of the DEVA001 are able to make measurements of encoder position synchronised in hardware to several different events including: timer, pulse generator, axis compare and digital input. For this example the pulse generator is programmed to generate an event every 10 counts of the master encoder. Every time an event is generated the card is programmed to capture all encoder positions. Captured data is placed in a user configurable FIFO buffer and may be retrieved by user software at any time.

1.2.3 Telescope directional feedback

To measure the direction of a telescope on a turntable a single turn absolute SSI encoder may be employed at the centre of rotation. If greater resolution is required a multi-turn encoder may be employed at the circumference of the table. An absolute SSI encoder has the advantage that the telescope position is always available and hence the telescope does not need to be rotated through 360° whenever the power is re-applied.

1.3 Support software

A variety of software drivers and libraries are provided with the interface card to enable software development to be performed within a number of operating systems and applications. Software support is an ongoing activity, if support for a particular application or operating system is not currently provided, please call the Deva office to determine its availability.

1.3.1 Windows 7/Vista/XP 32/64

Both PCIe and PCI cards are supported in Windows 7/Vista/XP 32/64 by a Windows driver model driver (WDM). The driver supports a standard programming interface. Please refer to section 3 'Device Driver Usage' for more details.

1.3.2 National Instruments LabVIEW

LabVIEW support is provided via an LLB of VI's which allow the access to the device driver functions. These VI's allow users to develop powerful applications very quickly using the LabVIEW system.

1.3.3 Digital Readout

A powerful digital readout is supplied which is capable of exercising all the common facilities of the DEVA001. This is useful to allow users to quickly verify that the DEVA001 is installed correctly and to make checks of their system without having to write their own software.

The DRO will handle up to four cards (12 axes) and displays both absolute and incremental positions. The DRO also displays the Marker register, Probe register and allows control of the Probe and Footswitch options.

To allow the user to read meaningful values each axis of the DRO may be independently scaled to display real units.

For the incremental card all facilities such as Marker and Probe may be toggled on and off. If a Probe is activated the probed values are shown in the Probe register, if a Marker is activated the value is shown in the Marker register.

For the absolute SSI card it is possible to configure all features via a pop-up menu.

1.4 Accessories

The DEVA001 incremental encoder interface is supplied with three high density 15 way 'D' type connectors and matching shells.

Deva can of course supply a wide range of encoders pre-wired and tested for use with the DEVA001.

2 Installation and configuration

2.1 Software support CDROM

The DEVA001 3-axis encoder interface card is supplied with a software support CDROM containing support for all DEVA001 encoder interfaces along with support and information for many of Deva's other products. The CDROM includes the following items:

- Windows 7/Vista/XP 32/64 bit device drivers, providing coherent encoder interface card hardware management.
- Windows 7/Vista/XP 32/64 bit DLL and import library, containing the driver access functions.
- 'C' language library routines and header files, which create a simple interface to device driver functions.
- Example 'C' programs, illustrating card read/write using device driver functions.
- Visual Basic 6 Module, which provides constant and function definitions to allow simple DLL access.
- Example Visual Basic 6 programs, illustrating card read/write using device driver functions.
- .Net 2.0 Assembly, which provides a complete interface to the standard DLL.
- Example C# programs, illustrating card read/write using device driver functions.
- National Instruments LabVIEW driver VI library.
- Demonstration Digital Readout Program for Windows 7/Vista/XP 32 and 64 bit.

2.2 PCI/PCIe Plug and Play cards

2.2.1 System requirements

The device driver library functions and the demonstration software require a PC computer with one spare PCI/PCIe slot and Windows 7, Vista or XP 32/64 bit operating systems. Both 32 and 64 bit operating systems are fully supported.

2.2.2 Device driver installation

2.2.2.1 MS-DOS installation

The MSDOS driver has been discontinued please contact Deva if you have a requirement for this driver.

2.2.2.2 Windows 7 / Vista / XP installation

During the first re-boot after the card has been installed windows will indicate that a new device has been found and will start the standard driver installation procedure. If this does not occur it is possible to initiate this process manually via the 'add new hardware' icon in the control panel or via the windows device manager.

Follow the instructions and when requested select 'have disk' and then browse to the directory on the installation CDROM containing the Deva001.inf file.

For example for PCIe card :

\PC interface products\Deva001\Issue5.x\Drivers\

Click ok and follow instructions to complete the installation.

The installation may be tested using the supplied DRO program which may be found on the CD in the utils\Win32 or utils\Win64 directories.

3 Device Driver Usage

3.1 Device driver functions

The supplied Windows7/Vista/XP 32/64 bit device drivers provide a simple method of accessing card functions and remove the need for direct register programming. Use of the device driver ensures that the user's application software is compatible with other Deva products and is protected from any future changes in the card hardware or register layout. The device driver determines the total number of axes and I/O available from all the cards in a system. This section describes the functions provided by the device driver whilst the compatibility chart in section 3.2 details the functions available from particular cards.

3.1.1 System information

Command & equate	Channel	Rd/Wr	Operation
0 VECTOR	Not used	Rd	Provides a vector to the device driver command handler.
1 NUM_AXES	Not used	Rd	Returns the total number of axis channels available from the installed cards.
2 NUM_TIMERS	Not used	Rd	Returns the total number of timers available from the installed cards.
3 NUM_INPUTS	Not used	Rd	Returns the total number of digital inputs available from the installed cards.
4. NUM_DACS	Not used	Rd	Returns the total number of digital to analog converters available from the installed cards.
5. NUM_OUTPUTS	Not used	Rd	Returns the total number of digital outputs available from the installed cards.
7. NUM_BOARDS	Not used	Rd	Returns the number of encoder cards present in the system.
8. CARD_TYPE	Not used	Rd	Returns the card type identifier.
9. VERSION_NUM	Not used	Rd	Returns the device driver version number multiplied by 100.

3.1.2 Channel information

Command & equate	Channel	Rd/Wr	Operation
10 CNT_16	Axis no.	Rd/Wr	Allows direct 16 bit read/write of the counter chip registers.

11 MODE Axis no. Rd/Wr Allows access to the mode registers of the incremental encoder counter chip or the SSI mode register.

INC_MODE Incremental mode number formed from a variety of bit fields.

Bit	Function
0-3	Count mode: 5=QUADx4AB, quadrature AB decode (default) 8=CNTAx2DIRB, count rising and falling edges of A, B selects count up/down direction 9=CNTADIRB, count rising edges of A, B selects count up/down direction
4	Invert quadrature signal A
5	Invert quadrature signal B
6	Invert marker signal M
7	Invert zero signal Z
8	Disable marker synchronisation with signals A & B
9	Count inhibit

SSI_MODE SSI mode number formed from a variety of bit fields.

Bit	Function
0-2 (ISA)	Frequency 0=2.50Mhz, 1=1.25Mhz, 2=625kHz, 3=313kHz, 4=156kHz, 5-7=78.1kHz
0-2 (PCI)	Frequency 0=2.78Mhz, 1=1.39Mhz, 2=694kHz, 3=347kHz, 4=174kHz, 5-7=86.8kHz
3-7	Reserved
8-11	Offset 0 to 15 bits
12-13	Extra bit, 0=None, 1=Power fail, 2=Even Parity, 3=Odd parity
14	0=Binary, 1=Gray
15	Output control, 0=Off, 1=On
16-18	Acquisition mode, 0=One Shot, 1=Timer, 2-3=Continuous, 4=Channel Digital Input 0
18-22	Reserved
23	Read Complete Interrupt Enable, 0=Off (Default), 1=On
24-29	Data length 1 to 32
30-31	Reserved

12 AXIS_SIZE Axis no. Rd Returns the number of 16 bit registers allocated to an input channel.

13 ENCODER_TYPE Axis no. Rd 0 = Incremental, 1 = SSI

14 AXIS_INPUTS Axis no. Rd Returns the status of the axis (post filter & inverter) inputs. The bit fields indicate 0 or 1 depending on the state of the axis inputs.

INC_INPUTS Incremental inputs register formed from a variety of bit fields.

Bit	Status
0	Quadrature input A
1	Quadrature input B
2	Marker input M
3	Zero input Z
4	Limit input 0
5	Limit input 1

SSI_INPUTS — SSI inputs register formed from a variety of bit fields.

Bit	Status
0	Raw SSI data input

15 AXIS_STATUS — Axis no. — Rd — Returns the axis status register. The bit fields indicate 0 or 1 depending on the status bit.

INC_STATUS — Incremental status register formed from a variety of bit fields.

Bit	Status
0	Power supply failure
1	Quadrature error

SSI_STATUS — SSI status register formed from a variety of bit fields.

Bit	Status
0	Power supply failure
1	Parity failure
2	Read complete occur

16 AXIS_OUT_EN — Axis no. — Rd — Enables the axis digital outputs. Specifying 1/0 in each bit field enables/disables the equivalent digital output.

INC_OUT_EN — N/A

SSI_OUT_EN — SSI axis digital output enable register formed from a variety of bit fields.

Bit	Status
0	SSI Read Trigger
1	SSI Read Complete

3.1.3 Marker information

Command & equate	Channel	Rd/Wr	Operation
20 MARK_16	Axis no.	Rd	Returns the value of a 16-bit counter register latched by the last marker function (or any other function the marker latch source is set to). See commands 23 and 26.
21 MARK_INPUT	Axis no.	Rd	Returns either 0 or 1 depending on the state of the marker input.
22 MARK_INT	Axis no.	Rd/Wr	Allows access to the card interrupt controller mask. Writing a value of 1/0 enables/disables an interrupt from the marker input. This function is for special applications only.

23	MARK_FUNC	Axis no.	Rd/Wr	Writing a value of 1/0 enables/disables the channel marker function.
24	MARK_INT_VECT	Axis no.	Rd/Wr	Allows access to the interrupt vector or interrupt call-back executed by a marker input interrupt.
25	MARK_INT_OCCUR	Axis no.	Rd	Returns a value of 1 every time a marker input interrupt has occurred.
26	MARK_LATCH_SEL	Axis no.	Rd/Wr	Controls the source event of the 'Marker' associated latch of each axis.

Value	Source
0	Marker Occur
1	Zero Occur
2	Pulse Generator 0 Sync Occur
3	User Event 0 Occur
4	Channel Digital Input 0 Occur
5	Axis Compare 0 Sync Occur
6	Axis Compare 1 Sync Occur
7	N/A (Axis Compare 0 Enable)

27	MARK_OUT_EN	Axis no.	Rd/Wr	Writing a value of 1/0 enables/disables the marker digital output.

3.1.4 Zero information

Command & equate		Channel	Rd/Wr	Operation
30	ZERO_INPUT	Axis no.	Rd	Returns either 0 or 1 depending on the state of the zero input.
31	ZERO_INT	Axis no.	Rd/Wr	Allows access to the card interrupt controller mask. Writing a value of 1/0 enables/disables an interrupt from the zero input. This function is for special applications only.
32	ZERO_FUNC	Axis no.	Rd/Wr	Writing a value of 1/0 enables/disables the channel zero function.
33	ZERO_INT_VECT	Axis no.	Rd/Wr	Allows access to the interrupt vector or interrupt call-back executed by a zero input interrupt.
34	ZERO_INT_OCCUR	Axis no.	Rd	Returns a value of 1 every time an encoder zero interrupt has occurred.

3.1.5 Extended axes

Command & equate		Channel	Rd/Wr	Operation
40	AXIS_32	Axis no.	Rd/Wr	Allows access to the 32-bit counter register values or to pseudo-incremental 32-bit position for absolute SSI encoders.

41	MARK_32	Axis no.	Rd	Returns the value of a 32-bit counter register latched by the last marker function (or any other function the marker latch source is set to). See commands 23 and 26.
42	VEL_INST	Axis no.	Rd	Returns the counter velocity per interrupt time.
43	VEL_FILT	Axis no.	Rd	Returns 10 times the counter velocity per interrupt period filtered over 10 samples.
44	ACCEL_INST	Axis no.	Rd	Returns the counter acceleration per interrupt time.
45	ACCEL_FILT	Axis no.	Rd	Returns 10^2 times the counter acceleration per interrupt period^2 filtered over 10 samples.
46	PROBE_32	Axis no.	Rd	Returns the value of a 32-bit counter register latched by the last probe function (or any other function the probe latch source is set to). See commands 93 and 100.
47	ABSOLUTE_32	Axis no.	Rd	Returns the 32-bit absolute position latched by the last read of an absolute SSI encoder.
48	TIMER_32	Axis no.	Rd	Returns the value of a 32-bit counter value latched by the last timer 1 function (or any other function the timer latch source is set to). See commands 65 and 64.

3.1.6 Input status

Command & equate	Channel	Rd/Wr	Operation
50 INPUT	Input no.	Rd	Returns either 0 or 1 depending on the state of the input.

3.1.7 Timer information

Command & equate	Channel	Rd/Wr	Operation
60 TIMER	Timer no.	Rd/Wr	This command allows access to the interval values of the on-board user timers. The timer intervals are programmed in units of 0.1 ms.
61 TIMER_INT	Timer no.	Rd/Wr	Writing a value of 1/0 enables/disables the user timer interrupt.
62 TIMER_INT_VECT	Timer no.	Rd/Wr	Allows access to the interrupt vector or interrupt call-back executed by the user timer interrupt.
63 TIMER_INT_OCCUR	Timer no.	Rd	Returns a value of 1 every time a user timer interrupt has occurred.
64 TIMER_LATCH_SEL	Axis no.	Rd/Wr	Controls the source event of the timer latch of each axis.

Value	Source
0	Timer 1 Sync Occur

1	N/A (Read Counter)
2	Pulse Generator 0 Sync Occur
3	User Event 0 Occur

65	TIMER1_FUNC	Axis no.	Rd/Wr	Writing a value of 1/0 enables/disables the user timer function for the specific axis. This allows for position readings latched on every timer 1 interval of the equivalent board.
66	TIMER_OUT_EN	Timer no.	Rd/Wr	Enables the timer digital outputs. Specifying 1/0 in each bit field enables/disables the equivalent digital output.

