EX1200-7500

64-CHANNEL, 2.5 MHZ DIGITAL INPUT/OUTPUT

USER'S MANUAL

P/N: 82-0127-005
Released October 8, 2010
Revised February 26, 2013

VTI Instruments Corp.

2031 Main Street
Irvine, CA 92614-6509
(949) 955-1894

TABLE OF CONTENTS

CERTIFICATION

VTI Instruments Corp. (VTI) certifies that this product met its published specifications at the time of shipment from the factory. VTI further certifies that its calibration measurements are traceable to the United States National Institute of Standards and Technology (formerly National Bureau of Standards), to the extent allowed by that organization's calibration facility, and to the calibration facilities of other International Standards Organization members. Note that the contents of this document are subject to change without notice.

WARRANTY

The product referred to herein is warranted against defects in material and workmanship for a period of one year from the receipt date of the product at customer's facility. The sole and exclusive remedy for breach of any warranty concerning these goods shall be repair or replacement of defective parts, or a refund of the purchase price, to be determined at the option of VTI.

For warranty service or repair, this product must be returned to a VTI Instruments authorized service center. The product shall be shipped prepaid to VTI and VTI shall prepay all returns of the product to the buyer. However, the buyer shall pay all shipping charges, duties, and taxes for products returned to VTI from another country.

VTI warrants that its software and firmware designated by VTI for use with a product will execute its programming when properly installed on that product. VTI does not however warrant that the operation of the product, or software, or firmware will be uninterrupted or error free.

LIMITATION OF WARRANTY

The warranty shall not apply to defects resulting from improper or inadequate maintenance by the buyer, buyer-supplied products or interfacing, unauthorized modification or misuse, operation outside the environmental specifications for the product, or improper site preparation or maintenance.

VTI Instruments Corp. shall not be liable for injury to property other than the goods themselves. Other than the limited warranty stated above, VTI Instruments Corp. makes no other warranties, express or implied, with respect to the quality of product beyond the description of the goods on the face of the contract. VTI specifically disclaims the implied warranties of merchantability and fitness for a particular purpose.

TRADEMARKS

Java Runtime Environment™ are trademarks or registered trademarks of Sun Microsystems, Inc. or its subsidiaries in the United States and other countries. LabVIEW™ and LabWindows/CVI™ are trademarks of National Instruments Corporation. Visual Basic®, Windows®, and Internet Explorer® are registered trademarks of the Microsoft Corporation or its subsidiaries. Linux® is a registered trademark of the Linux Foundation. IVI™ is a trademark of the IVI Foundation. Bonjour™ is a trademark of Apple, Inc.

RESTRICTED RIGHTS LEGEND

Use, duplication, or disclosure by the Government is subject to restrictions as set forth in subdivision (b)(3)(ii) of the Rights in Technical Data and Computer Software clause in DFARS 252.227-7013.

VTI Instruments Corp.
2031 Main Street
Irvine, CA 92614-6509 U.S.A.

GENERAL SAFETY INSTRUCTIONS

Review the following safety precautions to avoid bodily injury and/or damage to the product. These precautions must be observed during all phases of operation or service of this product. Failure to comply with these precautions, or with specific warnings elsewhere in this manual, violates safety standards of design, manufacture, and intended use of the product. Note that this product contains no user serviceable parts or spare parts.

Service should only be performed by qualified personnel. Disconnect all power before servicing.

TERMS AND SYMBOLS

These terms may appear in this manual:

WARNING Indicates that a procedure or condition may cause bodily injury or death.

CAUTION Indicates that a procedure or condition could possibly cause damage to equipment or loss of data.

These symbols may appear on the product:

ATTENTION - Important safety instructions

Frame or chassis ground

Indicates that the product was manufactured after August 13, 2005. This mark is placed in accordance with *EN 50419, Marking of electrical and electronic equipment in accordance with Article 11(2) of Directive 2002/96/EC (WEEE).* End-of-life product can be returned to VTI by obtaining an RMA number. Fees for take-back and recycling will apply if not prohibited by national law.

WARNINGS

Follow these precautions to avoid injury or damage to the product:

Use Proper Power Cord To avoid hazard, only use the power cord specified for this product.

Use Proper Power Source To avoid electrical overload, electric shock, or fire hazard, do not use a power source that applies other than the specified voltage.

The mains outlet that is used to power the equipment must be within 3 meters of the device and shall be easily accessible.

Power Consumption Prior to using EX1200 series plug-in modules, it is imperative that the power consumption of all modules that will be installed in the mainframe be calculated for all power supply rails. The required information can be found in Appendix B of the EX1200 Series User's Manual (P/N: 82-0127-000). ***Failure to do so may result in damaging the instrument and/or the mainframe***.

WARNINGS (CONT.)

Avoid Electric Shock

To avoid electric shock or fire hazard, do not operate this product with the covers removed. Do not connect or disconnect any cable, probes, test leads, etc. while they are connected to a voltage source. Remove all power and unplug unit before performing any service. ***Service should only be performed by qualified personnel.***

Ground the Product

This product is grounded through the grounding conductor of the power cord. To avoid electric shock, the grounding conductor must be connected to earth ground.

Operating Conditions

To avoid injury, electric shock or fire hazard:

- Do not operate in wet or damp conditions.
- Do not operate in an explosive atmosphere.
- Operate or store only in specified temperature range.
- Provide proper clearance for product ventilation to prevent overheating.
- DO NOT operate if any damage to this product is suspected. ***Product should be inspected or serviced only by qualified personnel.***

Improper Use

The operator of this instrument is advised that if the equipment is used in a manner not specified in this manual, the protection provided by the equipment may be impaired.
Conformity is checked by inspection.

SUPPORT RESOURCES

Support resources for this product are available on the Internet and at VTI Instruments customer support centers.

VTI Instruments Corp.
World Headquarters

VTI Instruments Corp.
2031 Main Street
Irvine, CA 92614-6509

Phone: (949) 955-1894
Fax: (949) 955-3041

VTI Instruments
Cleveland Instrument Division

5425 Warner Road
Suite 13
Valley View, OH 44125

Phone: (216) 447-8950
Fax: (216) 447-8951

VTI Instruments
Lake Stevens Instrument Division

3216 Wetmore Avenue, Suite 1
Everett, WA 98201

Phone: (949) 955-1894
Fax: (949) 955-3041

VTI Instruments, Pvt. Ltd.
Bangalore Instrument Division

642, 80 Feet Road
Koramangala IV Block
Bangalore – 560 034
India

Phone: +91 80 4040 7900
Phone: +91 80 4162 0200
Fax: +91 80 4170 0200

Technical Support

Phone: (949) 955-1894
Fax: (949) 955-3041
E-mail: support@vtiinstruments.com

Visit http://www.vtiinstruments.com for worldwide support sites and service plan information.

SECTION 1

INTRODUCTION

FEATURES

The EX1200-7500 is a high-performance I/O module with eight ports of 8 bits (64 channels). Each 8-bit port may be configured as an input or output under program control. The I/O may be either single buffered to provide real time data access or double buffered to provide synchronized data. As part of the EX1200 series, up to 384 channels can be accommodated in a 1U full-rack mainframe, or combined with other plug-ins to configure a measure and control subsystem.

The EX1200-7500 has the flexibility to source the input and output clocks from the front panel, allowing very large numbers of channels to be synchronized to collect or present data to a UUT. Additionally, input data can be time-stamped to IEEE 1588 precision as part of the EX1200 scan engine. Deep on-board memory (up to 2 MB) can be used to generate patterns on output channels at rates up to 2.5 MHz. In order to ease overall system cabling, all clamping diodes and open collector channels are pulled up internally, rather than on a per-channel basis.

Each channel can sink 300 mA, and includes built-in clamping diodes, making this module ideal for driving and sensing external devices such as relays.

EX1200-7500 SPECIFICATIONS

GENERAL SPECIFICATIONS	
DATA INPUT CHARACTERISTICS	
V_{IN} (high)	> 40% of Vclamp
V_{IN} (low)	< 16% of Vclamp
V_{IN} (max)	60 V
DATA OUTPUT CHARACTERISTICS	
V_{OUT} (high)	> 2 V to 60 V
V_{OUT} (low)	< 1.5 V @ 300 mA
VOLTAGE RANGE	
Internal voltage source	±3.3 V, ±5.0 V, ±12.0 V, and ±24.0 V
User	> 2 V up to 60 V
MODES	
Immediate	Inputs and outputs read and written via software control
Asynchronous	Channels are latched into memory via external clock
Pattern	Buffered pattern generation and acquisition controlled by internal or external clock
GATE (PATTERN MODE)	
	Programmable active low or high
CHANNEL CONFIGURATION (PATTERN MODE)	
	32 inputs, 32 outputs
MEMORY DEPTH	
Output or input enabled	2 MB
Output and input enabled	1 MB
MAXIMUM EXTERNAL CLOCK RATE	
Pattern gen. disabled	2.5 MHz
Pattern gen. enabled	2 MHz
MAXIMUM PATTERN UPDATE RATE	
Pattern gen. disabled	2.5 MHz
Pattern gen. enabled	2 MHz
DATA INPUT CLOCK SOURCES	
	Internal clock, front panel input
POWER CONSUMPTION	
3.3 V	0.260 A
5 V	0.450 A
24 V	0.0240 A

EX1200-7500 Accessories

Connector Information	
Strain Relief Bracket Kit (Includes Connector)	
VTI Part Number	70-0363-504 (recommended accessory)
Strain Relief Bracket Kit (Without Connector)	
VTI Part Number	70-0363-503
Crimp Pin	
VTI Part Number	52-0109-000 (includes 100 crimp pins)
Manufacturer/Part Number	ERNI 234064
Mating Connector	
VTI Part Number	27-0088-160 (one per board)
Manufacturer/Part Number	ERNI 024070
Crimp Pin	
VTI Part Number	27-0088-000
Manufacturer/Part Number	ERNI 014729
Crimp Tool (DIN)	
VTI Part Number	46-0010-000
Manufacturer/Part Number	ERNI 014374
Extraction Tool (DIN)	
VTI Part Number	46-0011-000
Manufacturer/Part Number	ERNI 471555
Terminal Block Information	
Description	EX1200-TB160SE, single-ended module
VTI Part Number	70-0367-005
Unterminated Cable Assembly	
Description	160-pin, unterminated cable assembly, 3 ft
VTI Part Number	70-0363-505

SECTION 2

USING THE INSTRUMENT

UNPACKING

When an EX1200-7500 is unpacked from its shipping carton, the contents should include the following items:

- An EX1200-7500
- *LXI Quick Start Guide*
- *EX1200-7500 User's Manual* (this manual)
- *EX1200-7500 IVI, Linux, or LabView Driver* (included on Distribution CD)

All components should be immediately inspected for damage upon receipt of the unit. ESD precautions should be observed while unpacking and installing the instrument into an EX1200 series mainframe.