Bit	Status	Pulse Width
0	Timer Occur	Occur to serviced
1	Timer Terminal Count	1us

3.1.8 DAC Control

Command & equate		Channel	Rd/Wr	Operation
70	DAC_MV	Axis no.	Rd/Wr	Allows access to the analog output channel for each axis in a system. The value is in units of mV.
71	DAC_UV	Axis no.	Rd/Wr	Allows access to the analog output channel for each axis in a system. The value is in units of uV.

3.1.9 Output control

Command & equate		Channel	Rd/Wr	Operation
80	OUTPUT	Output no.	Rd/Wr	Allows access to the system digital outputs. The output is energised with a logical 1.
81	SERVO_ENABLE	Axis no.	Rd/Wr	Allows access to the system servo enable outputs for each axis. The output is energised with a logical 1.

3.1.10 Probe information

Command & equate		Channel	Rd/Wr	Operation
90	PROBE_16	Axis no.	Rd	Returns the value of a 16-bit counter register latched by the last probe function (or any other function the probe latch source is set to). See commands 93 and 100.
91	PROBE_INPUT	Board no.	Rd	Returns either 0 or 1 depending on the state of the probe input.

92	PROBE_INT	Board no.	Rd/Wr	Allows access to the card interrupt controller mask. Writing a value of 1/0 enables/disables the probe input interrupt. This function is for special applications only.
93	PROBE_FUNC	Board no.	Rd/Wr	Writing a value of 1/0 enables/disables the channel probe function. The probe function is level triggered by the probe input and the function reset automatically after execution of the function.
94	PROBE_INT_VECT	Board no.	Rd/Wr	Allows access to the interrupt vector or interrupt call-back executed by a probe input interrupt.
95	PROBE_INT_OCCUR	Board no.	Rd	Returns a value of 1 every time a probe input interrupt has occurred.
96	PROBE_SENSE	Board no.	Rd/Wr	Allows access to the probe sense control. Writing a value of 0/1 sets probe operation to active high/low.
97	PROBE_LED	Board no.	Rd/Wr	0 = Off, 1 =On, 2 = Auto
98	PROBE_SOUND	Board no.	Rd/Wr	0 = Off, 1 =On, n = time in ms
99	PROBE_FOOTSWITCH	Board no.	Rd/Wr	0 = Off, 1 = Auto, 2 = Auto (Inverted)

3.1.11 Probe information extended

Command & equate		Channel	Rd/Wr	Operation
100	PROBE_LATCH_SEL	Axis no.	Rd/Wr	Controls the source event of the 'Probe' associated latch of each axis.

Value	Source
0	Probe Occur
1	N/A (Read Counter)
2	Pulse Generator 0 Sync Occur
3	User Event 0 Occur
4	Channel Digital Input 0 Occur
5	Axis Compare 0 Sync Occur
6	Axis Compare 1 Sync Occur
7	N/A (Axis Compare 1 Enable)

101	PROBE_OUT_EN	Board no.	Rd/Wr	Enables the probe digital outputs. Specifying 1/0 in each bit field enables/disables the equivalent digital output.

Bit	Status	Pulse Width
0	Probe Occur	Occur to serviced
1	Footswitch Occur	Occur to serviced

3.1.12 Time-stamper information

Command & equate	Channel	Rd/Wr	Operation

110	TIMESTAMP_NOW	Board no.	Rd/Wr	Allows access to the current value of the 32-bit time-stamper register in units of 1μs.
111	TIMESTAMP_EVENT	Board no.	Rd	Allows access to the latched value of the 32-bit time-stamper register in units of 1μs.
112	TIMESTAMP_SEL	Board no.	Rd/Wr	Controls the source event of the Time-stamper latch.

Value	Source
0	Probe Occur
1	Timer 1 Sync Occur
2	User Event 0 Occur
3	Board Digital Input 0 Occur
4	Pulse Generator 0 Sync Occur
5	Axis Compare 0 Sync Occur
6	Axis Compare 1 Sync Occur
7	Reserved

3.1.13 Pulse generator information

Command & equate		Channel	Rd/Wr	Operation
120	PULSEGEN0	Axis no.	Rd/Wr	Allows access to the 16-bit pulse generator register in units of 1 quadrature count.
121	PULSEGEN0_MODE	Axis no.	Rd/Wr	Controls the mode of operation of the pulse generator.

Bit	Function
0	Direction when in Uni-directional mode (1/0 = +/-)
1	Bi-directional operation (1/0 = Bi-directional / Uni-directional)
2	Hardware Start/Stop by Axis Comparators (1/0 = On/Off)
3	Deglitch - Do not repeat the same pulse sequentially (1/0 = On/Off)

122	PULSEGEN0_EN	Axis no.	Rd/Wr	Writing a value of 1/0 enables/disables the pulse generator function.
123	PULSEGEN0_OCCUR	Axis no.	Rd	Returns a value of 1 every time a pulse generator interrupt has occurred.
124	PULSEGEN0_OUT_EN	Axis no.	Rd/Wr	Enables the pulse generator 0 digital outputs. Specifying 1/0 in each bit field enables/disables the equivalent digital output.

Bit	Status	Pulse Width
0	Pulse Generator 0 Occur	Occur to serviced
1	Pulse Generator 0 Terminal Count	Quadrature count width

125	PULSEGEN0_INT_VECT	Axis no.	Rd/Wr	Allows access to the interrupt vector or interrupt call-back executed by a pulse generator 0 interrupt.

3.1.14 Axis compare information

Command & equate	Channel	Rd/Wr	Operation
130 COMPARE0	Axis no.	Rd/Wr	Allows access to the 32-bit axis compare 0 register.
131 COMPARE0_FUNC	Axis no.	Rd/Wr	Writing a value of 1/0 enables/disables the axis compare 0 function.
132 COMPARE0_OCCUR	Axis no.	Rd	Returns a value of 1 every time an axis position compare 0 has occurred.
133 COMPARE0_OUT_EN	Axis no.	Rd/Wr	Writing a value of 1/0 enables/disables the axis compare 0 digital output.
134 COMPARE0_INT_VECT	Axis no.	Rd/Wr	Allows access to the interrupt vector or interrupt call-back executed by an axis compare 0 interrupt.
135 COMPARE1	Axis no.	Rd/Wr	Allows access to the 32-bit axis compare 1 register.
136 COMPARE1_FUNC	Axis no.	Rd/Wr	Writing a value of 1/0 enables/disables the axis compare 1 function.
137 COMPARE1_OCCUR	Axis no.	Rd	Returns a value of 1 every time an axis position compare 1 has occurred.
139 COMPARE1_INT_VECT	Axis no.	Rd/Wr	Allows access to the interrupt vector or interrupt call-back executed by an axis compare 1 interrupt.

3.1.15 User event information

Command & equate	Channel	Rd/Wr	Operation
140 USEREVENT0_OCCUR	Board no.	Rd/Wr	Returns a value of 1 every time a board user event 0 has been triggered and acknowledged. Writing any value triggers the board user event.
141 USEREVENT0_OUT_EN	I/O no.	Rd/Wr	Writing a value of 1/0 enables/disables the user event 0 digital output at the specified I/O bit, provided such mapping is possible.

3.1.16 Input event information

Command & equate	Channel	Rd/Wr	Operation
150 BOARD_INPUT_EN	Board no.	Rd/Wr	Writing a value of 1/0 enables/disables the equivalent board digital input positive edge detector.
151 BOARD_INPUT_OCCUR	Board no.	Rd	Returns a value of 1 every time a board digital input interrupt has occurred.
152 BOARD_INPUT_INT_VECT	Board no.	Rd/Wr	Allows access to the interrupt vector or interrupt call-back executed by an board digital input interrupt.

155	AXIS_INPUT_EN	Axis no.	Rd/Wr	Writing a value of 1/0 enables/disables the equivalent axis digital input positive edge detector.
156	AXIS_INPUT_OCCUR	Axis no.	Rd	Returns a value of 1 every time an axis digital input positive edge has occurred.
157	AXIS_INPUT_INT_VECT	Axis no.	Rd/Wr	Allows access to the interrupt vector or interrupt call-back executed by an axis digital input interrupt.

3.1.17 Digital I/O information

Command & equate	Channel	Rd/Wr	Operation
160 IO	I/O no.	Rd/Wr	Allows access to individual digital I/O register bits.
161 IO_DIR	I/O no.	Rd/Wr	Allows access to the direction of individual digital I/Os. This function only affects I/Os that can be individually configured. For further hardware details please refer to section 4.3.3 'Digital IO connections' of this manual.
165 IO_32	Reg no.	Rd/Wr	Allows access to 32 digital I/O register bits.
166 IO_32_DIR	Reg no.	Rd/Wr	Allows access to the direction of 32 digital I/Os. For I/Os whose direction can only be configured in groups, all bits of the group need to be set to the same direction. For further hardware details please refer to section 4.3.3 'Digital IO connections' of this manual.
169 NUM_IOS	N/A	Rd	Returns the total number of digital I/O available.

3.1.18 FIFO buffer information

The following tables list a number of tasks to be carried out when setting up and using a FIFO buffer.

Set up task	**Related Function(s)**
Configure buffer *clock* (which event triggers a data capture operation)	BUF_CLK_CHANNEL BUF_CLK_TYPE BUF_CLK_DIV
Configure buffer *data block* (how many and which *data elements* to be captured per buffer clock pulse)	BUF_NUM_DATA BUF_DATA_INDEX BUF_DATA_CHANNEL BUF_DATA_TYPE
Set buffer *size*	BUF_SIZE
Set buffer *mode*	BUF_MODE

Usage task	**Related Function**
Enable buffer	BUF_EN
Monitor amount of buffer contents	BUF_STATUS
Read buffer contents	BUF_READ

Command & equate	Channel	Rd/Wr	Operation
170 BUF_CLK_CHANNEL	Buffer no.	Rd/Wr	Controls the channel (timer / board / axis) of the event that triggers a buffer data capture.
171 BUF_CLK_TYPE	Buffer no.	Rd/Wr	Controls the type of event that triggers a buffer data capture.

Value	Timer Event
0	Timer 1 Occur
Value	**Board Event**
1	Probe Occur
2	Board Digital Input 0 Occur
3	User Event 0 Occur
Value	**Axis Event**
16	Marker Occur
17	Zero Occur
18	Pulse Generator 0 Occur
19	Axis Compare 0 Occur
20	Axis Compare 1 Occur
21	Channel Digital Input 0 Occur
22	SSI Read Complete Interrupt Occur

172 BUF_CLK_DIV Buffer no. Rd/Wr Controls the buffer clock divider.

173 BUF_NUM_DATA Buffer no. Rd/Wr Specifies the number of data elements to be captured in a FIFO buffer block, on each clock pulse.

174 BUF_DATA_INDEX Buffer no. Rd/Wr Selects which data element of a FIFO buffer block is accessed by data element functions BUF_DATA_CHANNEL and BUF_DATA_TYPE.

175 BUF_DATA_CHANNEL Buffer no. Rd/Wr Controls the channel (buffer / board / axis) of the data to be captured.

176 BUF_DATA_TYPE Buffer no. Rd/Wr Controls the type of data to be captured.

Value	Buffer Data	Note
0	Buffer clock counter	Resets on a buffer Enable, Flush or Configuration
Value	**Board Data**	
1	Time Stamp Now	See command 110 TIMESTAMP_NOW
2	Time Stamp Event	See command 111 TIMESTAMP_EVENT
3	Digital I/O	See command 165 IO_32
Value	**Axis Data**	
16	Timer Latch / SSI Latch	See command 48 TIMER_32 / 47 ABSOLUTE_32
17	Marker Latch	See command 41 MARK_32
18	Probe Latch	See command 46 PROBE_32
19	Incremental Position	See command 40 AXIS_32

180 BUF_SIZE Buffer no. Rd/Wr Allows the user to detect or specify the FIFO buffer size in data elements. Writing to this function disables and initialises data in the buffer. Buffer memory allocation succeeds if a non-zero value is returned.

181 BUF_MODE Buffer no. Rd/Wr Controls the mode of operation of the FIFO buffer.

Bit	Mode
0	Logging mode (1/0 = Discard old data when full / Discard new data when full)

182	BUF_EN	Buffer no.	Rd/Wr	Writing a value of 1/0 enables/disables FIFO buffer logging.
183	BUF_FLUSH	Buffer no.	Wr	Writing to this function clears the contents of the FIFO buffer. The value parameter passed to this command is ignored.
184	BUF_STATUS	Buffer no.	Rd	Allows access to the status register of the FIFO buffer.

Value	Status
-1	Overflow
0	Empty
+ve	Number of data elements currently in buffer

185	BUF_READ	Buffer no.	Rd	Returns a single data element from the FIFO buffer.
187	BUF_MEMFREE	Not used	Rd	Returns the total amount of free memory available to FIFO buffers in units of data elements.
188	BUF_MAXDATA	Not used	Rd	Returns the maximum number of data elements that can be captured by a FIFO buffer on each clock pulse.
189	NUM_BUFFERS	Not used	Rd	Returns the total number of FIFO buffers available to the system.

3.1.19 Software call-back information

Command & equate		Channel	Rd/Wr	Operation
200	NUM_LOSTCALLBACKS	Not used	Rd	Returns the number of lost software call-backs since this function was last read.