DETERMINE SYSTEM POWER REQUIREMENTS

The power requirements of the EX1200-7500 is provided in the *Specifications* section of *Section 1*. It is imperative that the EX1200 mainframe provides adequate power for the modules installed. For more information on EX1200 mainframe power consumption, please refer *Appendix B* of the *EX1200 Series User's Manual* (P/N: 82-0127-000). The user should confirm that the power budget for the system (for the chassis and all modules installed therein) is not exceeded on any voltage line.

It should be noted that if the mainframe cannot provide adequate power to the module, the instrument might not perform to specification and possibly damage the power supply. In addition, if adequate cooling is not provided, the reliability of the instrument will be jeopardized and permanent damage may occur. Damage found to have occurred due to inadequate cooling will void the warranty on the instrument in question.

PLUG-IN MODULE INSTALLATION

Before installing a plug-in module into an EX1200 system, make sure that the mainframe is powered down. Insert the module into the base unit by orienting the module so that the metal cover of the module can be inserted into the slot of the base unit. Position the cover so that it fits into the module's slot groove. Once the module is properly aligned, push the module back and firmly insert it into the backplane connector. See Figure 2-1 for guidance.

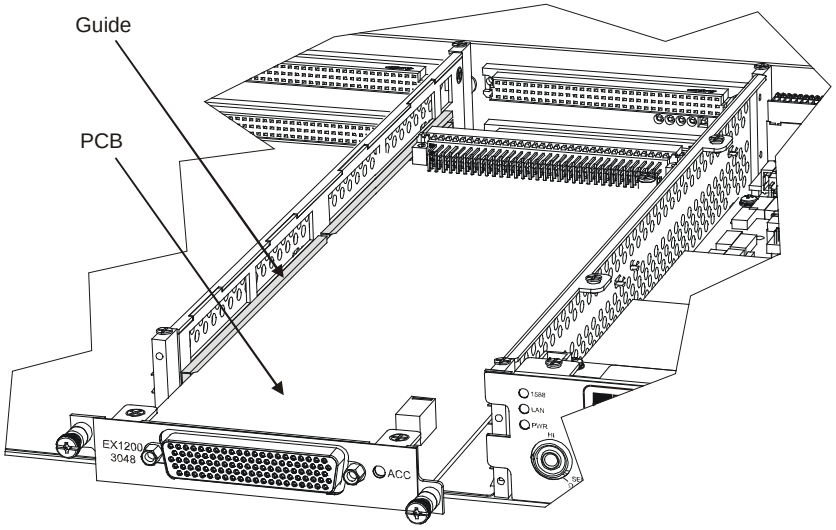


FIGURE 2-1: MODULE INSTALLATION (EX1200-3048 USED AS EXAMPLE)

MAXIMIZING MEASUREMENT PERFORMANCE

This section discusses tips and procedures that can help maximize the actual performance realized with the EX1200-7500 and aid the user in avoiding some common pitfalls associated with making measurements.

Warm-up Time

As the EX1200-7500 doesn't require calibration and there are no adjustments for the instrument to meet its published specifications, there is no warm up required.

Voltage Outputs

A channel output is pulled to the configured voltage through a pull-up resistor. The actual output that will be seen by the unit under test is defined by the standard voltage divider equation:

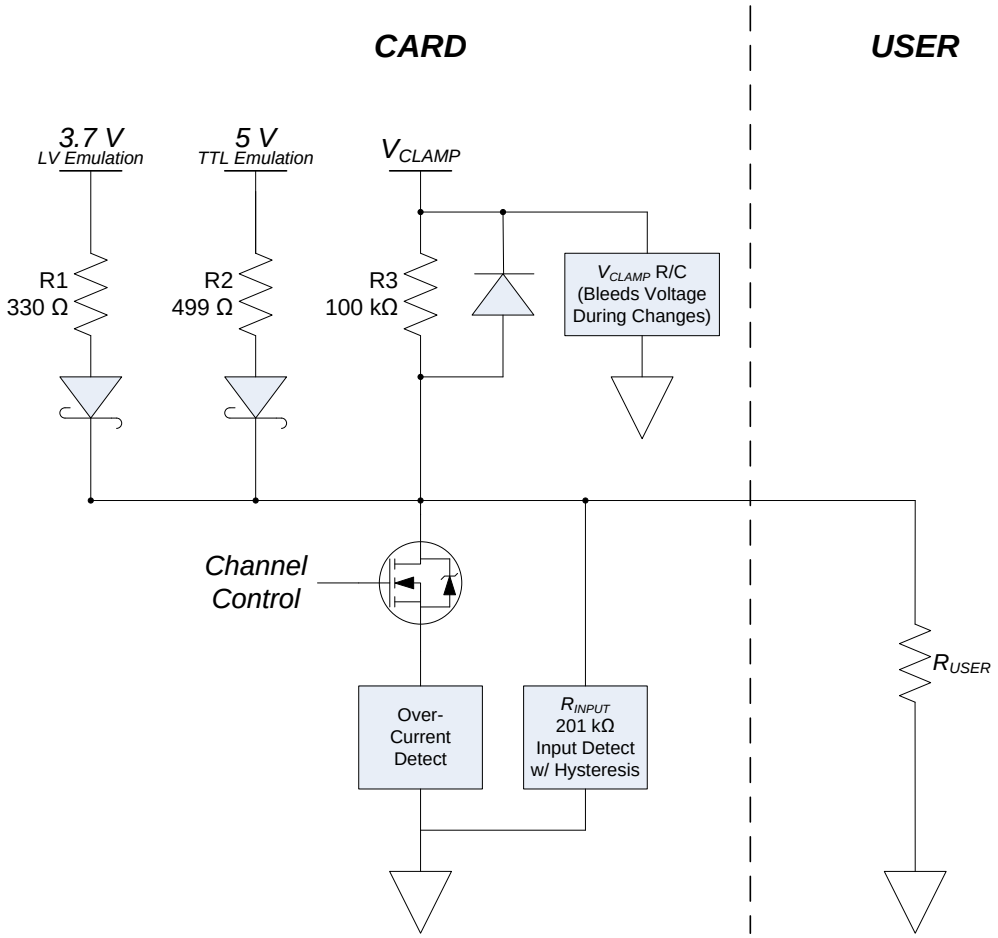
$$V_0 = V_{CLAMP}\left(\frac{R_2}{R_1+R_2}\right)$$

Where,

V_0 is the output voltage
V_{CLAMP} is the user configured voltage
R_1 is the internal pull-up resistor
R_2 is the external load presented by the unit under test

In normal mode, R_1 is 100 kΩ. In TTL emulation mode, R_1 is 350 Ω, so the pull-up to V_{CLAMP} is much stronger. TTL emulation mode is available for the 3.3 V and 5 V voltage levels.

If voltages higher than those provided by the equation above are desired, the user can supply a voltage with an external power supply in conjunction with a low-value pull-up resistor to get closer to the user's V_{CLAMP} voltage. The 300 mA sink capability of the channel still must be observed.



Voltage Output			
Mode	**Internal Supply**	**Pull-Up (R_X)**	***Open Circuit V_0***
Normal	24 V	100 kΩ (R_3)	~15.84 V
Normal	12 V	100 kΩ (R_3)	~7.92 V
Normal	5 V	100 kΩ (R_3)	~3.3 V
Normal	3.3 V	100 kΩ (R_3)	~2.47 V
TTL Emulation	5 V*	499 Ω (R_2)	~4.64 V
LV Emulation	3.7 V*	330 Ω (R_1)	~3.34 V
**For TTL and LV modes, there is approximately a 0.35 V drop due to the Schottky diode, resulting in an actual internal supply voltage of 4.65 V and 3.35V, respectively.*			
Mode	**External Supply**	**Load**	***$V_{O\text{-}USER}$***
Normal	V_{CLAMP}	R_{USER}	$V_{CLAMP}\left[\frac{R_{USER} \parallel R_{INPUT}}{100k\Omega + \left(R_{USER} \parallel R_{INPUT}\right)}\right]$

FIGURE 2-2: VOLTAGE OUTPUT BLOCK DIAGRAM

CONNECTOR PIN/SIGNAL ASSIGNMENT

The connector pins and their signal assignments are shown in Table 2-1 and Figure 2-1 below. For mating connector and accessory information, please see the *EX1200-7500 Specifications*.

Pin	Signal	Pin	Signal	Pin	Signal	Pin	Signal	Pin	Signal
A1	GND	**B1**	GND_C	**C1**	GND	**D1**	GND	**E1**	GND
A2	GND	**B2**	GND	**C2**	GND	**D2**	GND	**E2**	GND
A3	DATA1.1	**B3**	GND	**C3**	DATA2.1	**D3**	GND	**E3**	DATA7.3
A4	DATA1.2	**B4**	GND	**C4**	DATA2.2	**D4**	GND	**E4**	DATA7.4
A5	DATA1.3	**B5**	GND	**C5**	DATA2.3	**D5**	GND	**E5**	DATA7.5
A6	DATA1.4	**B6**	DATA7.1	**C6**	DATA2.4	**D6**	GND	**E6**	DATA7.6
A7	GND	**B7**	GND	**C7**	GND	**D7**	DATA7.2	**E7**	GND
A8	GND	**B8**	GND	**C8**	GND	**D8**	GND	**E8**	DATA7.7
A9	DATA1.5	**B9**	USER_V1	**C9**	DATA2.5	**D9**	GND	**E9**	DATA7.8
A10	DATA1.6	**B10**	GND	**C10**	DATA2.6	**D10**	USER_V7	**E10**	GND
A11	DATA1.7	**B11**	GND	**C11**	DATA2.7	**D11**	GND	**E11**	DATA8.1
A12	DATA1.8	**B12**	GND	**C12**	DATA2.8	**D12**	GND	**E12**	DATA8.2
A13	GND	**B13**	USER_V2	**C13**	GND	**D13**	EXT_CLK	**E13**	DATA8.3
A14	GND	**B14**	GND	**C14**	GND	**D14**	GND	**E14**	DATA8.4
A15	DATA3.1	**B15**	GND	**C15**	DATA4.1	**D15**	GND	**E15**	GND
A16	DATA3.2	**B16**	USER_V3	**C16**	DATA4.2	**D16**	GND	**E16**	GND
A17	DATA3.3	**B17**	GND	**C17**	DATA4.3	**D17**	USER_V8	**E17**	DATA8.5
A18	DATA3.4	**B18**	GND	**C18**	DATA4.4	**D18**	GND	**E18**	DATA8.6
A19	GND	**B19**	GND	**C19**	GND	**D19**	GND	**E19**	DATA8.7
A20	GND	**B20**	USER_V4	**C20**	GND	**D20**	GND	**E20**	DATA8.8
A21	DATA3.5	**B21**	GND	**C21**	DATA4.5	**D21**	EXT_GATE_GEN	**E21**	GND
A22	DATA3.6	**B22**	GND	**C22**	DATA4.6	**D22**	GND	**E22**	GND
A23	DATA3.7	**B23**	GND	**C23**	DATA4.7	**D23**	GND	**E23**	DATA6.1
A24	DATA3.8	**B24**	USER_V5	**C24**	DATA4.8	**D24**	USER_V6	**E24**	DATA6.2
A25	GND	**B25**	GND	**C25**	GND	**D25**	GND	**E25**	DATA6.3
A26	GND	**B26**	GND	**C26**	GND	**D26**	EXT_GATE_ACQ	**E26**	DATA6.4
A27	DATA5.1	**B27**	GND	**C27**	DATA5.5	**D27**	GND	**E27**	GND
A28	DATA5.2	**B28**	GND	**C28**	DATA5.6	**D28**	GND	**E28**	GND
A29	DATA5.3	**B29**	GND	**C29**	DATA5.7	**D29**	GND	**E29**	DATA6.5
A30	DATA5.4	**B30**	GND	**C30**	DATA5.8	**D30**	GND	**E30**	DATA6.6
A31	GND	**B31**	GND_C	**C31**	GND	**D31**	GND	**E31**	GND
A32	DATA6.8	**B32**	GND	**C32**	DATA6.7	**D32**	GND	**E32**	GND

TABLE 2-1: EX1200-7500 CONNECTOR PIN SIGNAL ASSIGNMENT

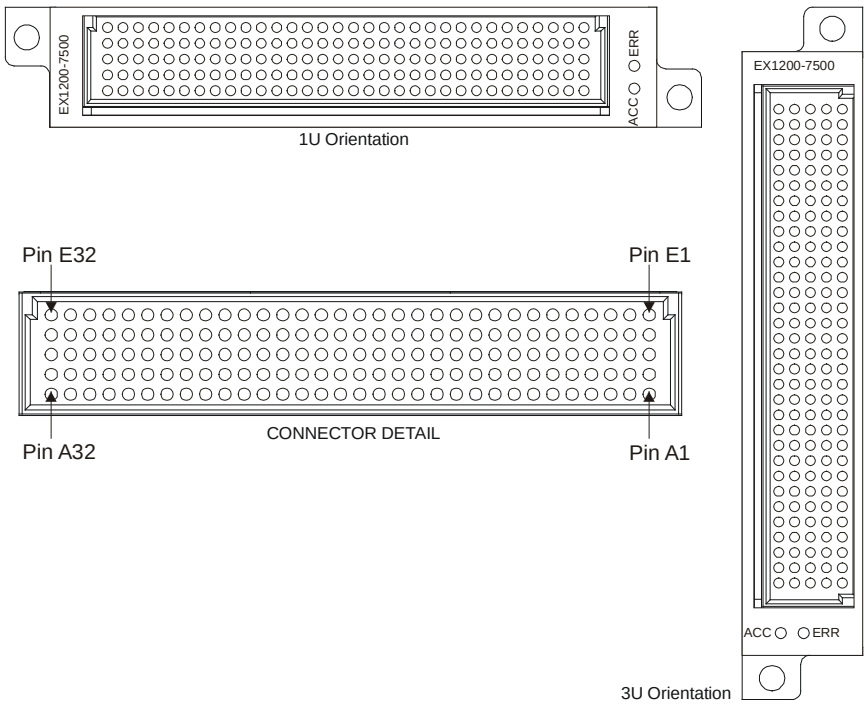


FIGURE 2-3: EX1200-7500 FRONT PANEL DETAIL

Front Panel Connector Pins Description

Pin Name	Description
EXT_CLK	EXT_CLK is a bi directional signal. Based on Pattern mode clock source, this signal will act as input or output. As an output, the EXT_CLK signal outputs to the front panel during Pattern generation and acquisition when the Pattern mode clock source for is “internal”. This output is also controlled by the “Clock Output Enable” bit. As an input, the signal is used as input for Pattern generation and acquisition when the Pattern mode clock source is “External” and is “Trigger Source” during Asynchronous mode..
EXT_GATE_ACQ	This signal is an input to the EX1200-7500. Based on the trigger source selected, this signal is used during pattern acquisition as a External sync signal.
EXT_GATE_GEN	Another EX1200-7500 input that, based on the trigger source selected, is used during pattern generation as a External sync signal.

EX1200-TB160SE Terminal Block

VTI offers a single-ended terminal block for the EX1200-7500 (P/N: 70-0367-005). The terminal block simplifies cabling by providing screw-terminal blocks for user wiring. Signal pin mapping for the EX1200-7500 can be seen in Table 2-2.

TB Ref	Signal	Conn Pin	TB Ref	Signal	Conn Pin	TB Ref	Signal	Conn Pin	TB Ref	Signal	Conn Pin
T1	GND_D	E2	T41	DATA3.1	A15	T81	GND_D	C14	T121	DATA3.6	A22
T2	GND_D	D2	T42	GND_D	A14	T82	GND_D	B14	T122	DATA3.7	A23
T3	DATA7.3	E3	T43	GND_D	A13	T83	DATA4.1	C15	T123	GND_D	A20
T4	GND_D	D3	T44	DATA1.8	A12	T84	GND_D	B15	T124	DATA3.4	A18
T5	DATA7.4	E4	T45	DATA1.6	A10	T85	DATA4.2	C16	T125	GND_D	A19
T6	GND_D	D4	T46	USER_V7	D10	T86	USER_V3	B16	T126	DATA3.2	A16
T7	DATA7.5	E5	T47	GND_D	E10	T87	DATA4.3	C17	T127	DATA3.3	A17
T8	GND_D	D5	T48	DATA2.7	C11	T88	GND_D	B17	T128	GND_D	D32
T9	DATA7.6	E6	T49	GND_D	B11	T89	DATA4.4	C18	T129	DATA6.6	E30
T10	GND_D	D6	T50	DATA1.7	A11	T90	GND_D	B18	T130	GND_D	D30
T11	GND_D	E7	T51	GND_D	B12	T91	GND_D	C19	T131	DATA6.5	E29
T12	DATA7.2	D7	T52	DATA2.8	C12	T92	GND_D	B19	T132	GND_D	D29
T13	DATA7.7	E8	T53	USER_V2	B13	T93	GND_D	C20	T133	GND_D	E31
T14	GND_D	D8	T54	GND_D	C13	T94	USER_V4	B20	T134	GND_D	D31
T15	DATA7.8	E9	T55	DATA1.5	A9	T95	DATA6.7	C32	T135	DATA5.8	C30
T16	GND_D	D9	T56	GND_D	A8	T96	GND_D	E1	T136	GND_D	B30
T17	DATA2.1	C3	T57	GND_D	A7	T97	DATA6.4	E26	T137	GND_D	C31
T18	GND_D	B3	T58	DATA1.4	A6	T98	EXT_GATE_ACQ	D26	T138	GND_C	B31
T19	GND_D	C2	T59	DATA.1.3	A5	T99	DATA6.2	E24	T139	GND_D	B29
T20	GND_D	B2	T60	DATA1.2	A4	T100	USER_V6	D24	T140	DATA5.7	C29
T21	DATA2.2	C4	T61	DATA1.1	A3	T101	GND_D	E27	T141	GND_D	B28
T22	GND_D	B4	T62	GND_D	A2	T102	GND_D	D27	T142	DATA5.6	C28
T23	DATA2.3	C5	T63	GND_D	B10	T103	GND_D	E28	T143	GND_D	C25
T24	GND_D	B5	T64	DATA2.6	C10	T104	GND_D	D28	T144	GND_D	B25
T25	DATA2.4	C6	T65	GND_D	E15	T105	DATA6.1	E23	T145	GND_D	B27
T26	DATA7.1	B6	T66	GND_D	D15	T106	GND_D	D23	T146	DATA5.5	C27
T27	GND_D	C7	T67	GND_D	E16	T107	GND_D	E22	T147	GND_D	B26

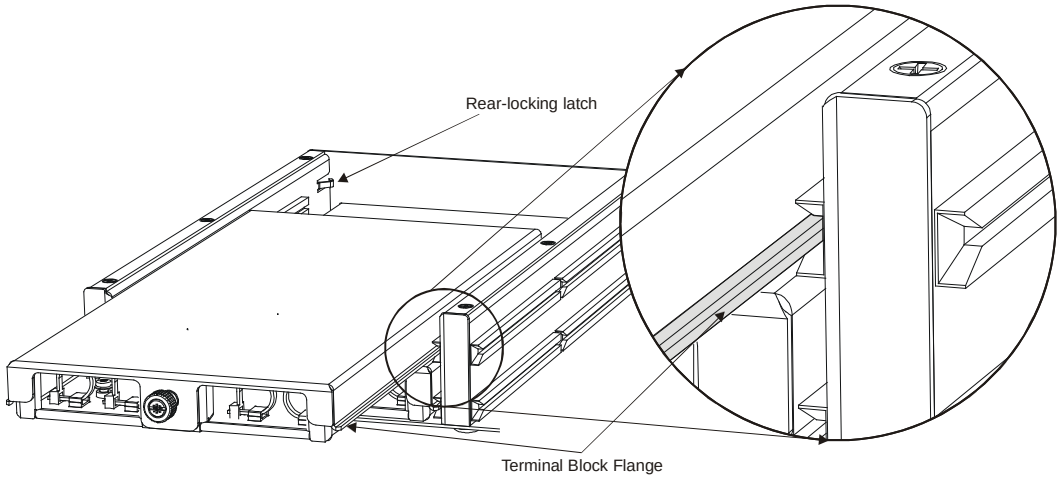
			7								
T28	GND_D	B7	T68	GND_D	D16	T108	GND_D	D22	T148	GND_D	C26
T29	GND_D	C8	T69	DATA8.5	E17	T109	GND_D	D25	T149	GND_D	A26
T30	GND_D	B8	T70	USER_V8	D17	T110	DATA6.3	E25	T150	DATA5.1	A27
T31	DATA2.5	C9	T71	DATA8.6	E18	T111	DATA8.8	E20	T151	DATA5.2	A28
T32	USER_V1	B9	T72	GND_D	D18	T112	GND_D	D20	T152	DATA4.3	A29
T33	DATA8.1	E11	T73	DATA8.7	E19	T113	DATA4.6	C22	T153	DATA5.4	A30
T34	GND_D	D11	T74	GND_D	D19	T114	GND_D	B22	T154	GND_D	A31
T35	GND_D	D12	T75	DATA4.5	C21	T115	DATA4.7	C23	T155	GND_D	E32
T36	DATA8.2	E12	T76	GND_D	B21	T116	GND_D	B23	T156	GND_C	B1
T37	EXT_CLK	D13	T77	GND_D	D1	T117	DATA4.8	C24	T157	DATA6.8	A32
T38	DATA8.3	E13	T78	DATA3.5	A21	T118	USER_V5	B24	T158	GND_D	A1
T39	GND_D	D14	T79	EXT_GATE_GEN	D21	T119	DATA3.8	A24	T159	GND_D	B32
T40	DATA8.4	E14	T80	GND_D	E21	T120	GND_D	A25	T160	GND_D	C1

TABLE 2-2: EX1200-7500 TO EX1200-160TB PIN AND SIGNAL MAPPING

Terminal Block Receiver

The EX1200-TBR chassis is a 1U receiver capable of housing six terminal blocks. The EX1200-TBR ships with rubber feet for table top installations, but may be fitted with rackmount ears for installation into a test rack (P/N: 70-0367-010).