3.2 Function compatibility

No.	Equate	DEVA001 issue 3.x	DEVA001 issue 4.1.	DEVA001 issue 4.2+, 5.0+
0	VECTOR	Yes	No	No
1	NUM_AXES	3, channel 0..2	3, channel 0..2	3, channel 0..2
2	NUM_TIMERS	2, timer 1 for user only	2, timer 1 for user only	2, timer 1 for user only
3	NUM_INPUTS	6, input 0..5	0	0
4	NUM_DACS	0	0	0
5	NUM_OUTPUTS	0	0	0
7	NUM_BOARDS	0	0	Number of encoder cards
8	CARD_TYPE	Yes	Yes	Yes
9	VERSION_NUM	Yes	Yes	Yes
10	CNT_16	Yes	Yes	Yes
11	MODE	Mode 5 or SSI_CMR	Mode 5 or SSI_CMR	INC or SSI mode
12	AXIS_SIZE	2 x 16 bits	2 x 16 bits	2 x 16 bits
13	ENCODER_TYPE	Yes	Yes	Yes
14	AXIS_INPUTS	No	No	Yes
15	AXIS_STATUS	No	No	Yes
16	AXIS_OUT_EN	No	No	Yes
20	MARK_16	Yes	Yes	Yes
21	MARK_INPUT	Yes	Yes	Yes
22	MARK_INT	Yes	No	Yes
23	MARK_FUNC	Yes	Yes	Yes
24	MARK_INT_VECT	Yes	Yes	Yes
25	MARK_INT_OCCUR	Yes	Yes	Yes
26	MARK_LATCH_SEL	Defaults to marker	Defaults to marker	Yes
27	MARK_OUT_EN	No	No	Yes
30	ZERO_INPUT	Yes, channel 0 shared with probe input	No	Yes
31	ZERO_INT	Yes	No	Yes
32	ZERO_FUNC	Zero function performed in software	No	Zero function performed in software
33	ZERO_INT_VECT	Yes	No	Yes
34	ZERO_INT_OCCUR	Yes	No	Yes
40	AXIS_32	32 bit hardware counter	32 bit hardware counter	32 bit hardware counter
41	MARK_32	32 bit hardware latch value	32 bit hardware latch value	32 bit hardware latch value
42	VEL_INST	Software generated using timer 1 interrupt system	Yes based on software timer	Yes
43	VEL_FILT	Software generated using timer 1 interrupt system	Yes based on software timer	Yes
44	ACCEL_INST	Software generated using timer 1 interrupt system	Yes based on software timer	Yes
45	ACCEL_FILT	Software generated using timer 1 interrupt system	Yes based on software timer	Yes
46	PROBE_32	32 bit hardware latch value	32 bit hardware latch value	32 bit hardware latch value
47	ABSOLUTE_32	Yes	Yes	Yes

No.	Equate	DEVA001 issue 3.x.	DEVA001 issue 4.1	DEVA001 issue 4.2+, 5.0+
48	TIMER_32	No	No	32 bit hardware latch value
50	INPUT	Yes	No	No
60	TIMER	Timer 1 value x 0.1ms	Timer 1 value x 0.1ms *	Timer 1 value x 0.1ms
61	TIMER_INT	Timer 1	No	Yes
62	TIMER_INT_VECT	Timer 1	No	Yes
63	TIMER_INT_OCCUR	Timer 1	No	Yes
64	TIMER_LATCH_SEL	No	No	Yes
65	TIMER1_FUNC	No	No	Yes
66	TIMER_OUT_EN	No	No	Yes
70	DAC_MV	No	No	No
71	DAC_UV	No	No	No
80	OUTPUT	No	No	No
81	SERVO_ENABLE	No	No	No
90	PROBE_16	Yes	Yes	Yes
91	PROBE_INPUT	ZX / Renishaw option	Renishaw option	Renishaw option
92	PROBE_INT	Yes	No	Yes
93	PROBE_FUNC	Yes	Yes	Yes
94	PROBE_INT_VECT	Yes	Yes	Yes
95	PROBE_INT_OCCUR	Yes	Yes	Yes
96	PROBE_SENSE	Yes	No	No
97	PROBE_LED_MODE	Yes	Auto	Auto
98	PROBE_SOUND	Yes	Yes	Yes
99	PROBE_FOOTSWITCH	Yes	Yes	Yes
100	PROBE_LATCH_SEL	Defaults to probe	Defaults to probe	Yes
101	PROBE_OUT_EN	No	No	Yes
110	TIMESTAMP_NOW	No	No	Yes
111	TIMESTAMP_EVENT	No	No	Yes
112	TIMESTAMP_SEL	No	No	Yes
120	PULSEGEN0	No	No	Yes
121	PULSEGEN0_MODE	No	No	Yes
122	PULSEGEN0_EN	No	No	Yes
123	PULSEGEN0_OCCUR	No	No	Yes
124	PULSEGEN0_OUT_EN	No	No	Yes
125	PULSEGEN0_INT_VECT	No	No	Yes
130	COMPARE0	No	No	Yes
131	COMPARE0_FUNC	No	No	Yes
132	COMPARE0_OCCUR	No	No	Yes
133	COMPARE0_OUT_EN	No	No	Yes
134	COMPARE0_INT_VECT	No	No	Yes
135	COMPARE1	No	No	Yes
136	COMPARE1_FUNC	No	No	Yes
137	COMPARE1_OCCUR	No	No	Yes
139	COMPARE1_INT_VECT	No	No	Yes
140	USEREVENT0_OCCUR	No	No	Yes
141	USEREVENT0_OUT_EN	No	No	Yes

* 1ms granularity

No.	Equate	DEVA001 issue 3.x.	DEVA001 issue 4.1	DEVA001 issue 4.2+, 5.0+
150	BOARD_INPUT_EN	No	No	Yes
151	BOARD_INPUT_OCCUR	No	No	Yes
152	BOARD_INPUT_INT_OCCUR	No	No	Yes
155	AXIS_INPUT_EN	No	No	Yes
156	AXIS_INPUT_OCCUR	No	No	Yes
157	AXIS_INPUT_INT_VECT	No	No	Yes
160	IO	No	No	Yes
161	IO_DIR	No	No	No
165	IO_32	No	No	Yes
166	IO_32_DIR	No	No	Yes
169	NUM_IOS	No	No	Yes
170	BUF_CLK_CH	No	No	Yes
171	BUF_CLK_TYPE	No	No	Yes
172	BUF_CLK_DIV	No	No	Yes
173	BUF_NUM_DATA	No	No	Yes
174	BUF_DATA_INDEX	No	No	Yes
175	BUF_DATA_CH	No	No	Yes
176	BUF_DATA_TYPE	No	No	Yes
180	BUF_SIZE	No	No	Yes
181	BUF_MODE	No	No	Yes
182	BUF_EN	No	No	Yes
183	BUF_FLUSH	No	No	Yes
184	BUF_STAT	No	No	Yes
185	BUF_READ	No	No	Yes
187	BUF_MEMFREE	No	No	Yes
188	BUF_MAXDATA	No	No	Yes
189	NUM_BUFFERS	No	No	Yes
200	NUM_LOSTCALLBACKS	No	Yes	Yes

3.3 Device driver programming from 'C'

In order to simplify the user software required to access the MS-DOS and Windows 9x/NT4/2000/ME/XP device drivers, a selection of functions are supplied on the distribution CDROM. The functions are prototyped in the 'C' header file *enclib.h*. This section describes the 'C' functions provided for device driver access:

3.3.1 short open_encoder (void)

Opens the device driver and provides access to the functions provided.

Entry	:	none	
Exit	:	returns	0 if no error
		returns	-1 if error

3.3.2 void close_encoder (void)

Closes the device driver.

Entry	:	none	
Exit	:	returns	0 if no error
		returns	-1 if error

3.3.3 long read_encoder (short command, short channel)

Returns in a 32 bit integer the result of reading the device driver. See section 3.1 for a description of the command and channel parameters.

Entry	:	command	16 bit command
		channel	16 bit channel
Exit	:	returns	32 bit value

3.3.4 void write_encoder (short command, short channel, long value)

Writes a 32 bit integer to the device driver. See section 3.1 for a description of the command and channel parameters

Entry	:	command	16 bit command
		channel	16 bit channel
		value	32 bit value
Exit	:	none	

3.3.5 short enclib_callback (short receive, long priority);

Enables software call-backs. A user level function can be defined as call-back function by setting the function address (function pointer) as the interrupt vector value, using the appropriate *_INT_VECT function of section 3.1. Please note that this function is required only for Microsoft Windows operating systems. Call-backs are currently available only to a single software application / process.

Entry	:	receive	16 bit flag (1 to enable, 0 to disable)
		priority	32 bit call-back thread priority (defined in 'winbase.h')
			• For high speed operations:
			THREAD_PRIORITY_TIME_CRITICAL
			THREAD_PRIORITY_HIGHEST
			THREAD_PRIORITY_ABOVE_NORMAL
			THREAD_PRIORITY_NORMAL
			• For not real-time notifications:
			THREAD_PRIORITY_BELOW_NORMAL
			THREAD_PRIORITY_LOWEST
			THREAD_PRIORITY_IDLE
Exit	:	returns	0 if no error
		returns	1 if already enabled for this process
		returns	2 if a resource allocation error occurs

3.3.6 Example 'C' programming

```
/*---------------------------------------------------------------------------
     Example program to demonstrate device driver access
---------------------------------------------------------------------------*/
#include <stdlib.h>
#include <conio.h>
#include <stdio.h>
#include "enclib.h"
void main(void)
{
     long axis_position[12];       /* up to 4 cards x 3 axes */
     short num_channels;
     short i;
//   open device driver, exit if error
     if (open_encoder()==-1)
           exit(1);
//   read number of installed encoder channels
     num_channels=read_encoder(NUM_AXES,0L);

//   loop while not key pressed
     while(!kbhit())
           {
//         display axis positions
           for (i=0;i<num_channels;i++)
                        {
                        axis_position[i]=read_encoder(AXIS_32,i);
                        printf("%1u:%08lx ",i,axis_position[i]);
                        }
           printf("\r");
           }
//   close device driver
     close_encoder();
}
```

3.4 Device driver programming from Visual Basic

In order to simplify the user software required to access the Windows 7/Vista/XP 32/64 bit device drivers, a selection of Visual Basic functions is supplied on the distribution disk. The functions are declared in the *enclib.bas* module. This section describes the Visual Basic functions provided for device driver access:

3.4.1 Function open_encoder () As Integer

Opens the device driver and provides access to the functions provided.

Entry	:	none	
Exit	:	returns	0 if no error
		returns	-1 if error

3.4.2 Function close_encoder () As Integer

Closes the device driver.

Entry	:	none	
Exit	:	returns	0 if no error
		returns	-1 if error

3.4.3 Function read_encoder (ByVal com As Integer, ByVal chan As Integer) As Long

Returns in a 32 bit integer the result of reading the device driver. See section 3.1 for a description of the command and channel parameters

Entry	:	command	16 bit command
	:	channel	16 bit channel
Exit	:	returns	32 bit value

3.4.4 Function write_encoder (ByVal com As Integer, ByVal chan As Integer, ByVal value As Long) As Integer

Writes a 32 bit integer to the device driver. See section 3.1 for a description of the command and channel parameters

Entry	:	command	16 bit command
	:	channel	16 bit channel
	:	value	32 bit value
Exit	:	none	

3.4.5 Example Visual Basic programming

```
'--------------------------------------------------------------------------
'    Encoder card MSVB example
'--------------------------------------------------------------------------
Option Explicit
Dim Axes_name(1 To 12) As String
Dim Version_string As String
Dim Channel As Integer
Dim temp As Integer
Private Sub Form_Load()
    If (open_encoder() <> 0) Then
        Call MsgBox("Unable to load Driver Info", vbExclamation, "Digital
Read Out")
        End
    End If
    Call write_encoder(TIMER, 1, 10)
    Call write_encoder(TIMER_INT, 1, 1)

   For Channel = 1 To 3
        Call write_encoder(MARK_FUNC, Channel - 1, 1)
        Load LabelName(Channel)
        With LabelName(Channel)
            .Top = .Top + .Height * Channel
            .Visible = True
        End With
        Load LabelPosition(Channel)
        With LabelPosition(Channel)
            .Top = .Top + .Height * Channel
            .Visible = True
        End With
        Load LabelMark(Channel)
        With LabelMark(Channel)
            .Top = .Top + .Height * Channel
            .Visible = True
        End With
        Load LabelVelocity(Channel)
```

```
            With LabelVelocity(Channel)
                .Top = .Top + .Height * Channel
                .Visible = True
            End With
            Load LabelAccel(Channel)
            With LabelAccel(Channel)
                .Top = .Top + .Height * Channel
                .Enabled = True
                .Visible = True
            End With
        Next Channel
    End Sub
    Private Sub Form_Unload(Cancel As Integer)
        Call close_encoder
    End Sub
    Private Sub TimerUpdate_Timer()
        LabelName(1).Caption = "x"
        LabelName(2).Caption = "y"
        LabelName(3).Caption = "z"
        For Channel = 1 To 3
            LabelPosition(Channel).Caption = read_encoder(AXIS_32, Channel - 1)
            LabelMark(Channel).Caption = read_encoder(MARK_32, Channel - 1)
            LabelVelocity(Channel).Caption = read_encoder(VEL_INST, Channel - 1)
            LabelAccel(Channel).Caption = read_encoder(ACCEL_INST, Channel - 1)
        Next Channel
    End Sub
```

3.5 Device driver programming from C# .Net

In order to simplify the user software required to access the Windows 7/Vista/XP 32/64 bit device drivers, a selection of .Net functions are supplied on the distribution disk as the enclibNet.dll. This section describes the .Net functions provided for device driver access.

The DEVA001 .Net assembly consists of a single namespace: "Deva". In which is a single static class "Enclib". Within Enclib, all necessary methods and types commonly found in the 'C' header file.

3.5.1 public static short open_encoder()

Opens the device driver and provides access to the functions provided.

Entry	:	none	
Exit	:	returns	0 if no error
		returns	-1 if error

3.5.2 public static void close_encoder()

Closes the device driver.

Entry	:	none	
Exit	:	returns	0 if no error
		returns	-1 if error

3.5.3 public static int read_encoder(short command, short channel)

Returns a 32bit integer, which contains the result from the device driver. See section 3.1 for a description of the command and channel parameters.