To install a terminal block into the EX1200-TBR, insert the flanges on the side of the terminal block into the guide rails of the desired slot. Continue to push the terminal block into the receiver until it is secured by the rear-locking latch of the receiver. To remove the terminal block from the EX1200-TBR, hold the center thumbscrew on the terminal block, then pull the terminal block from the receiver.



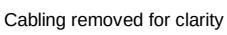


FIGURE 2-4: TERMINAL BLOCK INSTALLATION INTO THE EX1200-TBR

BPL_INSFAIL BEHAVIOR

The EX1200 platform backplane has a BPL_INSFAIL line that indicates to all modules that a severe failure has occurred. When this line is asserted, some modules put themselves into a known state, such as opening all relays. The EX1200-7500 does not respond to this signal.

SECTION 3

PROGRAMMING THE INSTRUMENT

INTRODUCTION

This section provides programming examples for the EX1200-7500. Additional information can be found in the driver help file. If the instrument will be used on a Linux system, a .chm viewer must be installed on the host PC (examples of these programs can be found at the following URL: http://www.linux.com/news/software/applications/8209-chm-viewers-for-linux.)

RELATED SOFTWARE COMPONENTS

IVI-COM Driver
IVI-C Driver
LabView Driver
Linux C++ Driver

USING THE DRIVER

The EX1200-7500 may be used in a variety of environments including: Visual Basic, C#, C++, LabView. VTI instruments provides a IVI-C and IVI-COM compliant driver as well as a shared object that can be used on Linux systems that comply with the Linux Standard Base (Version 3.1).

Here is how to use the driver in each environment:

1) **Visual Studio C++**

```
#import "IviDriverTypeLib.dll" no_namespace
#import "VTEXDio.dll" no_namespace
```

2) **C#**

Add a reference to VTEXDio.dll in the project. Include the following at the top of any code file that will access the driver:

```
using VTI.VTEXDio.Interop;
```

3) **C/C++ on Windows**

Link against VTEXDio.lib and include VTEXDio.h in the file.

4) **C++ on Linux**

Link against /opt/vti/lib/libdio.so and include all the headers in /opt/vti/include in the source file.

5) **LabView**

Copy the driver package to the <Labview>/instr.lib directory and access all relevant VIs

USING THE EX1200-7500

INITIALIZING\CLOSING THE INSTRUMENT

The base interface of the EX1200-7500 IVI driver, VTEXDio (LibDio on Linux), is used to open and close connections to the instrument as well as containing pointers to all other interfaces to access the functionality of the instrument.

Establishing a Connection

Visual Studio C++

```
#import "IviDriverTypeLib.dll" no_namespace
#import "VTEXDio.dll" no_namespace

int main()
{
  //Windows driver creation
  ::CoInitialize(NULL); //Start the COM layer
  try
  {
    IVTEXDioPtr Dio(__uuidof(VTEXDio));

    /*The driver is given an empty options string. If more than one DIO card in
      included in the mainframe, an option such as a slot number must be provided.
      This is because the DIO driver does not support more than one card per driver
      instance. Note that the reset flag is also set so that the unit is started
      clean.*/

    Dio->Initialize("TCPIP::10.20.1.5::INSTR", VARIANT_TRUE, VARIANT_TRUE, "");

    // Use the Driver
    Dio->Close();
  }
  catch(...)
  {
    // Handle any exceptions thrown
  }
  return 0;
}
```

Option Strings

The VTEX drivers provide option strings that can be used when Initializing an instrument. The option string values exist to change the behavior of the driver. The following options strings are available on VTI IVI drivers:

- **Simulate**: Allows the user to run a program without commanding switch card or instruments. This option is useful as a debugging tool.
- **Cache**: Per the IVI specification, this option "specifies whether or not to cache the value of attributes." Caching allows IVI drivers to maintain certain instrument settings to avoid sending redundant commands. The standard allows for certain values to be cached always or never. In VTI IVI-drivers, all values used are of one of these types. As such, any values entered have no effect.
- **QueryInstrumentStatus**: Queries the instrument for errors after each call is made. As implemented in the VTI IVI drivers, instruments status is always queried regardless of the value of this property.

- **DriverSetup**: Must be last, and contains the following properties:
  - **Logfile**: Allows the user to specify a file to which the driver can log calls and other data.
  - **Logmode**: Specifies the mode in which the log file is opened. The allowed modes are:
    - **w**: truncate s the file to zero length or creates a text file for writing.
    - **a**: opens the file for adding information to the end of the file. The file is created if it does not exist. The stream is positioned at the end of the file.
  - **LogLevel**: Allows the user to determine the severity of a log message by providing a level-indicator to the log entry.
  - **Slots**: This is the most commonly used option and it allows for a slot number or a slot number and a card model to be specified.
    "Slots=(2)" - Just slot 2.
    "Slots=(2=EX1200_3048)" - slot and card model
    "Slots=(2,3)" - Multiple slots
- **InterchangeCheck**: Boolean option that enables/disables IVI Interchangeability checking. As implemented in the VTI IVI drivers, values entered for this property have no effect.
- **RangeCheck**: Boolean option that enables or disables driver validation of user-submitted values. As implemented in the VTI IVI drivers, validation of user inputs is always performed at the firmware level regardless of this property's value.
- **RecordCoercions**: Boolean option that enables driver recording of coercions. As implemented in the VTI IVI drivers, coercions are handled in the firmware and cannot be recorded.

BASIC OPERATION

The EX1200-7500 has two basic modes of operation: Normal mode and Pattern mode. In **Normal** mode, the instrument's ports can be configured independently and written to/read from at any time. In **Pattern** mode, four ports are inputs and four ports are output and they are all clocked off a single source. Every time the clock ticks data is written to the output and read from the input. The `VTEXDio.Mode` property is used to select between Normal and Pattern modes.

NORMAL MODE

In Normal mode, each port can be configured individually. This is done through the `VTEXDio.Normal.Ports` interface. Each port is part of the Items and is referenced by the port name (which is a string). The Port naming is as follows:

Port Name	Port Index
PORT1	1
PORT2	2
PORT3	3
PORT4	4
PORT5	5
PORT6	6
PORT7	7
PORT8	8

Configuring Ports

Once a reference to the port has been acquired, setting the port's properties adjusts its behavior.

Port Parameters

- **Data**: This is used to either get or set the data on the selected port. Setting data on an input will have no effect. Setting data on a port with no voltage set is not allowed.
- **Direction**: Determines if the port is input or output.
- **OverCurrent**: (Read Only) Indicates if the device is in an over-current condition.

- **LatchedOverCurrent**: The `OverCurrent` property gives the current state of the over-current indication. This property indicates whether an over-current condition has occurred since the card was powered on. It can be reset with the `ResetOverCurrent` method below.
- **Polarity**: Sets the polarity of the port. If the port is an input, the user will see the readback inverted. If the port is an output, values written to the port will be inverted on writing and readback, leading to the user retrieving the same value that was written. The output will be the inverse of this value.
- **VoltageRange**: Sets the voltage range of the port to one of the supported voltage ranges listed below.
- **Voltage Source**: Sets the voltage source to **Internal** or **User**. In **User** voltage mode, an external voltage must be applied to the USER voltage pins on the front panel for proper operation of the device (see Table 2-1 for more information on User connector pins).

Port Methods

- **Configure**: Configures **Direction**, **Polarity**, **VoltageSource**, and **VoltageRange** in a single function call.
- **ResetOverCurrent**: Resets the **LatchedOverCurrent** property.

Configuring Ports

C++

```
// The voltage range to 5 V.
Dio->Normal->Ports->Item["PORT1"]->VoltageRange = 5.0;

// Cneck the over-current state
if(Dio->Normal->Ports->Item["PORT1"]->LatchedOverCurrent == 1)
{
  //The port is in an over-current state, so action should be taken here.
}

// Read the input
int data = Dio->Normal->Ports->Item["PORT1"]->Data;

// Set the polarity to inverse.
Dio->Normal->Ports->Item["PORT1"]->Polarity = VTEXDioPolarityInverse;

// Read the input again
data = Dio->Normal->Ports->Item["PORT1"]->Data;

//Set the 2nd port to be an output
Dio->Normal->Ports->Item["PORT2"]->Direction = VTEXDioDirectionOutput;
Dio->Normal->Ports->Item["PORT2"]->VoltageRange = 3.3;

// Set the highest bit
Dio->Normal->Ports->Item["PORT2"]->Data = 128;

/* Changing the Polarity of an output changes both the output and readback polarity.
This means the user will always read back what they wrote, but the opposite signal
will be placed on the line. */
Dio->Normal->Ports->Item["PORT2"]->Polarity = VTEXDioPolarityInverse;
Dio->Normal->Ports->Item["PORT2"]->Data = 128; //Looks the same as above, really
outputting 127
data = Dio->Normal->Ports->Item["PORT2"]->Data; //Will return 128.

/* The Configure call sets all of the values for a particular port quickly. Here, the
port is configured with a direction of Output, a Polarity of Normal, a VoltageSource
of User, and a voltage of 5V TTL Emulation (which has no effect, since the voltage
source is set to User. */
Dio->Normal->Ports->Item["PORT3"]->Configure(VTEXDioDirectionOutput,
      VTEXDioPolarityNormal, VTEXDioVoltageSourceUser, -1.0);

// Write to multiple ports at one time
```

```
Dio->Normal->WritePorts(VTEXDioDataWidth16, 2, 255); //Writes a data of 255 on
      port 2, and 0 on port 3.
```

TTL Emulation Mode Vs Standard Mode

The EX1200-7500 normally operates with open-collector inputs and outputs. In some cases, users may desire the card to have a stronger pull-up. If the voltage source is set to `VTEXDioTTLEmulation` or `VTEXDioLVEmulation`, a pull-up resistor of 499 Ω (TTL) or 330 Ω (LV) is respectively enabled, providing approximately 10 mA of current sourcing capability into a short at the card's outputs. This allows standard high-impedance digital inputs to see voltages close to V_{CLAMP}. The voltage at the output will depend on the voltage divider set up by the cards pull up and the load the user places on the output. See *Voltage Outputs* in *Section 2* for more detail.

Async Mode

Asynchronous, or Async mode, is an additional functionality that can be used in Normal mode by calling `Normal.InitiateAsync`. In Async mode, all of the properties normally available in Normal mode remain available and additional `Async` properties are enabled. A 2 MB data FIFO is also enabled, controllable by either an external trigger signal or a software trigger. When an asynchronous trigger is received, the device records the current state of all eight ports as well as the configurations of those ports. The asynchronous trigger can be asserted at any time and the configuration returned in the data FIFO is always be consistent.

PATTERN MODE

As described previously, each port can be read or written to at any time when in Normal mode. When in pattern mode, the ports are clocked synchronously. The card implements two independent operation modes in Pattern mode: Pattern Generation (Output) and Pattern Acquisition (Input). Either or both of these modes can be enabled at the same time.

In pattern mode, ports 1 through 4 are used as ~~inputs~~outputs, while ports 5 through 8 are ~~outputs~~inputs, regardless of the `PatternMode` property state. If either Acquisition or Generation is enabled alone, the EX1200-7500 can use 2 MB of memory for FIFO data or pattern storage. If both are enabled, the DIO divides the memory evenly, using 1 MB for FIFO data and 1 MB for pattern storage.

In Pattern mode, the ports are not individually controllable, instead, Pattern Generation controls all four of its ports simultaneously and Pattern Acquisition does the same. Polarity can still set, but, as noted above, the direction is fixed.

Global Parameters

- **PatternMode**: Determines whether the pattern generation, pattern acquisition, or both sections are enabled. When Pattern Generation is enabled, the maximum InternalClockFrequency is reduced to 2 MHz. This value is coerced to 2 MHz if the current value is above that threshold.
- **SendSoftwareTrigger**: Sends a software command to the device which triggers when the trigger source is set to Software.
- **Clock Source**: Determines whether an internal or external clock source will be used.
- **ClockOutputEnabled**: Determines if the internal pattern clock is output on the clock pin (D13) instead of this pin being an input.
- **InternalClockFrequency**: Determines the internal clock frequency. The minimum frequency for the EX1200-7500 is 38.1469 Hz while the maximum is 2.5 MHz when Pattern Generation is not enabled. Due to hardware constraints, values above 2 MHz are not accepted if 1) the EX1200-7500 is in Pattern Generation mode or 2) if both Pattern Acquisition and Pattern Generation are enabled.

Independent Parameters for Pattern Acquisition and Generation

- **Abort**: This method aborts pattern acquisition and/or generation.
- **ConfigurePorts**: This method sets some of the most frequently changed properties (i.e. polarity, gate polarity, and voltage range.)
- **GatePolarity**: This property defined the gate polarity as either being active-high or active-low.
- **Initiate**: This method starts pattern generation and/or acquisition.
- **LoopBack**: This method determines whether the acquisition FIFO stops acquiring data when it is full or whether it loops back and overwrites the oldest FIFO data.
- **Polarity**: This property sets the polarity of the port. If the port is an input, the user will see the readback inverted. If the port is an output, values written to the port will be inverted on writing and readback, leading to the user retrieving the same value that was written. The output will be the inverse of this value.
- **Triggered**: This property indicates whether the device has been triggered since the last initiate command was given.
- **TriggerSource**: This property sets the trigger source for pattern acquisition and/or generation.
- **VoltageRange**: This property set the voltage level that will be output (if port is an output) or the range the port detects on (if the port is an input). In Pattern Mode, the only allowed values are 3.3 V, 5 V, and 0 V (the default, power-on state where no inputs/outputs are allowed and will return an error if initiated).

Triggering Model

The pattern acquisition and generation modes have independent trigger models that work the same way (see Figure 3-1).

The EX1200-7500 does not generate or acquire patterns until it is in the SAMP state. When the card is first powered on, it is in the default state. In this state, the user sets up trigger parameters, patterns (if generation mode is enabled), and port configurations. To move from this state, the `Pattern.Input.Initiate()` and the `Pattern.Output.Initiate()` calls are used. Once in the INIT state, the module is waiting for a trigger. If the trigger source is **Immediate**, the EX1200-7500 immediately passes through the INIT state, otherwise, it waits for the trigger selected by the `TriggerSource` property to be asserted. Once in the TRIGD state, the module waits for the gate line to be asserted. If the `TriggerSource` is **SW**, then the TRIGD state is bypassed and the module proceeds to the SAMP state. For all other trigger types, the module waits for the external gate pin to be asserted before proceeding to the SAMP state. There is an independent gate pin for both Pattern Input (D26) and Pattern Output (D12). The gate pins are pulled high by default, so, if they are not connected, the card will move directly to the SAMP state. Once in the SAMP state, the pattern input will read the ports and store them to the FIFO on every rising edge of the clock and the pattern output will output a pattern from the FIFO on every rising edge of the clock. When the gate signal is de-asserted, the module returns to the IDLE state. The `Abort` method moves the EX1200-7500 from any state to the IDLE state. Completion of the pattern output will do the same unless looback is set to `VTEXDioFifoOverwrite`, then pattern output will never complete and must be aborted.

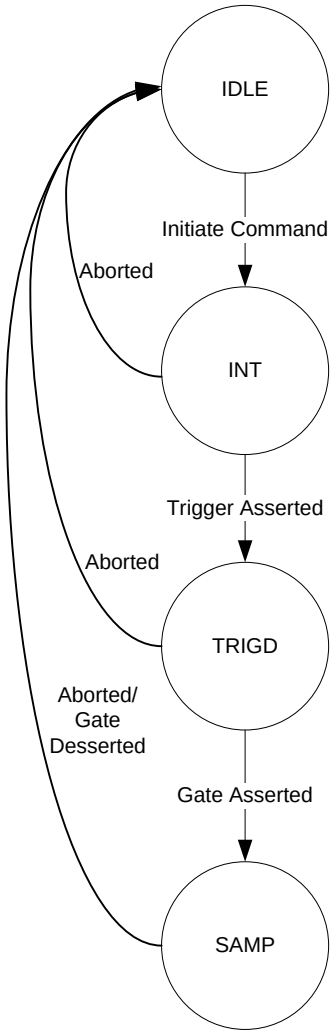


FIGURE 3-1: PATTERN MODE TRIGGER MODEL

Pattern Acquisition

When the trigger model has advanced to the SAMP state, the state of all the ports will be stored to the FIFO. There are four ports that will acquire data and 8 bits per port. The state of all the data is stored as a single 32-bit integer with the least significant bit being bit 0 of port 1 and the most significant bit being bit 7 of port 4. Reading the FIFO count property of the `VTEXDIO.Pattern.Input` interface will show the number of pages in the FIFO, where one page is the state of all four ports.

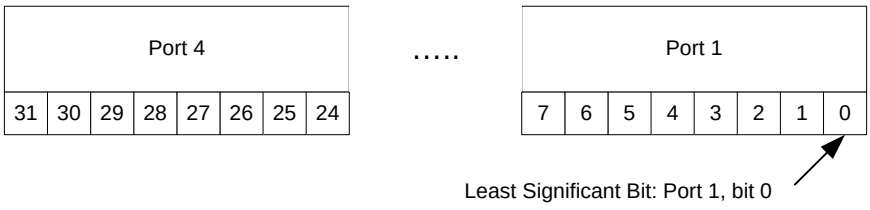


FIGURE 3-2: DATA STRUCTURE

Reading Data

- **Pattern.Input.Read()**: This method removes data from the FIFO.

Acquisition Only Functionality

- **FifoClear**: This method clears the pattern acquisition FIFO.
- **FifoCount**: This property sets the number of samples in the pattern acquisition FIFO.
- **Read**: This method reads data from the pattern acquisition FIFO.

Pattern Acquisition

C++

```
#import "IviDriverTypeLib.dll" no_namespace
#import "VTEXDio.dll" no_namespace

int main()
{
  //Windows driver creation
  ::CoInitialize(NULL); //Start the COM layer
  try
  {
     IVTEXDioPtr Dio(__uuidof(VTEXDio));

    /*We chose to give this driver an empty options string. If you have more than one
DIO card in
      your chassis, you will have to give it options such as a slot number. This is
because the DIO
      driver does not support more than one card per driver instance. Note also that
we set the reset
      flag so that we can get a clean start witht he unit */
    Dio->Initialize("TCPIP::10.20.1.5::INSTR", VARIANT_TRUE, VARIANT_TRUE, "");

   /*First we need to change the mode of the card to Pattern mode */

   Dio->Mode = VTEXDioModePattern;

   /*Now that we're in pattern mode, we can change the common pattern settings,
including the pattern clock
     and also the PatternMode property. We're going to set the latter to Pattern
Acquisition only, so that
     we don't have to share the Pattern Acquisition memory with Pattern Generation. */

   Dio->Pattern->PatternMode = VTEXDioPatternModeAcquisition;
   Dio->Pattern->ClockSource = VTEXDioClockSourceInternal;
   Dio->Pattern->InternalClockFrequency = 1.0; //This will be coerced to the lowest
frequency we support

   /* Now that we have set up the common Pattern Mode settings, we can set up Pattern
      Acquisition. As with the Normal mode settings, we can change several settings
      including Polarity and VoltageRange. Unlike Normal mode, if we change these
      settings they apply to all ports, not only one port. */

   //The Pattern Acquisition functionality only supports 3.3V and 5V non-TTL ranges,
and the special value 0.
   Dio->Pattern->Input->VoltageRange = 5.0;
   //A backplane line, the front panel pin, or software can be the trigger source
   Dio->Pattern->Input->TriggerSource = VTEXDioTriggerSourceBPL0;
   //This setting will stop the fifo from taking more data when it is full.
   Dio->Pattern->Input->LoopBack = VTEXDioFifoStop;

   /* Now that we have configured the Acquisition functionality, we need to enable it.
The Initiate
      function causes the card to enter the trigger model, and Abort forces an exit.
For now, we will
```

```
      ignore the Gate and Software Trigger functionalities as there are separate
examples for these */

   Dio->Pattern->Input->Initiate();

   // Wait for a while

   Dio->Pattern->Input->Abort();

   /* Now we've taken some data, we need to read it back from the device. First, we
create a SAFEARRAY
      type to store it. The Windows/Microsoft programming documentation for .net and
COM devices explains
      this data type in more detail. We will give the Read function arguments to get
back 100 pages of data
      with a timeout of 60 seconds*/

   SAFEARRAY* Data = NULL; //Note that we expect this array to be empty when it is
passed to Read.
   Dio->Pattern->Input->Read(100, 60000, &Data);

   Dio->Close();

   ::CoUninitialize();
  }
  catch(...)
  {
     //Error handling here
  }
  return 0;
}
```

Pattern Generation

Pattern generation is used to generate a sequence of bits synchronously. As with pattern acquisition, the device generates no output until the SAMP state is reached. When the device is in the SAMP state, it will output the next state of the selected pattern on the rising edge of each clock.