Entry	:	command	16 bit command
	:	channel	16 bit channel
Exit	:	returns	32 bit value

3.5.4 public static void write_encoder(short command, short channel, int value)

Writes a 32bit integer to the device driver. See section 3.1 for a description of the command and channel parameters

Entry	:	command	16 bit command
	:	channel	16 bit channel
	:	value	32 bit value
Exit	:	none	

3.5.5 Example C# programming

```
/*-------------------------------------------------------------------------
 *    Encoder card MSCS example
 * ------------------------------------------------------------------------*/

using System;
using Deva;

namespace Deva_mscsExample
{
  class Program
  {
    static void Main()
    {
      Console.WriteLine("DEVA001 MSC# Example");

      int num_channels = 12;  /* up to 4 cards x 3 axes */
      int[] axis_position = new int[num_channels];

      //open device driver, exit if error
      if (Enclib.open_encoder() == -1)
        return;

      //read number of installed encoder channels
      num_channels = Enclib.read_encoder((short)Enclib.COMMANDS.NUM_AXES, 0);

      //loop while key not pressed
      while(!Console.KeyAvailable)
      {
        //display axis positions
        for (short index = 0; index < num_channels; index++)
        {
          axis_position[index] = Enclib.read_encoder((short)Enclib.COMMANDS.AXIS_32, index);
          Console.WriteLine("Axis:" + index.ToString() + "\tPosition: " +
axis_position[index]);
        }
        Console.SetCursorPosition(0, 1);
      }

      //close device driver
      Enclib.close_encoder();
    }
  }
}
```

4 PCI/PCIE Incremental encoder interface hardware

4.1 Functional description

The DEVA001 is designed to interface up to three incremental encoders or linear scales to the PC PCI/PCIE bus. The following sections describe the various functions of the interface, for detailed description of how to program these facilities please refer to section 4.4.

4.1.1 Quadrature input

Each encoder channel has four differential / single ended input circuits designated A, B, Z and M. The A and B inputs accept the quadrature signals from the encoder and drive an up/down counter via a x4 directional discriminator circuit. The counter has 32bit resolution and may be read at any time. The maximum count rate is in excess of 10MHz.

4.1.2 Marker input

The M input circuit designated marker is a positive edge triggered input which can synchronously latch the counter value. To gain the most accurate result the input conditioning circuit latches the counter when both A and B inputs are at a logic one level. The user must therefore phase the A and B signals carefully to meet this criteria. Should this not be achievable circuit operation is still possible although the latched value will not be accurate to a single count. Correct phasing may also be achieved by using the input invertors on signals A, B & M.

To use the marker input the marker circuit must be enabled. Once a positive edge has occurred on the marker input the counter is latched when both A and B are at logic one. The marker latch register may now be read while the counter continues to maintain position. To enable the maker latch to capture a further marker event the marker function should be disabled and then re-enabled. Marker synchronisation with signals A & B can optionally be disabled.

4.1.3 Zero input

The Z input circuit designated zero is a positive edge triggered input which can asynchronously latch the counter value. It has been provided to maintain compatibility with previous issues and offer more flexibility when connecting encoders. The marker function which accurately latches the counter value is equally suited to performing a zero type operation by using the latched value as an offset which is subsequently subtracted from the counter reading.

4.1.4 Renishaw Probe Interface

The DEVA001 incorporates a Renishaw compatible probe input. This input provides the correct signal conditioning to allow a volt-free contact type touch probe to be used. The probe function works like the marker function but latches all three axis counters simultaneously. Unlike the marker function it operates asynchronously and hence does not require that the A and B signals are at a logic one level. The probe function has its own enable bit.

The card also incorporates an output to drive the probe led, an output to drive an audible sounder and a further input to allow the probe to be controlled via a footswitch.

4.1.5 Digital IO

The DEVA001 has 48 digital IO lines arranged in 3 groups of 16 bits. The first group of 16 bits may be associated with special input and output functions specific to other features of the card and has hardwired directions. The direction of the other 2 groups may be programmed in blocks of 8 bits.

4.1.6 Pulse Generator

The DEVA001 has a 16bit Pulse Generator circuit which can produce a pulse every n counts of the X axis encoder, where n is programmable from 1 to 65536. The pulse can be sent directly to a digital output, cause a hardware event or generate an interrupt. Please refer to the event system diagrams for details of which actions can be triggered by the pulse generator event.

The pulse generator range of operation (start and stop positions) may optionally be controlled through hardware. If hardware control is selected the axis compare functions are used as start and stop positions. Please note that the pulse generator function needs to be enabled for this feature to operate.

The pulse generator has both uni-directional and bi-directional modes, configurable direction when in uni-directional mode and incorporates an optional de-glitch circuit that does not allow a pulse to be generated at the same position sequentially.

4.1.7 Axis Compare

The X axis has two axis compare registers which may be used to generate events when the axis position matches the compare position. The axis compare registers are shared with the Marker and Probe latches.

4.1.8 Timer

The timer is an 8 bit interval timer which can be programmed in steps of 0.1ms up to a maximum of 25.6ms. This timer may be used to generate interrupts, latch encoder positions or trigger a digital output for external circuitry synchronisation.

4.1.9 Time Stamper

The time stamper is a 16 bit high speed timer with a resolution of 1us. It can be programmed to latch its value on an event so to provide an accurate time stamp at which the event occurred.

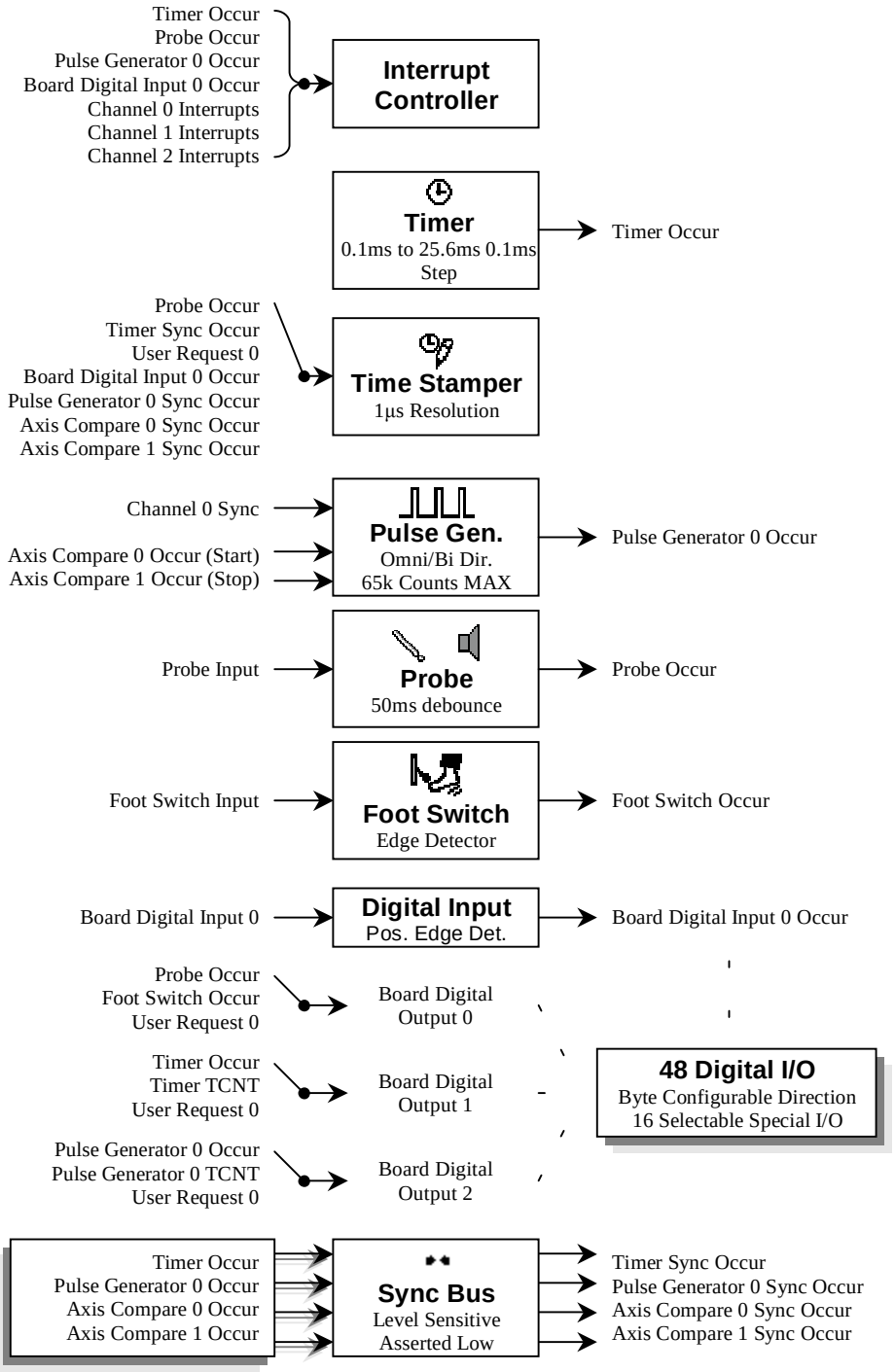
4.1.10 Event System

Many features of the card produce events. When an event occurs a flag called the occur flag is set. The occur flag may be polled by the user to establish that the event has occurred.

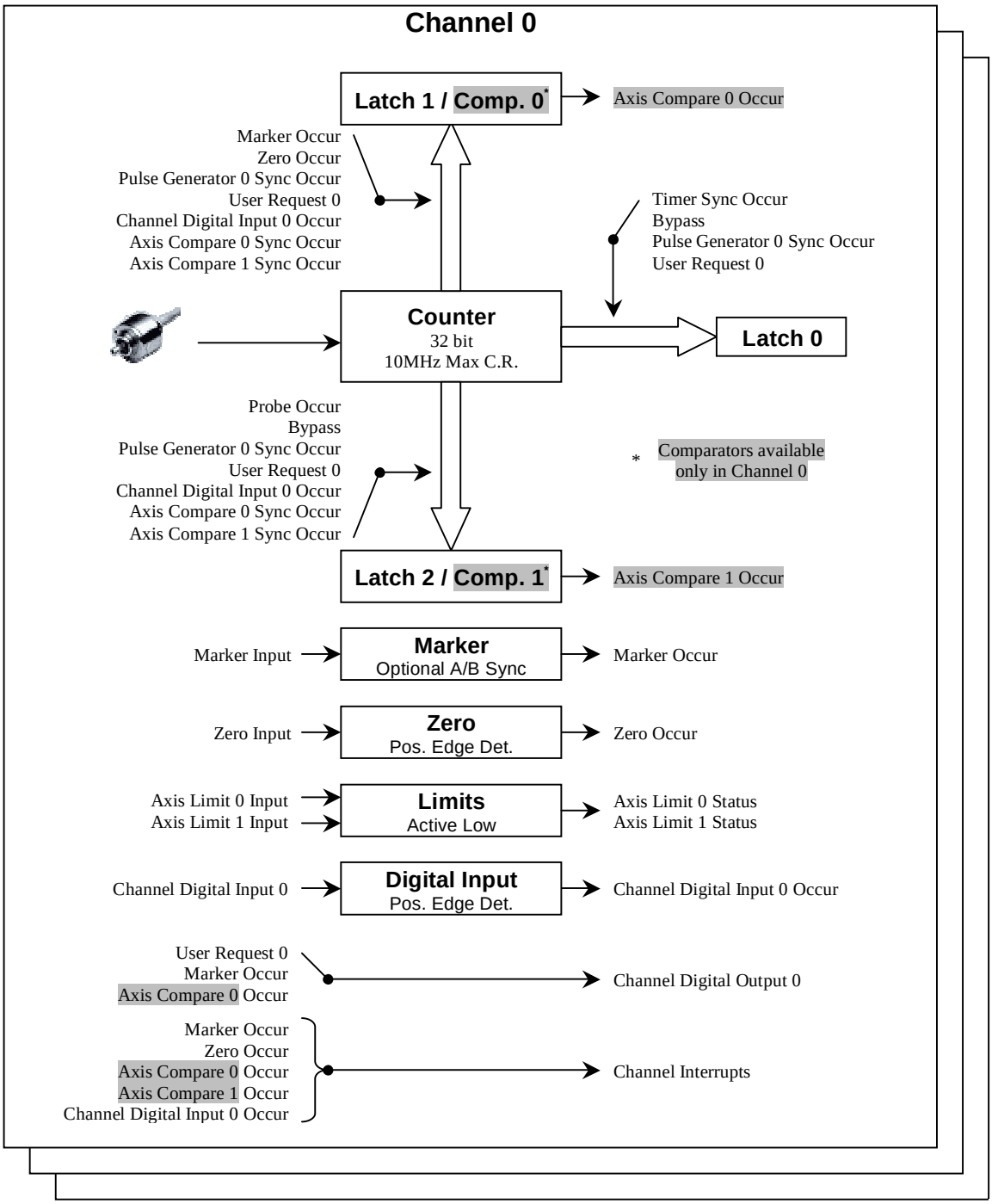
The card has a variety of actions which may be configured to respond to events. The actions include latching the current position in one of three latches, generating an interrupt, triggering a digital output or latching a time stamp of the event.

The following diagrams show the board level and channel level events and actions which may be programmed.

4.1.10.1Incremental board level block diagram

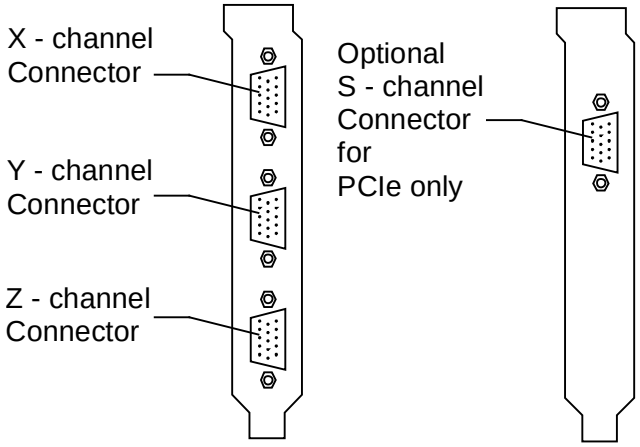


4.1.10.2Incremental channel level block diagram



4.2 Connection details

The DEVA001 has three input channels for quadrature signals from incremental encoders, designated x, y & z. Connection of each channel to the outside world is made through a 15 way HD-type connector. The designation of the connectors is shown in the diagram below.