Patterns

A pattern is an array of 32-bit integers stored in memory and identified by a number chosen by the user. The property `MaxPatterns` determines how many patterns the user can store in memory. The maximum length of each pattern is given by the `MaxPatternLength` property. This value will decrease as `MaxPatterns` is increased. The maximum value of `MaxPatterns` is 512, giving the user pattern IDs of 0 through 511. Calling `Pattern.Output.CreatePattern()` creates a pattern at the specified index with an array of data provided by the user.

End State

When the device has reached the end of the pattern in memory it will either loop over the pattern again or stop output data. This is configured by the `VTEXDio.Pattern.Output.LoopBack` property. If this parameter is set to `VTEXDioFifoStop`, then the EX1200-7500 will advance to the End State. The device will also go to the End State when Abort is called. The End State is intended to be a “safe” state for the outputs.

The recommended method for setting up patterns to be generated is:

1) Set the `MaxPatterns` property. This can be from 1 to 512 in powers of 2.
2) Read the `MaxPatternLength` propery to ensure that no pattern will exceed the limit.
3) Use the `CreatePattern` method to load all patterns that will be used.
4) Use the `CurrentPattern` property to set the pattern that will be output.
5) Initiate the device.

WARNING	Do not set the **MaxPatterns** property after creating patterns. This method will erase all previously configured patterns.

Gen Only Functionality

- **CreatePattern**: Stores a pattern in the unit for execution later.
- **CurrentPattern**: Sets the pattern that will be used when initiated. This can only be modified in the IDLE state.
- **MaxPatternLength**: (Read Only) Gets the maximum number of points available for a pattern.
- **MaxPatterns**: The maximum number of patterns allowed to be loaded to the device. The lower this number is the more memory available to individual patterns. When this value is configured all current patterns are deleted. Note that this is true even if the value is reset to the previous value.
- **PatternCount**: (Read Only) The number of patterns currently loaded to the device.
- **RetrievePattern**: Reads back the pattern specified by the pattern ID.
- **GetEndState**: Gets the currently set End State.
- **SetEndState**: Sets the End State.

Using Multiple, Synchronized Instruments

Multiple EX1200-7500 cards can be synchronized in Pattern or Async modes via the simple expedient of tying their clocks together. Since in these modes data acquisition or generation are totally dependent on clocking, the cards will automatically be synchronized when the clocks are.

DRIVER INTERFACES

The following is a list of programmatic interfaces to the EX1200-7500 with a description of the functionality they provide. Please refer to the help file installed with the driver for a programming reference that includes all methods with their parameters as well as all enumerations.

- **IVTEXDio**: Used to open and close connections to the instrument and set the mode the instrument is in.
- **DriverOperation**: IVI standard. Provides control over the manner in which the driver operates.
- **Identity**: IVI standard. Provides detailed version information about the driver, connected hardware, and firmware versions.
- **Normal**: Used for Reading and Writing data to multiple ports in normal mode and configuring Async mode.
- **Ports**: Used to read and write data to a single port and to configure port state such as Voltage range and polarity.
- **Pattern**: Used to configure global pattern settings
- **Input**: Used to configure input ports and read pattern input data
- **Output**: Used to configure output ports and write pattern output data
- **Platform**: Used to Log driver function calls and to retireve the serial number of the instrument.
- **Utility**: IVI standard. Provides useful functionality not specific to this driver, such as Reset.

SECTION 4

SFP OPERATION

INTRODUCTION

EX1200s offer an embedded web page which provides network configuration control, time configuration, and the ability to perform firmware upgrades. To facilitate discovery of the mainframe, VTI provides the **L**AN **In**strument **C**onnection and **U**pgrade (**LInC-U**) utility on the *VTI Instruments Corp. Drivers and Product Manuals CD* included with the EX1200 mainframe in the *EX Platforms Requisites* directory.

To open the embedded web page, start the **LInC-U** utility by navigating to **Start → Programs → VTI Instruments Corporation → LInC-U Utility → LInC-U Utility**. Once the utility is run, LInC-U will scan the network to discover all LAN-based VTI instruments. Once the scan is complete, the **Discovery Devices** tab will appear and show the instruments that were discovered, as shown in Figure 4-1. To open the web page, click on the hostname hyperlink in the **Discover Devices** tab. The IP address of the EX1200 can also be viewed from this window as well as its firmware version.

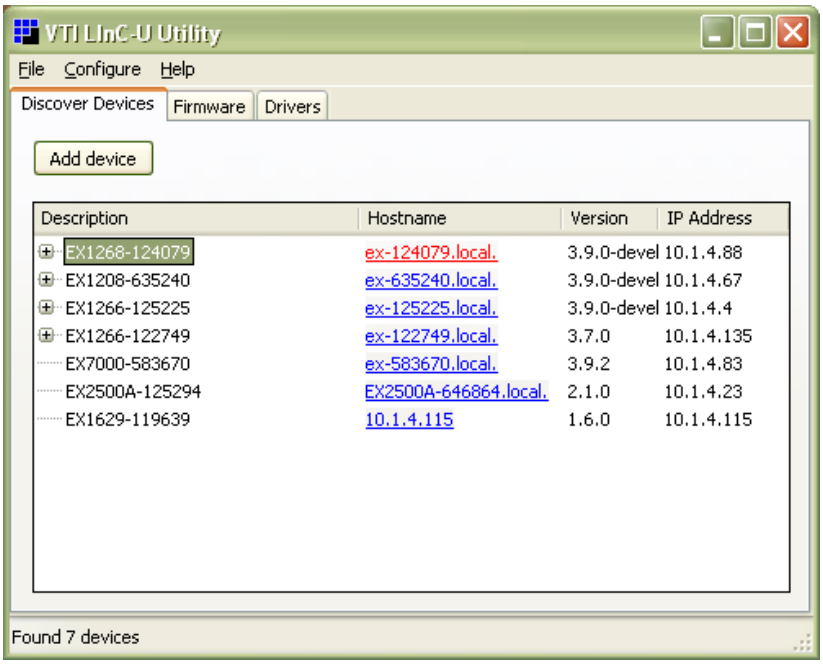


FIGURE 4-1: LINC-U DISCOVERY TAB WITH AN EX1268 SELECTED

Alternatively, the EX1200 may also be discovered using Internet Explorer's Bonjour for Windows plug-in, by entering the mainframe's IP address into the address bar of any web browser to view the embedded web page, or using VXI-11. For more information on discovery methods, refer to the *EX1200 Series User's Manual* (P/N: 82-0127-000).

GENERAL WEB PAGE OPERATION

When initial connection is made to the EX1200, the instrument home page, **Index**, appears (see Figure 4-2). This page displays instrument-specific information including:

- Model
- Manufacturer
- Serial Number
- Description
- LXI Class
- LXI Version
- Hostname
- MAC Address
- IP Address
- Netmask
- Instrument Address String
- Firmware Version
- IEEE 1588 Time

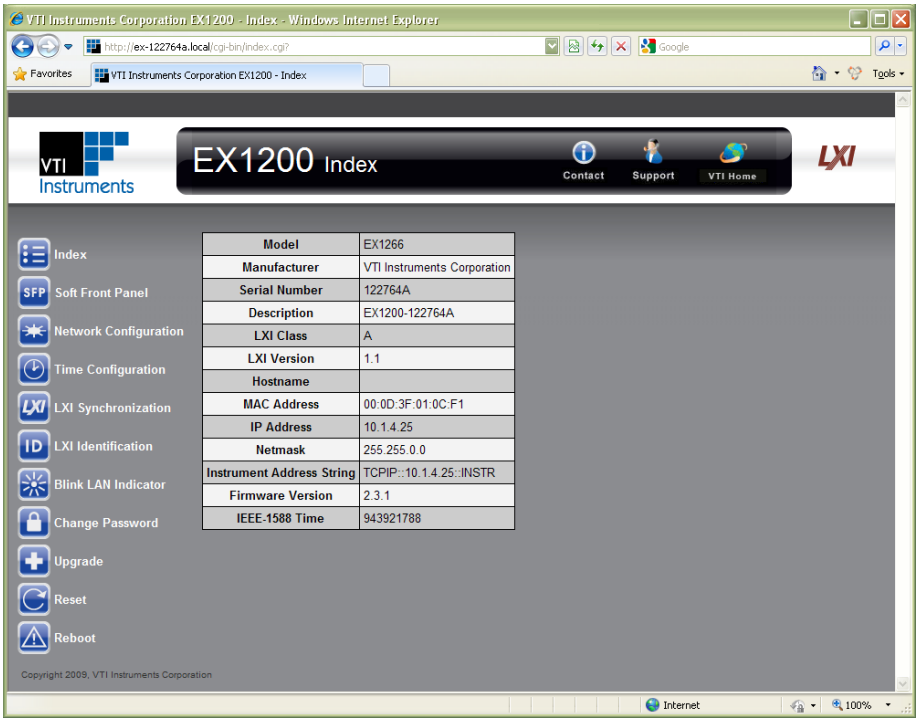


FIGURE 4-2: EX1200 MAIN WEB PAGE

The **Index** is accessible from any other instrument page by clicking on the EX1200 web page header. The EX1200 **Command Menu** is displayed on the left-hand side of every internal web page. The entries on the command menu represent three types of pages:

Status This type of page performs no action and accepts no entries. It provides operational status and information only. The **Index** page is an example of a status page.

Action This type of page initiates a command on the instrument, but does not involve parameter entry. The **Reboot** page is an example of an action page.

Entry This type of page displays and accepts changes to the configuration of the instrument. The **Time Configuration** page is an example of an entry page.

Use of the entry-type web pages in the EX1200 are governed by a common set of operational characteristics:

- Pages initially load with the currently-entered selections displayed.
- Each page contains a **Submit** button to accept newly entered changes. Leaving a page before submitting any changes has the effect of canceling the changes, leaving the instrument in its original state.
- Navigation through a parameter screen is done with the **Tab** key. The **Enter** key has the same function as clicking the **Submit** button and cannot be used for navigation.

Notes on Web Page Use

If a window needs to be resized, this should be done when the window opens. Resizing requires a refresh which causes the current state to be lost.

VTI Instruments Logo

The VTI Instruments logo that appears on the upper left of all EX1200 web pages is a link to the VTI Instruments corporate website: http://www.vtiinstruments.com.

The remainder of this discussion will focus on the EX1200-7500 soft front panel. For more information on other EX1200 soft front panel elements, please refer to the *EX1200 Series User's Manual*.

EX1200-7500 SOFT FRONT PANEL

To navigate to the EX1200-7500 soft front panel, click on **Soft Front Panel** in the **Command Menu** (see Figure 4-3). Next, select **DIO ex1200-7500** from the list of cards installed in the EX1200.

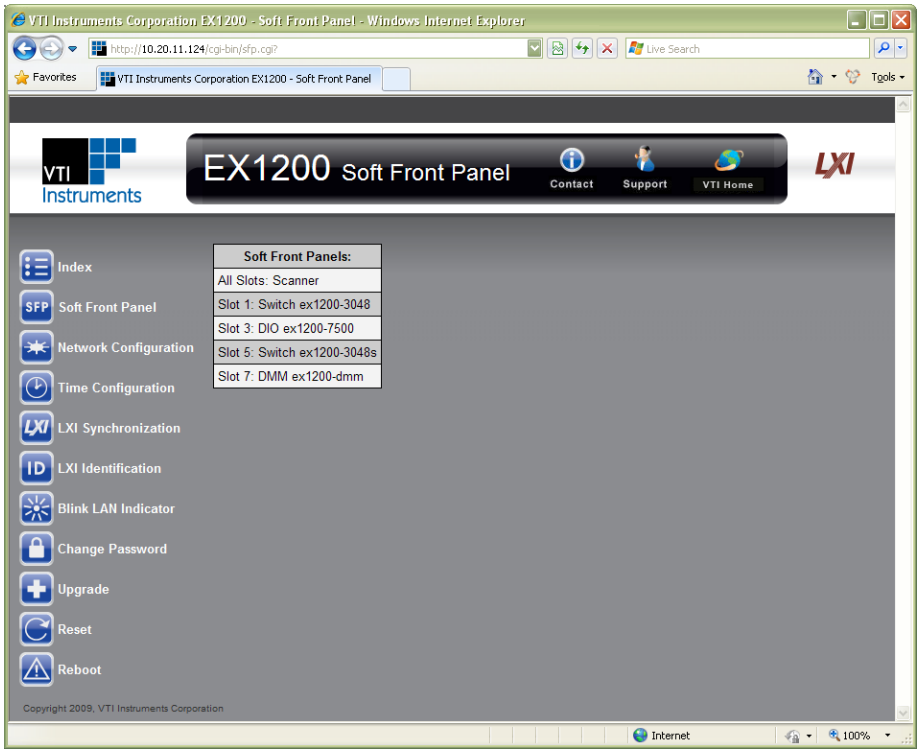


FIGURE 4-3: EX1200 SOFT FRONT PANEL MAIN PAGE

MONITOR AND CONTROL PAGE

By default, the EX1200-7500 SFP opens to the **Monitor and Control** view. From this view, the user can define the DIO's mode, its ports configurations, and allow access the data stored in FIFO. Although the SFP does not expose the entire functionality of the DIO, the SFP can be used to set up the EX1200-7500 in most applications.

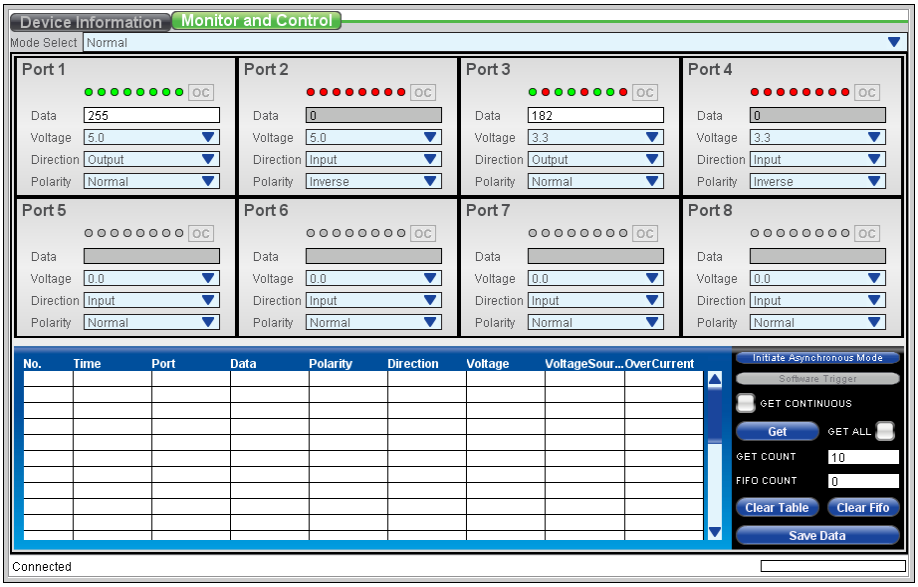


FIGURE 4-4: EX1200-7500 SOFT FRONT PANEL (NORMAL MODE)

At the top of the Monitor and Control menu, the **Mode Select** drop menu is used to set the EX1200-7500 for either **Normal** or **Pattern** mode. The settings for each mode will now be discussed.

NORMAL MODE CONFIGURATION

When normal mode is selected, the user can define each port independently. Normal mode is further discussed in *Normal Mode* in *Section 3*.

Port Configuration Fields

For each Port, the following settings can be configured.

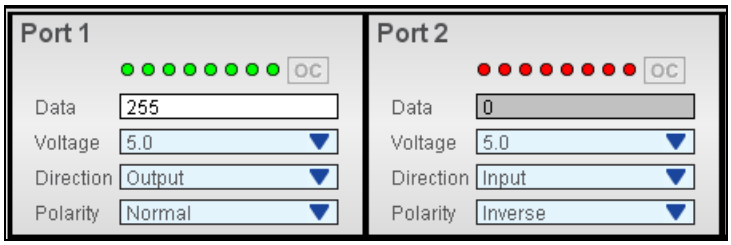


FIGURE 4-5: PORT CONFIGURATION FIELDS

- **Data Indicators**: The LED indicators in each port configuration section indicate the data value this is being sent by or received by the indicated port. The data is an 8-bit integer with the green LEDs indicating an output high (1) and a red LED indicating an output low (0).
- **OC**: If an over-current event occurs, the **OC** indicator will turn **red**, indicating that actions should be taken by the user.

- **Data**: When the direction is set to **Output**, the user can enter the data value that will be sent on the selected port.
- **Voltage**: Select the voltage range for the port. Allowable values are **-2.0**, **-1.0**, **0** (default), **3.3**, **5.0**, **12.0**, **24.0**, and **User Voltage Source**. Note that the **-2.0** and **-1.0** are not voltage values, but correlate to LV emulation and TTL emulation, respectively.
- **Direction**: Sets the port as either an **Input** or an **Output**.
- **Polarity**: Sets the ports polarity as either **Normal** or **Inverse**.

Data Acquisition Section

When data is received, it is placed in FIFO memory and remains there until read from memory. The EX1200-7500 can be configured to read the FIFO data by using the **Data Acquisition Field**, which provides the following options.

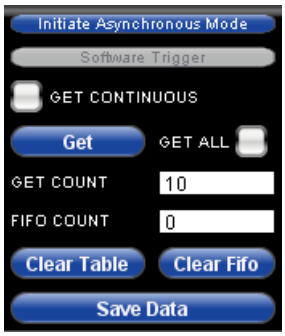


FIGURE 4-6: DATA ACQUISITION SECTION

- **Initiate/Abort Asynchronous Mode**: This modifies how data is collected by placing the DIO in Asynchronous mode (see *Async Mode* in *Section 3* for more information).
- **Software Trigger**: When in Asynchronous mode, clicking on the button will generate a software trigger that causes the current state and configuration of all eight ports to be recorded.
- **Get Continuous** checkbox: If enabled, this button disables the **Get** button and continually populates the data table as data is acquired.
- **Get** button: When clicked, data is immediately pulled from FIFO memory.
- **Get All** checkbox: When enabled, clicking the **Get** button will retrieve all of the current data. Once the FIFO is empty, it will stop acquiring retrieving data.
- **Get Count** field: Indicates the number of data points that will be returned when the **Get** button is clicked. Should be less than or equal to the **FIFO Count**.
- **FIFO Count** field: Indicates the number of data points available in the FIFO memory.
- **Clear Table** button: When clicked, all of the data currently in the table is erased.
- **Clear FIFO** button: When clicked, any data points currently stored in the FIFO memory will be erased.
- **Save Data** button: When clicked, the data in the table can be saved. A .csv file is generated by default.

Data Log

Once a read of the FIFO is initiated, the data from FIFO becomes available in the **Data Log Table** at the bottom of the SFP.

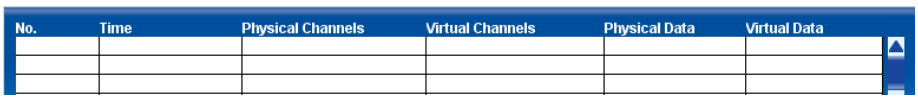
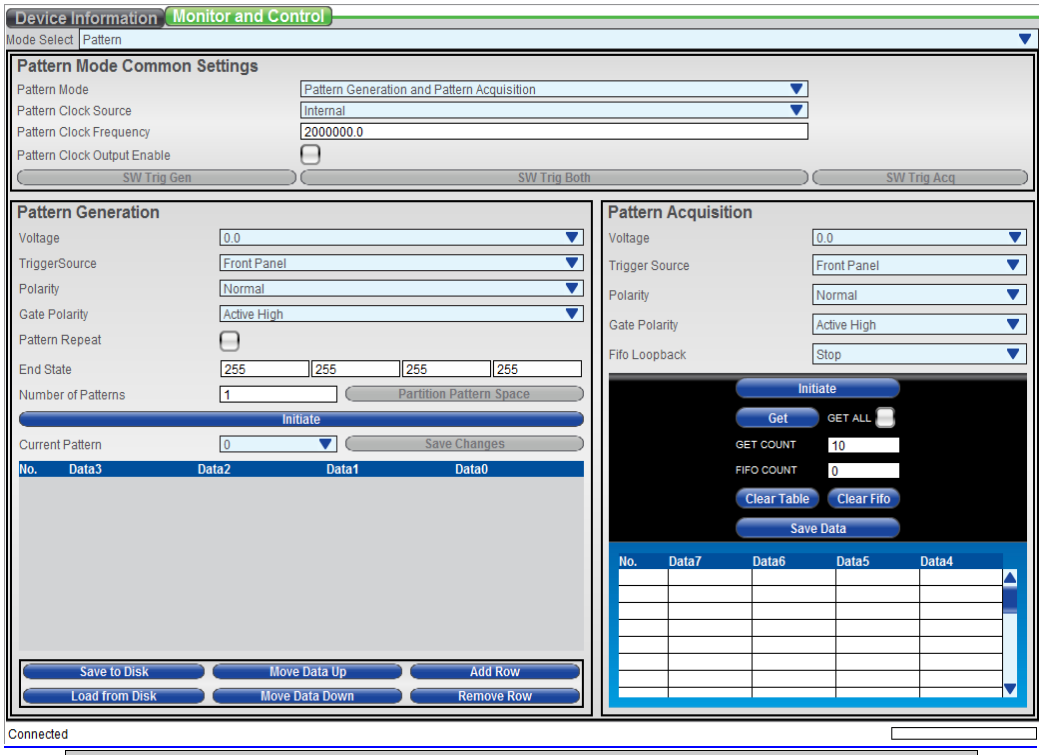