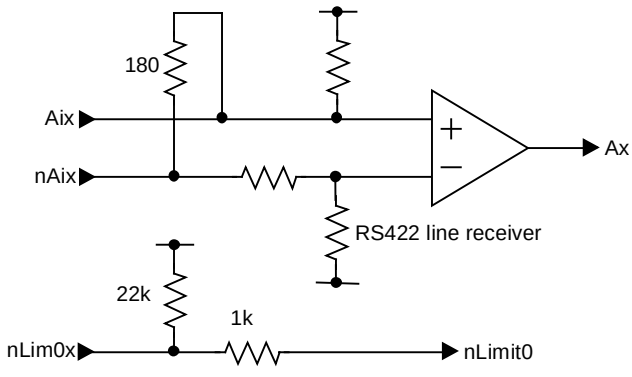
The PCIe card has an additional 4th channel. This is available on the 16 pin header labelled "S-CHAN". An additional ribbon cable with 15 way HD connector and PC bracket shown above is available upon request.

4.3 Encoder input connections

Connections to these sockets should be made with reference to the following pin-out table and simplified input circuit.

Pin Number	Signal	Function
1	Ai	A phase input
2	Bi	B phase input
3	Zi	Zero input
4	Mi	Marker input
5	nLim0	nLimit 0
6	nAi	nA phase input
7	nBi	nB phase input
8	nZi	nZero input
9	nMi	NMarker input
10	nLim1	nLimit 1
11	+12V	+12 volts supply
12	+5V	+5 volt supply
13	0V	0 volt common
14		
15	-12V	-12 volt supply*

*PCI only.



Note: Do not connect the 15-way D-type plug from a VGA monitor into one of the encoder input channels, as damage may result.

4.3.1 Input signal descriptions

The Ai & nAi, Bi & nBi inputs are differential pairs for connection to the A-phase and B-phase quadrature outputs of an incremental encoder.

The Mi & nMi inputs are differential inputs for the channel marker signal. This function will latch the counter reading for the relevant channel, allowing an accurate reading of the position of a moving encoder to be made at a specific instant. The marker function does not stop the counter itself, which is able to continue reading the encoder position and so it will not cause the card to lose track of the system's datum position.

The Zi & nZi may be used as an alternative to the Mi & nMi inputs. In this case the driver will zero the counter readings in software.

The inputs nLim0 and nLim1 are digital inputs intended to be used to connect normally closed overtravel inputs.

The differential inputs use RS422 levels which accept OV for logic low, and from +5V to +12V for logic high. Because they are differential inputs, one input should be low when the other is high. For Example, to trigger the marker function, set Mi input to high, and set nMi input to low. To turn the marker function off, reverse these voltages.

An internal resistor network is provided which allows connection of single ended signals to the non-inverting inputs.

The +/- 12V and +5V power supply pins may be used to supply the dc power requirements for the encoders. –12V is not available from the PCIe variant.

4.3.2 Renishaw probe input connections

The Renishaw connection kit provides a 9 way female 'D' connector which may be used with the bracket supplied or mounted in a free cut-out on the PC case. Connections to this socket should be made with reference to the following pin-out table.

Pin Number	Signal	Function
1	Led C	Led cathode
2	0v	(0v)
3	Led A	Led anode
4	Probe +	Probe
5	Probe -	(0v)
6	Siren +	Siren +5v
7	Siren -	(0v)
8	FootSw +	Footswitch input
9	FootSw -	(0v)

4.3.3 Digital IO connections

There are three 20 pin headers labelled IO0, IO1 and IO2 which provide access to 3 groups of 16 bits of digital IO. The IO conforms to 5v TTL levels, Voh min 2.4v at –8ma and Vil maximum 0.4 at 12ma.

4.3.3.1 Digital IO connector 0 connections

The direction of IO connector 0 pins is fixed. Pins 1 to 8 (IO0 to IO7) are outputs. Pins 9 to 16 (IO8 to IO15) are inputs.

Pin	Connector IO0	Special Function
1	IO0 / UR0 / MOx / AC0Ox	User request 0 / Marker occur x / Axis compare 0 occur
2	IO1 / UR0 / MOy	User request 0 / Marker occur y
3	IO2 / UR0 / MOz	User request 0 / Marker occur z
4	IO3 / UR0 / MOs	User request 0 / Marker occur s
5	IO4 / PO / FO / UR0	Probe occur / Footswitch occur / User request 0
6	IO5 / TMRO / TMRTCNT / UR0	Timer occur / Timer terminal count / User request 0
7	IO6 / PG0O / PG0TCNT / UR0	Pulse gen. 0 occur / Pulse gen. 0 terminal count / User request 0
8	IO7 / - / - / UR0	User request 0
9	IO8 / CDI0x	Channel x digital input 0
10	IO9 / CDI0y	Channel y digital input 0
11	IO10 / CDI0z	Channel z digital input 0
12	IO11 / CDI0s	Channel s digital input 0
13	IO12 / BDI0	Board digital input 0
14	IO13	
15	IO14	
16	IO15	
17	GND	
18	GND	
19	+5v	
20	+12v	

4.3.3.2 Digital IO connectors 1 and 2 connections

The direction of IO connectors 1 and 2 pins is software configurable in groups of 8 (1 to 8 and 9 to 16).

Pin	Connector IO1	Connector IO2
1	IO16	IO32
2	IO17	IO33
3	IO18	IO34
4	IO19	IO35
5	IO20	IO36
6	IO21	IO37
7	IO22	IO38
8	IO23	IO39
9	IO24	IO40
10	IO25	IO41
11	IO26	IO42
12	IO27	IO43
13	IO28	IO44
14	IO29	IO45
15	IO30	IO46
16	IO31	IO47
17	GND	GND
18	GND	GND
19	+5v	+5v
20	+12v	+12v

4.3.4 Sync connector

The sync connector provides a method of linking cards in hardware to allow events to be routed from one card to another. In general a ribbon cable connecting all 10 connections between all cards is required.

Pin Number	Function
1	Sync0 (Timer occur)
2	GND
3	Sync1 (Pulse Generator 0 occur)
4	GND
5	Sync2 (Axis Compare 0 occur)
6	GND
7	Sync3 (Axis Compare 1 occur)
8	GND
9	GND
10	GND

4.4 Direct hardware programming

The DEVA001 incremental encoder interface card is supplied with a variety of device drivers for Microsoft Windows operating systems which perform all low level access functions required for its operation. Using a device driver offers several benefits including the ability to re-use application software and routines with any of DEVA Electronics compatible products. The device drivers and the common software interface are described in section 3. For applications where hardware access is essential, the following sections give an overview of the register set and card functionality.

4.4.1 Register address map

The card implements several 32 bit read/write registers which are grouped by function and appear at different offsets within the memory space allocated to the card by the plug and play bios / operating system.

For PCIe the 32 bit registers appear in the lower 32 bits of 64 bit locations and hence all byte offsets are multiplied by 2.

Offset		Read Function	Write Function
PCI	PCIe		
00h	00h	**Special Function status / control register** SFS_{31} — 16 | SFR_{15} — 0	**Special Function clear / control register** SFC_{31} — 16 | SFR_{15} — 0
04h	08h	**Interrupt request / mask register** IRR_{31} — 16 | IMR_{15} — 0	**Interrupt mask register** IMR_{15} — 0
08h	10h	**Timer counter / latch** $TMRC_{31}$ — 16 | $TMRL_7$ — 0	**Timer latch** $TMRL_7$ — 0
0Ch	18h	**Time Stamper counter / latch** $TSTC_{31}$ — 16 | $TSTL_{15}$ — 0	
10h	20h	**Pulse Generator 0 counter / latch** $PG0C_{31}$ — 16 | $PG0L_{15}$ — 0	**Pulse Generator 0 latch** $PG0L_{15}$ — 0
20h	40h	**Digital I/O bus 0 mux / status** DOM_{31} — 16 | $DIOS_{15}$ — 0	**Digital I/O bus 0 mux / control** DOM_{31} — 16 | DOC_7 — 0
24h	48h	**Digital I/O bus 1 direction / status** $DIOD_{17}$ — 16 | $DIOS_{15}$ — 0	**Digital I/O bus 1 control** $DIOC_{15}$ — 0
28h	50h	**Digital I/O bus 2 direction / status** $DIOD_{17}$ — 16 | $DIOS_{15}$ — 0	**Digital I/O bus 2 control** $DIOC_{15}$ — 0
40h	80h	**X Channel status / control register** CSR_{31} — 16 | $TSTL_{15}$ — 0	**X Channel clear / control register** $CCLR_{31}$ — 16 | CCR_{15} — 0
44h	88h	**Y Channel status / control register** CSR_{31} — 16 | $TSTL_{15}$ — 0	**Y Channel clear / control register** $CCLR_{31}$ — 16 | CCR_{15} — 0
48h	90h	**Z Channel status / control register** CSR_{31} — 16 | $TSTL_{15}$ — 0	**Z Channel clear / control register** $CCLR_{31}$ — 16 | CCR_{15} — 0
	A0h	**S Channel status /** CSR_{31} — 16 | **S Channel clear /** $TSTL_{15}$ — 0	**S Channel clear / control register** $CCLR_{31}$ — 16 | CCR_{15} — 0

Offset		Read Function	Write Function
PCI	PCIe		
60h	C0h	**X Timer Latch** D31 — 0	**X Timer Latch** D31 — 0
64h	C8h	**Y Timer Latch** D31 — 0	**Y Timer Latch** D31 — 0
68h	D0h	**Z Timer Latch** D31 — 0	**Z Timer Latch** D31 — 0
	E0h	**S Timer Latch** D31 — 0	**S Timer Latch** D31 — 0
80h	100h	**X Marker latch** D31 — 0	**X Marker latch** D31 — 0
84h	108h	**Y Marker latch** D31 — 0	**Y Marker latch** D31 — 0
88h	110h	**Z Marker latch** D31 — 0	**Z Marker latch** D31 — 0
	120h	**S Marker latch** D31 — 0	**S Marker latch** D31 — 0
A0h	140h	**X Probe latch** D31 — 0	**X Probe latch** D31 — 0
A4h	148h	**Y Probe latch** D31 — 0	**Y Probe latch** D31 — 0
A8h	150h	**Z Probe latch** D31 — 0	**Z Probe latch** D31 — 0
	160h	**S Probe latch** D31 — 0	**S Probe latch** D31 — 0

4.4.1.1 Special function control register – PCI 00h, PCIe 00h

The lower 16 bits of the register at offset 00h are designated the Special function control register. This read/write register allows control of board based facilities. The register contents are zero after system reset.

Special function control register:

Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
UR0	PG0cr[3:0]				PG0ie	PG0e	DI0ie	TSTs[2:0]			TMRe	Fs	Be	Fe	Pe

Pe	Enable / Disable probe function
Fe	Enable / Disable footswitch function
Be	Enable / Disable buzzer output
Fs	Footswitch input sense
TMRe	Enable / Disable timer function
TSTs	Time stamper event select register
DI0ie	Enable / Disable board digital input 0 interrupt
PG0e	Enable / Disable pulse generator 0 function
PG0ie	Enable / Disable pulse generator 0 interrupt
PG0cr	Pulse generator 0 control register
UR0	Enable / Disable user request 0 flag

Time stamper event sources:

TSTs	Source
0	Probe occur flag
1	Timer Sync occur flag
2	User Request 0 flag
3	Board Digital Input 0 occur flag
4	Pulse Generator 0 Sync occur flag
5	Axis Compare 0 Sync occur flag
6	Axis Compare 1 Sync occur flag
7	Reserved

Pulse Generator 0 control register bit fields:

PG0cr bit	Function
0	Direction of operation (1=positive, 0=negative)
1	Bi-Directional mode (1=On, 0=Off)
2	Hardware Control Start/Stop (1=On, 0=Off)
3	Deglitch mode (1=On, 0=Off)

4.4.1.2 Special function clear register – PCI 00h, PCIe 00h

The upper 16 bits of the register at offset 00h are designated the Special function clear register. This write-only register can be used to clear occur flags of board-based facilities by writing '1' to the equivalent bit. The status of all relevant flags is cleared after system reset.

Special function clear register:

Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16
-	-	-	-	-	-	PG0c	DI0c	-	-	TSTc	TMRc	-	-	-	-

TMRc	Clear timer occur flag
TSTc	Clear time stamper overflow occur flag
DI0c	Clear board digital input 0 occur flag
PG0c	Clear pulse generator 0 occur flag

4.4.1.3 Special function status register – PCI 00h, PCIe 00h

The upper 16 bits of the register at offset 00h are designated the Special function status register. This read only register allows access to board based status information.

Special function status register:

Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16
-	-	-	-	-	-	PG0o	DI0o	AC1o	AC0o	TSTo	TMRo	Fi	nPi	Fo	Po

Po	Probe occur flag
Fo	Footswitch occur flag
nPi	Direct probe input (inverted)
Fi	Direct footswitch input
TMRo	Timer occur flag
TSTo	Time Stamper overflow occur flag
AC0o	Axis Compare 0 occur flag
AC1o	Axis Compare 1 occur flag
DI0o	Board Digital Input 0 occur flag
PG0o	Pulse Generator 0 occur flag

4.4.1.4 Interrupt mask register – PCI 04h, PCIe 08h

This 16 bit read / write register sets the interrupt masks to select which interrupt sources generate interrupts. A logic zero disables an interrupt and a logic one enables an interrupt. The register holds zero after system reset. The bit assignment for this register is as follows:

Interrupt mask:

Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
-	-	-	-	CH3ie	CH2ie	CH1ie	CH0ie	-	-	-	-	-	-	BRDie	TMRie

Bit	Description	Source
TMRie	Timer interrupt enable	Timer (reload) occur flag
BRDie	Board interrupts enable	Board occur flags (Probe, Pulse Generator 0, Board Digital Input 0)
CH0ie	Channel 0 interrupts enable	Channel 0 occur flags (Marker, Zero, Channel Digital Input 0, Axis Compare 0/1)
CH1ie	Channel 1 interrupts enable	Channel 1 occur flags (Marker, Zero, Channel Digital Input 0)
CH2ie	Channel 2 interrupts enable	Channel 2 occur flags (Marker, Zero, Channel Digital Input 0)
CH3ie	Channel 2 interrupts enable	Channel 3 occur flags (Marker, Zero, Channel Digital Input 0)

4.4.1.5 Interrupt request register – PCI 04h, PCIe 08h

This 16 bit read-only register indicates which interrupt sources have generated interrupts. Logic one indicates that an interrupt has occurred. To clear an interrupt request, the occurred flags of all associated interrupt sources must be cleared. The register holds 00h after system reset. Please note that since the PCI/PCIE interrupt system is level sensitive, all interrupt requests must be cleared to free the interrupt line allocated to the device. The bit assignment for this register is as follows:

Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16
-	-	-	-	CH3io	CH2io	CH1io	CH0io	-	-	-	-	-	-	BRDio	TMRio

TMRio	Timer interrupt occurred
BRDio	Board interrupt(s) occurred
CH0io	Channel 0 interrupt(s) occurred
CH1io	Channel 1 interrupt(s) occurred
CH2io	Channel 2 interrupt(s) occurred
CH3io	Channel 3 interrupt(s) occurred

4.4.1.6 Timer latch – PCI 08h, PCIe 10h

The lower 8 bits of the register at offset 08h are designated the Timer latch. This read / write register specifies the reload value (in 0.1ms) of the Timer function. This register holds FFh after system reset.

4.4.1.7 Timer counter – PCI 08h, PCIe 10h

The upper 16 bits of the register at offset 08h are designated the Timer counter. This read-only register returns the current value of the (count-down) timer counter (in 0.1ms). When the timer function is disabled, the timer counter resets to the value of the timer latch.

4.4.1.8 Time Stamper latch – PCI 0Ch, PCIe 18h

The lower 16 bits of the register at offset 0Ch are designated the Time Stamper latch. This read-only register allows access to the timestamp of the event associated with the Time Stamper. The register holds zero after system reset.

4.4.1.9 Time Stamper counter – PCI 0Ch, PCIe 18h

The upper 16 bits of the register at offset 0Ch are designated the Time Stamper counter. This read-only register returns the current value of the (count-up) time stamper counter (in 1us).

4.4.1.10 Pulse Generator 0 latch – PCI 10h, PCIe 20h

The lower 16 bits of the register at offset 10h are designated the Pulse Generator 0 latch. This read / write register specifies the reload value (in quadrature counts) of

the pulse generator 0 function. Writing to this register resets the pulse generator 0 counter. This register holds zero after system reset.

4.4.1.11 Pulse Generator 0 counter – PCI 10h, PCIe 20h

The upper 16 bits of the register at offset 10h are designated the Pulse Generator 0 counter. This read-only register returns the current value of the pulse generator 0 (up/down) counter (in quadrature counts). The pulse generator 0 counter resets to zero on a write operation to the pulse generator 0 latch. This register holds zero after system reset.

4.4.1.12 Digital I/O control / status register – PCI 20h, 24h, 28h, PCIe 40h,48h,50h

The lower 16 bits of these registers are designated the Digital I/O control / status registers. These read / write registers allow control of the digital outputs and access to the digital inputs of the equivalent I/O bus. The register contents are zero after system reset.

Please note that on Digital I/O bus 0, I/Os 0 to 7 are hardwired as outputs and I/Os 8 to 15 are hardwired as inputs.

4.4.1.13 Digital I/O bus 0 mux register – PCI 20h, PCIe 40h

The upper 16 bits of this register are designated the Digital I/O bus 0 mux register. This read / write register allows control of the multiplexers (source selectors) of digital outputs on I/O bus 0. For each digital output there are 2 selection bits, allowing for 4 possible signal sources. For a list of possible source routings on each pin, please refer to section 4.3.3.1.

Digital I/O bus 0 mux register:

Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16
DO7s[1:0]		DO6s[1:0]		DO5s[1:0]		DO4s[1:0]		DO3s[1:0]		DO2s[1:0]		DO1s[1:0]		DO0s[1:0]	

4.4.1.14 Digital I/O direction register – PCI 24h, 28h, PCIe 48h,50h

The upper 16 bits of this register are designated the Digital I/O direction registers. These read / write registers control the direction of digital I/Os.

Digital I/O direction register:

Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16
-	-	-	-	-	-	-	-	-	-	-	-	-	-	IOb1d	IOb0d

IOb0d	Direction of digital I/O byte 0 (1=Input, 0=Output)
IOb1d	Direction of digital I/O byte 1 (1=Input, 0=Output)

4.4.1.15Channel control register – PCI 40h, 44h, 48h, PCIe 80h,88h,90h,98h

The lower write-only 16 bits of these registers are designated the Channel control registers. These read / write registers allow control of channel based facilities. The register contents are zero after system reset.

Channel control register:

Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
L2s[2:0]			L1s[2:0]			L0s[1:0]		DI0ie	MSe	Ze	invZ	invM	invB	invA	Me

Me	Enable / Disable marker function
invA	Invert signal A
invB	Invert signal B
invM	Invert signal M
invZ	Invert signal Z
Ze	Enable / Disable zero function
MSe	Enable / Disable marker synchronisation with signals A & B
DI0ie	Enable / Disable channel digital input 0 interrupt
L0s	Latch 0 (Timer latch) select register
L1s	Latch 1 (Marker latch) select register
L2s	Latch 2 (Probe latch) select register

Source events of Latch 0 (Timer latch):

L0s	Source
0	Timer Sync occur flag
1	Count (default)
2	Pulse Generator 0 Sync occur flag
3	User Request 0 flag

Source events of Latch 1 (Marker latch):

L1s	Source
0	Marker occur flag
1	Zero occur flag
2	Pulse Generator 0 Sync occur flag
3	User Request 0 flag
4	Channel Digital Input 0 occur flag
5	Axis Compare 0 Sync occur flag
6	Axis Compare 1 Sync occur flag
7	Count

Source events of Latch 2 (Probe latch):

L2s	Source
0	Probe occur flag
1	N/A (direct counter read)
2	Pulse Generator 0 Sync occur flag
3	User Request 0 flag
4	Channel Digital Input 0 occur flag
5	Axis Compare 0 Sync occur flag
6	Axis Compare 1 Sync occur flag
7	Count

4.4.1.16 Channel time Stamper latch – PCI 40h, 44h, 48h, PCIe 80h,88h,90h,98h

The lower read-only 16 bits of these registers are designated the Time Stamper latch. This read-only register allows access to the timestamp of the event associated with the Time Stamper. The register holds zero after system reset.

4.4.1.17 Channel clear register and extended channel control register – PCI 40h, 44h, 48h, PCIe 80h,88h,90h,98h

The upper bits [31:24] of these registers are designated the Channel clear registers. These write-only registers can be used to clear occur flags of channel-based facilities by writing '1' to the equivalent bit. The status of all relevant flags is cleared after system reset. The upper bits [21:16] are designated the extended channel control registers and are write-only.

Channel clear and extended channel control register:

Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16
-	-	-	AC1c	AC0c	ER1c	ER0c	DI0c	-	-	CNTM[1:0]		CNTI	L3s[2:0]		

DI0c	Clear channel digital input 0 occur flag
ER0c	Clear error 0 (quadrature) occur flag
ER1c	Clear error 1 (power) occur flag
AC0c	Clear axis compare 0 occur flag
AC1c	Clear axis compare 1 occur flag

Source events of Latch 3 (Time stamp latch):

L3s	Source
0	Count
1	Marker occur flag
2	Zero occur flag
3	Pulse Generator 0 Sync occur flag
4	User Request 0 flag
5	Channel Digital Input 0 occur flag
6	Axis Compare 0 Sync occur flag
7	Axis Compare 1 Sync occur flag

CNTI	Count inhibit

Count mode:

CNTM	Source
0	Quadrature AB decode
1	Count rising and falling edges of A, B selects count up/down direction
2	Count rising edges of A, B selects count up/down direction

4.4.1.18Channel status register – PCI 40h, 44h, 48h, PCIe 80h,88h,90h,98h

The upper 16 bits of these registers are designated the Channel status registers. These read-only registers allow access to channel based status information.

Channel status register:

Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16
-	-	-	AC1o	AC0o	ER1o	ER0o	DI0o	nLIM1	nLIM0	Zo	Zi	Mi	Bi	Ai	Mo

Mo	Marker occur flag
Ai	Direct A channel input
Bi	Direct B channel input
Mi	Direct M (marker) channel input
Zi	Direct Z (zero) channel input
Zo	Zero occur flag
nLIM0	Direct Limit 0 channel input (inverted)
nLIM1	Direct Limit 1 channel input (inverted)
DI0o	Channel digital input 0 occur flag
ER0o	Error 0 (quadrature) occur flag
ER1o	Error 1 (power) occur flag
AC0o	Axis compare 0 occur flag
AC1o	Axis compare 1 occur flag

4.4.1.19Latch 0 (Timer latch) – PCI 60h, 64h, 68h, PCIe c0h,c8h,d0h,d8h

These three registers allow access to the three 32 bit Timer latches. Each latch stores the counter value when the associated source event occurs and should only be read when the occur flag of that event is set. Various source events can be selected by the equivalent source event selector, accessible through the channel control register. The register contents are zero after system reset.

4.4.1.20Latch 1 (Marker latch) – PCI 80h,84h,88h, PCIe 100h,108h,110h,118h

These three registers allow access to the three 32 bit Marker latches. Each latch stores the counter value when the associated source event occurs and should only be read when the occur flag of that event is set. Various source events can be selected by the equivalent source event selector, accessible through the channel control register. The register contents are zero after system reset.

4.4.1.21Latch 2 (Probe latch) – PCI a0h,a4h,a8h, PCIe 120h,128h,130h,138h

These three registers allow access to the three 32 bit Probe latches. Each latch stores the counter value when the associated source event occurs and should only be read when the occur flag of that event is set. Various source events can be selected by the equivalent source event selector, accessible through the channel control register. The register contents are zero after system reset.

5 PCI/PCIE Absolute SSI interface hardware

5.1 Functional description

The DEVA001 is designed to interface up to three absolute SSI encoders or linear scales to the PC PCI/PCIE bus. The following sections describe the various functions of the interface, for a detailed description of how to program these facilities please refer to section 5.3.

5.1.1 Programmable clock frequency, bit length and offset

The clock frequency and encoder bit length may be independently programmed for each encoder. The frequency may be set to one of eight values in the range 78 KHz to 2.5 MHz (see sections 5.2.2 for details). The bit length may be programmed in the range 1 to 32 bits. The time taken to make a reading will depend on both the clock frequency and the bit length as follows :

Reading time = (bit length + 2) / (Clock frequency)

Bit length	Clock frequency	Reading time
12	625 KHz	22.4us
13	625 KHz	24us
21	2.5 MHz	9.2us

The offset feature allows the data read from the encoder to be shifted left a further 0-15 bits. This is typically used to align the angular part of the encoder data to the 16 bit boundary. In this case the MSW represents the number of turns and the LSW represents the angular component.

Consider a system which has the following three encoders :

i 15 bit single turn encoder

ii 12 bit single turn encoder

iii 21 bit multi-turn encoder with 13 bits of angular resolution

By applying offsets of 1, 4 and 3 bits respectively the angular component of the measurement will be aligned to the LSW of the latch in each case. As a result of this all three encoders will return the same value for the same angle.

5.1.2 Power fail and parity detection

The DEVA001 can be programmed to read an additional bit after the last data bit. This bit may be treated as a parity bit or a power fail bit by the DEVA001. In the case of parity the DEVA001 performs a parity check and sets a status bit if the data fails

the test. In the case of power failure the extra bit is latched and may be read from the status register. These features may be independently selected for each channel.

5.1.3 Gray to binary conversion

The DEVA001 can perform Gray to binary conversion on incoming encoder data in hardware. This feature may be independently selected for each channel.

5.1.4 Programmable interval timer

The DEVA001 has an 8 bit interval timer which can be programmed in steps of 0.1ms up to a maximum of 25.6ms. This timer may be used to generate interrupts or start encoder readings.

5.1.5 Time Stamper

The time stamper is a 16 bit high speed timer with a resolution of 1us. It can be programmed to latch its value on an event so to provide an accurate time stamp at which the event occurred.

5.1.6 Digital IO

The DEVA001 has 48 digital IO lines arranged in 3 groups of 16 bits. The first group of 16 bits may be associated with special input and output functions specific to other features of the card and has hardwired directions. The direction of the other 2 groups may be programmed in blocks of 8 bits.

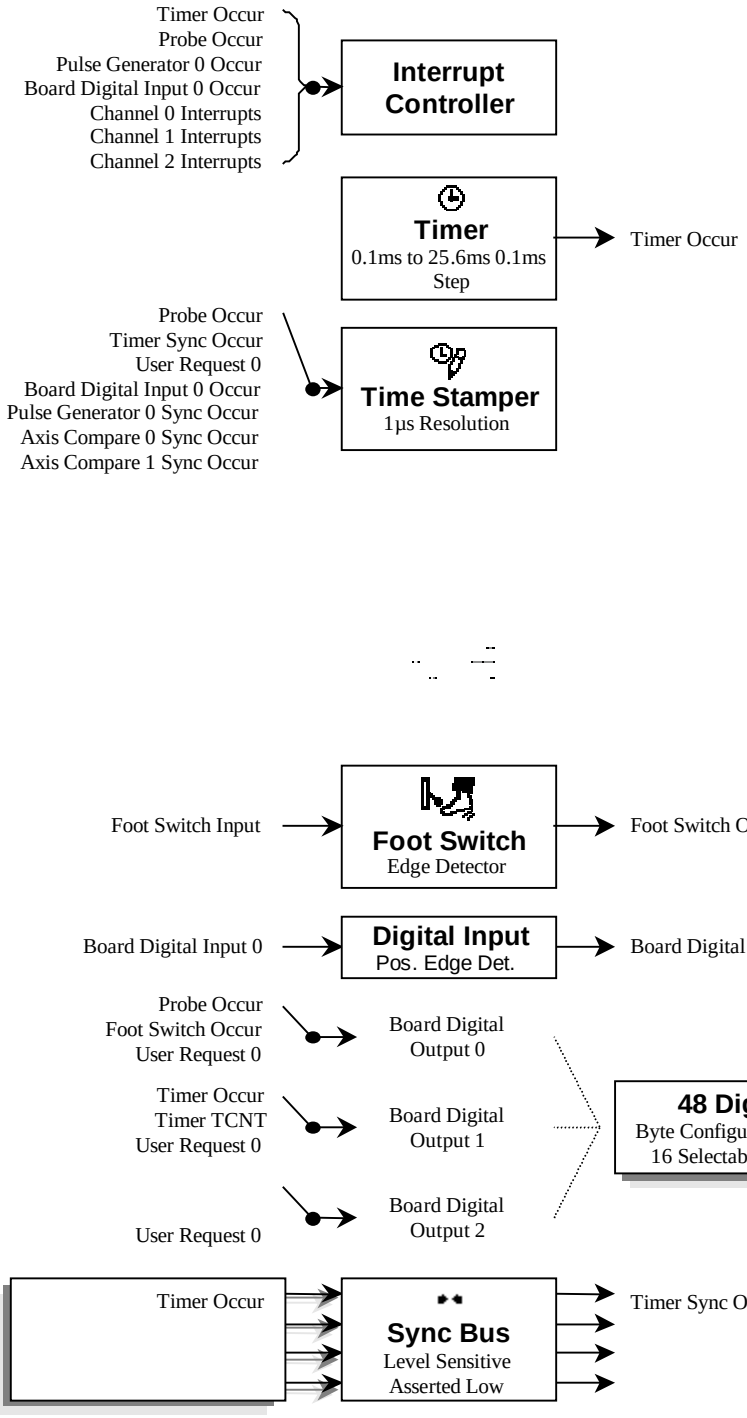
5.1.7 Event System

Many features of the card produce events. When an event occurs a flag called the occur flag is set. The occur flag may be polled by the user to establish that the event has occurred.

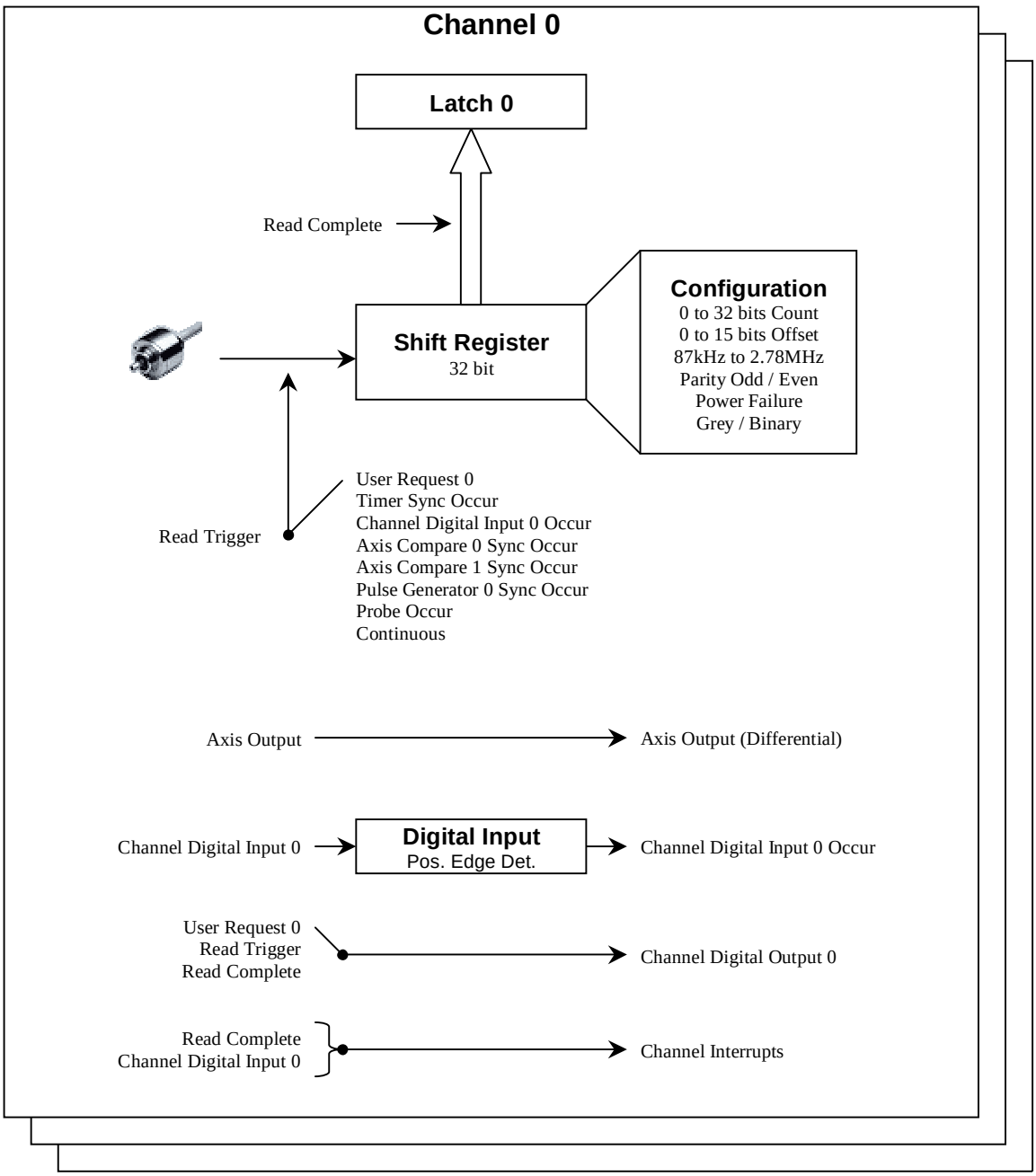
The card has a variety of actions which may be configured to respond to events. The actions include triggering an encoder position read, generating an interrupt, triggering a digital output or latching a time stamp of the event.

The following diagrams show the board level and channel level events and actions which may be programmed.

5.1.7.1 SSI board level block diagram

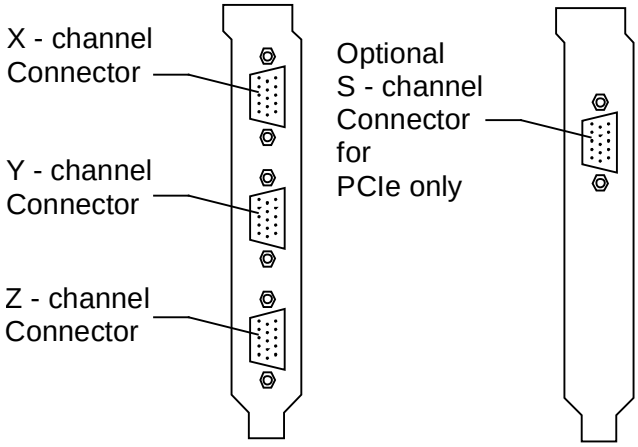


5.1.7.2 SSI channel level block diagram



5.2 Connection details

The DEVA001 has three input channels for SSI data signals from absolute encoders, designated x, y & z. Connection of each channel to the outside world is made through a 15 way D-type connector. The designation of the connectors is shown in the diagram below.



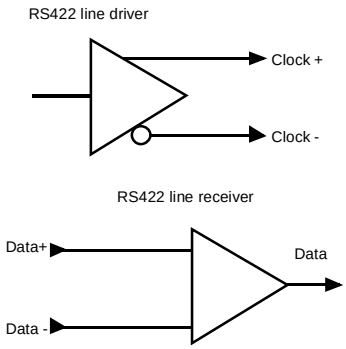
The PCIe card has an additional 4th channel. This is available on the 16 pin header labelled "S-CHAN". An additional ribbon cable with 15 way HD connector and PC bracket shown above is available upon request.

5.2.1 Encoder input connections

Connections to these sockets should be made with reference to the following pin-out table and simplified circuit shown below.

Pin Number	Signal	Function
1	**Clock+**	**Clock+** output
2	**Out+**	**Out+** output
3	**Data+**	**Data+** input
4	**Spare+**	**Spare+** output
5		
6	**Clock-**	**Clock-** output
7	**Out-**	**Out-** output
8	**Data-**	**Data-** input
9	**Spare-**	**Spare-** output
10		
11	+12V	+12 volts supply
12	+5V	+5 volt supply
13	0V	0 volt common
14		
15	-12V	-12 volt supply*****

*****PCI only.



Note: Do not connect the 15-way D-type plug from a VGA monitor into one of the encoder input channels, as damage may result.

5.2.2 SSI signal descriptions

The SSI interface uses two differential signals clock and data which have the connections **Clock+**, **Clock-**, **Data+** and **Data-.**

The DEVA001 has a third uncommitted signal designated Out with the connections **Out+** and **Out-** which may be controlled to select encoder dependent features such as forward/reverse. The **Spare+** and **Spare-** connections should be left unconnected.

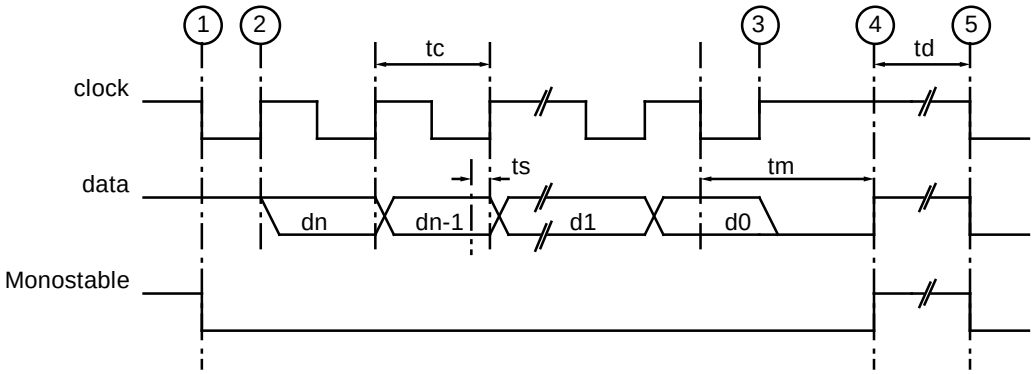
The Deva001 interrogates the encoder for its positional value by sending a pulse train from its clock output. The number of clock pulses depends on the bit length of the encoder.

The first high-low transition at point 1 triggers the encoders monostable and parallel data is stored in the encoders parallel to serial converter. While the monostable is at logic 0 no more parallel data can be stored in the parallel to serial converter.

At the first low-high transition at point 2 the most significant data bit is output by the encoder and received on the data input.

At each subsequent low-high transition in the pulse train, the next highest bit is transmitted. The pulses continuously re-trigger the monostable so that its output stays at logic zero, preventing further storage of data. When the least significant bit is

received by the DEVA001, the pulse train is terminated. The monostable is no longer triggered and at point 4 after an interval "tm" the output returns to logic 1, allowing the storage of new parallel data in the parallel to serial converter.



Register value	Clock frequency	Clock period (tc)	Setup time (ts)
0	2.78 MHz	0.36 uS	0.1 uS
1	1.39 MHz	0.72 uS	0.1 uS
2	926 KHz	1.08 uS	0.1 uS
3	694 KHz	1.44 uS	0.1 us
4	463 KHz	2.16 uS	0.1 uS
5	347 KHz	2.88 uS	0.1 uS
6	174 KHz	5.76 uS	0.1 uS
7	86.8 KHz	11.5 uS	0.1 uS

Under normal circumstances the encoder data should go to logic 0 after point 3, returning to logic 1 at point 4 indicating the encoder is ready to be read again. The DEVA001 waits for the data to return to logic 1 before it starts any subsequent readings.

If the data does not go to logic 0 at point 3 the DEVA001 assumes that synchronisation with the encoder has been lost at some time before or during the reading. The DEVA001 now outputs up-to 32 clock pulses until a logic 0 is detected, at this point the clock is stopped and the DEVA001 waits for the encoder mono-stable to time out.

This re-synchronisation process makes possible the continuous mode of operation which starts a new reading as soon as the encoder indicates it is ready without the side effect that could otherwise cause the system to lock up in an un-synchronised state in the presence of noise.

The +/-12V and +5V power supply pins may be used to supply the dc power requirements for the encoders. –12V is not available from the PCIe variant.

5.2.3 Digital IO connections

There are three 20 pin headers labelled IO0, IO1 and IO2 which provide access to 3 groups of 16 bits of digital IO. The IO conforms to 5v TTL levels, Voh min 2.4v at –8ma and Vil maximum 0.4 at 12ma.

5.2.3.1 Digital IO connector 0 connections

Pin	Connector IO0	Special Function
1	IO0 / UR0 / RTRIGx / RCOx	User request 0 / Read trigger x / Read complete occur x
2	IO1 / UR0 / RTRIGy / RCOy	User request 0 / Read trigger y / Read complete occur y
3	IO2 / UR0 / RTRIGz / RCOz	User request 0 / Read trigger z / Read complete occur z
4	IO3 / UR0 / RTRIGz / RCOs	User request 0 / Read trigger s / Read complete occur s
5	IO4 / UR0	User request 0
6	IO5 / TMRO / TMRTCNT / UR0	Timer occur / Timer terminal count / User request 0
7	IO6 / - / - / UR0	User request 0
8	IO7 / - / - / UR0	User request 0
9	IO8 / CDI0x	Channel x digital input 0
10	IO9 / CDI0y	Channel y digital input 0
11	IO10 / CDI0z	Channel z digital input 0
12	IO11 / CDI0s	Channel s digital input 0
13	IO12 / BDI0	Board digital input 0
14	IO13	
15	IO14	
16	IO15	
17	GND	
18	GND	
19	+5v	
20	+12v	

5.2.3.2 Digital IO connectors 1 and 2 connections

Pin	Connector IO1	Connector IO2
1	IO16	IO32
2	IO17	IO33
3	IO18	IO34
4	IO19	IO35
5	IO20	IO36
6	IO21	IO37
7	IO22	IO38
8	IO23	IO39
9	IO24	IO40
10	IO25	IO41
11	IO26	IO42
12	IO27	IO43
13	IO28	IO44
14	IO29	IO45
15	IO30	IO46
16	IO31	IO47
17	GND	GND
18	GND	GND
19	+5v	+5v
20	+12v	+12v

5.2.4 Sync connector

The sync connector provides a method of linking cards in hardware to allow events to be routed from one card to another. In general a ribbon cable connecting all 10 connections between all cards is required.

Pin Number	Function
1	Sync0 (Timer occur)
2	GND
3	Sync1 (Undefined)
4	GND
5	Sync2 (Undefined)
6	GND
7	Sync3 (Undefined)
8	GND
9	GND
10	GND

5.3 Direct hardware programming

The DEVA001 incremental encoder interface card is supplied with a variety of device drivers for Microsoft Windows operating systems which perform all low level access functions required for its operation. Using a device driver offers several benefits

including not having to read any more of this manual and the ability to re-use application software and routines with any of DEVA Electronics compatible products. The device drivers and the common software interface are described in section 3. For applications where hardware access is essential, the following sections give an overview of the register set and card functionality.

5.3.1 Register address map

The card implements several 32 bit read/write registers which are grouped by function and appear at different offsets within the memory space allocated to the card by the plug and play bios / operating system.

For PCIe the 32 bit registers appear in the lower 32 bits of 64 bit locations and hence all byte offsets are multiplied by 2.

Offset PCI	Offset PCIe	Read Function	Write Function
00h	00h	**Special Function status / control register** SFS_{31} — 16 | SFR_{15} — 0	**Special Function clear / control register** SFC_{31} — 16 | SFR_{15} —
04h	08h	**Interrupt request / mask register** IRR_{31} — 16 | IMR_{15} — 0	**Interrupt mask register** | IMR_{15} —
08h	10h	**Timer counter / latch** $TMRC_{31}$ — 16 | | $TMRL_7$ – 0	**Timer latch** | $TMRL_7$
0Ch	18h	**Time Stamper counter / latch** $TSTC_{31}$ — 16 | $TSTL_{15}$ — 0	
10h	20h	| |	| |
20h	40h	**Digital Output bus 0 mux / I/O bus 0 status** DOM_{31} — 16 | $DIOS_{15}$ — 0	**Digital Output bus 0 mux / control** DOM_{31} — 16 | | DOC_7
24h	48h	**Digital I/O bus 1 direction / status** | $DIOD_{17}$ – 16 | $DIOS_{15}$ — 0	**Digital I/O bus 1 control** | $DIOC_{15}$ —
28h	50h	**Digital I/O bus 2 direction / status** | $DIOD_{17}$ – 16 | $DIOS_{15}$ — 0	**Digital I/O bus 2 control** | $DIOC_{15}$ —
40h	80h	**X Channel status / control register** CSR_{31} — 16 | CCR_{15} — 0	**X Channel clear / control register** | CCR_{15} —
44h	88h	**Y Channel status / control register** CSR_{31} — 16 | CCR_{15} — 0	**Y Channel clear / control register** | CCR_{15} —
48h	90h	**Z Channel status / control register** CSR_{31} — 16 | CCR_{15} — 0	**Z Channel clear / control register** | CCR_{15} —
	A0h	**S Channel status / control register** CSR_{31} — 16 | CCR_{15} — 0	**S Channel clear / control register** | CCR_{15} —
60h	C0h	**X Latch** D_{31} — 0	
64h	C8h	**Y Latch** D_{31} — 0	
68h	D0h	**Z Latch** D_{31} — 0	
	D8h	**S Latch** D_{31} — 0	
C0h	180h	**X Channel control register 1** | CCR_{15} — 0	**X Channel control register 1** | CCR_{15} —
C4h	188h	**Y Channel control register 1** | CCR_{15} — 0	**Y Channel control register 1** | CCR_{15} —
C8h	190h	**Z Channel control register 1** | CCR_{15} — 0	**Z Channel control register 1** | CCR_{15} —
	198h	**S Channel control register 1** | CCR_{15} — 0	**S Channel control register 1** |

5.3.1.1 Special function control register – PCI 00h, PCIe 00h

The lower 16 bits of the register at offset 00h are designated the Special function control register. This read/write register allows control of board based facilities. The register contents are zero after system reset.

Special function control register:

Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
UR0	-	-	-	-	-	-	DI0ie	TSTs[2:0]			TMRe	-	Be	-	-

TMRe	Enable / Disable timer function
TSTs	Time stamper event select register
DI0ie	Enable / Disable board digital input 0 interrupt
UR0	Enable / Disable user request 0 flag

Time stamper event sources:

TSTs	Source
1	Timer Sync occur flag
2	User Request 0 flag
3	Board Digital Input 0 occur flag
4	Pulse Generator 0 Sync occur flag

5.3.1.2 Special function clear register – PCI 00h, PCIe 00h

The upper 16 bits of the register at offset 00h are designated the Special function clear register. This write-only register can be used to clear occur flags of board-based facilities by writing '1' to the equivalent bit. The status of all relevant flags is cleared after system reset.

Special function clear register:

Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16
-	-	-	-	-	-	-	DI0c	-	-	TSTc	TMRc	-	-	-	-

TMRc	Clear timer occur flag.
TSTc	Clear time stamper overflow occur flag.
DI0c	Clear board digital input 0 occur flag.

5.3.1.3 Special function status register – PCI 00h, PCIe 00h

The upper 16 bits of the register at offset 00h are designated the Special function status register. This read only register allows access to board based status information. These bits are treated as 'don't cares' when writing to this offset.

Special function status register:

Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16
-	-	-	-	-	-	-	DI0o	-	-	TSTo	TMRo	-	-	-	-

TMRo	Timer occur flag
TSTo	Time Stamper overflow occur flag
DI0o	Board Digital Input 0 occur flag

5.3.1.4 Interrupt mask register – PCI 04h, PCIe 08h

This 16 bit read / write register sets the interrupt masks to select which interrupt sources generate interrupts. A logic zero disables an interrupt and a logic one enables an interrupt. The register holds zero after system reset. The bit assignment for this register is as follows:

Interrupt mask:

Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
-	-	-	-	CH3ie	CH2ie	CH1ie	CH0ie	-	-	-	-	-	-	BRDie	TMRie

Bit	**Description**	**Source**
TMRie	Timer interrupt enable	Timer (reload) occur flag
BRDie	Board interrupts enable	Board occur flags (Probe, Board Digital Input 0)
CH0ie	Channel 0 interrupts enable	Channel 0 occur flags (Read complete, Channel Digital Input 0)
CH1ie	Channel 1 interrupts enable	Channel 1 occur flags (Read complete, Channel Digital Input 0)
CH2ie	Channel 2 interrupts enable	Channel 2 occur flags (Read complete, Channel Digital Input 0)
CH3ie	Channel 3 interrupts enable	Channel 3 occur flags (Read complete, Channel Digital Input 0)

5.3.1.5 Interrupt request register – PCI 04h, PCIe 08h

This 16 bit read-only register indicates which interrupt sources have generated interrupts. Logic one indicates that an interrupt has occurred. To clear an interrupt request, the occurred flags of all associated interrupt sources must be cleared. The register holds 00h after system reset. Please note that since the PCI/PCIE interrupt system is level sensitive, all interrupt requests must be cleared to free the interrupt line allocated the device. The bit assignment for this register is as follows:

Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16
-	-	-	-	CH3io	CH2io	CH1io	CH0io	-	-	-	-	-	-	BRDio	TMRio

TMRio	Timer interrupt occurred
BRDio	Board interrupt(s) occurred
CH0io	Channel 0 interrupt(s) occurred
CH1io	Channel 1 interrupt(s) occurred
CH2io	Channel 2 interrupt(s) occurred
CH3io	Channel 3 interrupt(s) occurred

5.3.1.6 Timer latch – PCI 08h, PCIe 10h

The lower 8 bits of the register at offset 08h are designated the Timer latch. This read / write register specifies the reload value (in 0.1ms) of the Timer function. This register holds FFh after system reset.

5.3.1.7 Timer counter – PCI 08h, PCIe 10h

The upper 8 bits of the register at offset 08h are designated the Timer counter. This read-only register returns the current value of the (count-down) timer counter (in 0.1ms). When the timer function is disabled, the timer counter resets to the value of the timer latch.

5.3.1.8 Time Stamper latch – PCI 0Ch, PCIe 18h

The lower 16 bits of the register at offset 0Ch are designated the Time Stamper latch. This read-only register allows access to the timestamp of the event associated with the Time Stamper. The register holds zero after system reset.

5.3.1.9 Time Stamper counter – PCI 0Ch, PCIe 18h

The upper 16 bits of the register at offset 0Ch are designated the Time Stamper counter. This read-only register returns the current value of the (count-up) time stamper counter (in 1us).

5.3.1.10 Digital I/O control / status register – PCI 20h, 24h, 28h, PCIe 40h,48h,50h

The lower 16 bits of these registers are designated the Digital I/O control / status registers. These read / write registers allow control of the digital outputs and access to the digital inputs of the equivalent I/O bus. The register contents are zero after system reset.

Please note that on Digital I/O bus 0, I/Os 0 to 7 are hardwired as outputs and I/Os 8 to 15 are hardwired as inputs.

5.3.1.11 Digital I/O bus 0 mux register – PCI 20h, PCIe 40h

The upper 16 bits of this register are designated the Digital I/O bus 0 mux register. This read / write register allows control of the multiplexers (source selectors) of digital outputs on I/O bus 0. For each digital output there are 2 selection bits, allowing for 4 possible signal sources. For a list of possible source routings on each pin, please refer to section 5.2.3.1.

Digital I/O bus 0 mux register:

Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16
DO7s[1:0]		DO6s[1:0]		DO5s[1:0]		DO4s[1:0]		DO3s[1:0]		DO2s[1:0]		DO1s[1:0]		DO0s[1:0]	

5.3.1.12 Digital I/O direction register – PCI 24h, 28h, PCIe 48h,50h

The upper 16 bits of this register are designated the Digital I/O direction registers. These read / write registers control the direction of digital I/Os.

Digital I/O direction register:

Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16
-	-	-	-	-	-	-	-	-	-	-	-	-	-	IOb1d	IOb0d

IOb0d	Direction of digital I/O byte 0 (1=Input, 0=Output)
IOb1d	Direction of digital I/O byte 1 (1=Input, 0=Output)

5.3.1.13 Channel control register 0– PCI 40h, 44h, 48h, PCIe 80h,88h,90h,98h

The lower 16 bits of these registers are designated Channel control register 0. These read / write registers are used to configure the data acquisition mode (such as clock frequency and bit length) from the SSI encoder. There is a register for each channel. The registers hold zero after a system reset.

Channel control register 0:

Bit 15	Bit 14	Bit 13 - Bit 12	Bit 11 - Bit 8	Bit 7 - Bit 3	Bit 2 - Bit 0
Out Output	Gray / Binary	Extra bit mode	Offset	Data length	Frequency
0 = Out Low 1 = Out High	0 = Binary mode 1 = Gray mode	0 = No Extra bit 1 = Power Fail bit 2 = Even Parity bit 3 = Odd Parity bit	0 to 15 bits	1 to 31 bits 0 = 32 bits	0 = 2.78 MHz 1 = 1.39 MHz 2 = 926 KHz 3 = 694 KHz 4 = 463 KHz 5 = 347 KHz 6 = 174 KHz 7 = 86.8 KHz

5.3.1.14Channel status register 0 – PCI 40h, 44h, 48h, PCIe 80h,88h,90h,98h

The upper 16 bits of these registers are designated Channel status register 0. These read-only registers allow access to channel based status information. These bits are treated as 'don't cares' when writing to these registers.

Channel status register 0:

Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16
-	-	-	-	-	-	-	-	-	-	-	DI0o	RCo	RARF	PWRF	Di

Di	Direct SSI data channel input
PWRF	Power failure (or no encoder) flag
PARF	Parity failure flag
RCo	Read complete occur flag
DI0o	Channel digital input 0 occur flag

5.3.1.15Channel control register 1 – PCI C0h,C4h,C8h, PCIe 180h,188h,190h,198h

The lower 16 bits of these registers are designated Channel control register 1. These read / write registers allow control of the SSI read trigger and other channel based facilities. The registers hold zero after a system reset.

Channel control register 1:

Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
-	-	-	-	-	-	-	-	-	DI0ie	RCie	DI0e	RCe	RTs[2:0]		

RTs	Read trigger event select register
RCe	Enable / Disable read complete function
DI0e	Enable / Disable channel digital input 0 function
RCie	Enable / Disable read complete interrupt
DI0ie	Enable / Disable channel digital input 0 interrupt

Read trigger events:

RTs	Source
0	User Request 0 flag
1	Timer Sync occur flag
2	Channel Digital Input 0 occur flag
7	N/A (Continuous)

52 Woodside Business Park
Birkenhead
Wirral CH41 1EL
United Kingdom

Tel +44 (0)151 647 3222
Fax +44 (0)151 647 4511

Email: support@deva.co.uk

Web : www.deva.co.uk