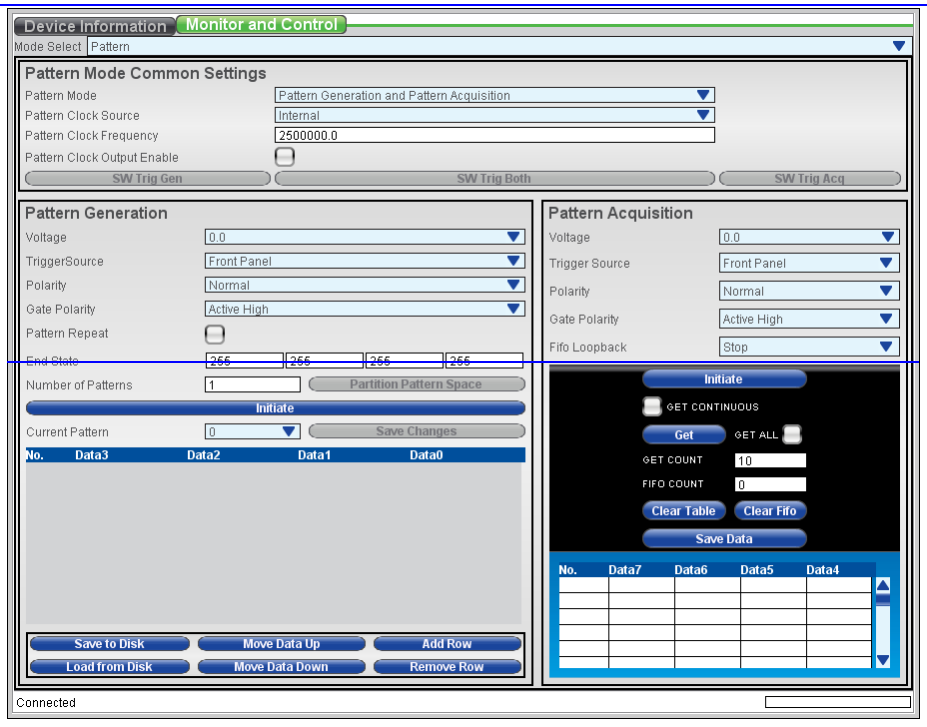
FIGURE 4-7: DATA LOG TABLE

- **No.**: Indicates the row number, for reference.
- **Time**: Indicates the IEEE 1588 time the event occurred.
- **Port**: Indicates the port the data was recorded for at the given **Time**.
- **Data**: Indicates the data was recorded on the port at the given **Time**.
- **Polarity**: Indicates whether the port was set to **Normal** or **Inverse** when data was recorded.
- **Direction**: Indicates whether the port was use as an **Input** or **Output** when data was recorded.
- **Voltage**: Indicates the voltage range port's voltage range when data was recorded.
- **VoltageSource**: Indicates whether the an **Internal** or **User** voltage source was used when data was recorded.
- **OverCurrent**: Indicates whether a over-voltage condition was present when data was recorded.

PATTERN MODE CONFIGURATION

When **Mode Select** is set to **Pattern**, the following settings become available to the user. Pattern mode is discussed in detail in *Pattern Mode* in *Section 3*.

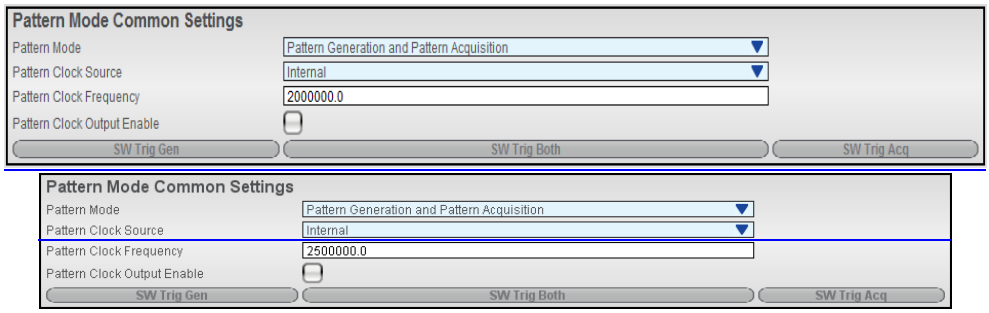




Note that the items available on this page depend on the **Pattern Mode** that is selected. When **Pattern Generation Only** is selected, the **Pattern Generation** section of the page expands to fill the whole page. This is also true when **Pattern Acquisition Only** is selected.

Pattern Mode Common Settings

In the **Pattern Mode Common Settings** section, the user defines characteristics that are common to both the pattern generation and acquisition modes.



- **Pattern Mode**: Determines whether pattern mode will only generate patterns, acquire patterns, or do both.
- **Pattern Clock Source**: Sets the clock source for pattern generation/acquisition to either **External** or **Internal**. Note that, if enabled, **Pattern Clock Output Enable** will to be cleared is selected to prevent the EXT_CLK pin (D13) from being used as both a clock input and output.
- **Pattern Clock Frequency**: When an internal clock is selected, this field become available and it defines the internal clock's frequency. For the Pattern Mode, ~~T~~the minimum frequency for the EX1200-7500 is 38.1469 while the maximum is 20~~5~~00000.0 (default).
- **Pattern Clock Output Enable**: When checked, the EX1200-7500 will output its clock from the EXT_CLK pin (D13) rather than using it as an external clock pin. Note that, if enabled, **Pattern Clock Source** is coerced to **Internal** to prevent D13 from being used as both a clock input and output.
- **SW Trig Gen** button: This button generates a software trigger for pattern generation only.
- **SW Trig Both** button: This button generates a software trigger for pattern generation and patter acquisition.
- **SW Trig Acq** button: This button generates a software trigger for pattern acquisition only.

Pattern Generation Settings

When pattern generation is enabled, the Pattern Generation section allows the user to configure how a pattern will be generated as well as configure the memory used for pattern generation.

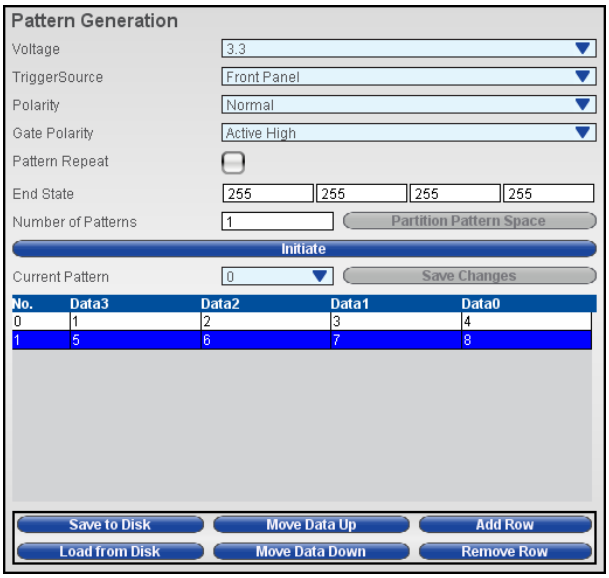


FIGURE 4-8: PATTERN GENERATION SECTION

- **Voltage**: Select the voltage. Allowable values are **0** (default), **3.3**, and **5.0**.
- **Trigger Source**: Selects the source from which the trigger will be generated. **Front Panel**, **Software Trigger**, **Backplane Line 0** through **7** are provided as options.
- **Polarity**: Sets the ports polarity as either **Normal** or **Inverse**.
- **Gate Polarity**: Sets the gate as either **Active High** or **Active Low**.
- **Pattern Repeat**: Determines whether the patter will repeat after completion or if it will move to the End State.
- **End State**: The End State is the state the device goes to when it is finished generating a pattern or is aborted
- **Number of Patterns**: Indicates the number patterns available for the user to use. The number must be a value between 1 and 512 that is a power of 2.
- **Partition Pattern Space** button: When a value is entered in the **Number of Patterns** field, the **Partition Pattern Space** button becomes active. Clicking on this button partitions the memory to the appropriate size based on the value set for **Number of Patterns**. For more information, see *Pattern Generation* in *Section 3*.
- **Initiate** button: When this button is clicked, pattern generation begins.
- **Current Pattern**: Indicates the number of the pattern that is being generated. The pattern numbers are indicted in the pattern data table below this field.
- **Save Changes** button: Saves any changes made to the pattern to the FIFO.
- **Save to Disk** button: Saves the pattern to disk as a CSV.
- **Load from Disk** button: Loads a pattern from disk. Must be in a CSV format.
- **Move Data Up** button: Moves a row of data up one row.
- **Move Data Down** button: Moves a row of data down one row.
- **Add Row** button: Clicking this button adds a row to the pattern data table. After adding a row, the value for **Data3** through **Data0** can be input.
- **Remove Row** button: Clicking this button will remove the row that is selected in the patter data table.

Pattern Acquisition Settings

When pattern acquisition is enabled, the **Pattern Acquisition** section allows the user to configure how and when pattern acquisition will take place.

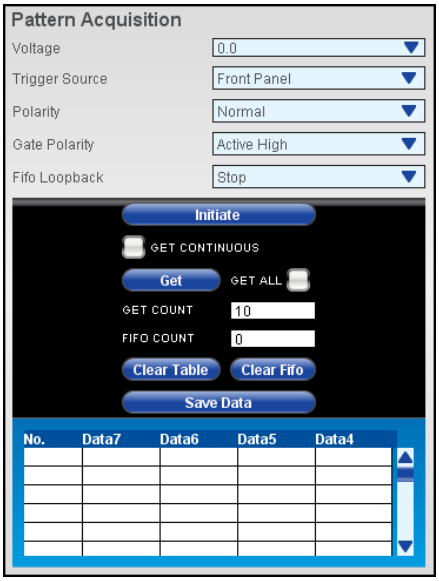


FIGURE 4-9: PATTER ACQUISITION SECTION

- **Voltage**: Select the voltage. Allowable values are **0** (default), **3.3**, and **5.0**.
- **Trigger Source**: Selects the source from which the trigger will be generated. **Front Panel**, **Software Trigger**, **Backplane Line 0** through **7** are provided as options.
- **Polarity**: Sets the ports polarity as either **Normal** or **Inverse**.
- **Gate Polarity**: Sets the gate as either **Active High** or **Active Low**.
- **FIFO Loopback**: Determines whether the patter will repeat after completion or if it will move to the End State.
- **Initiate** button: Clicking this button moves the DIO to the INIT state.
- **Get Continuous** checkbox: If enabled, this button disables the **Get** button and continually populates the data table as data is acquired.
- **Get** button: When clicked, data is immediately pulled from FIFO memory.
- **Get All** checkbox: When enabled, clicking the **Get** button will retrieve all of the current data. Once the FIFO is empty, it will stop acquiring retrieving data.
- **Get Count** field: Indicates the number of data points that will be returned when the **Get** button is clicked. Should be less than or equal to the **FIFO Count**.
- **FIFO Count** field: Indicates the number of data points available in the FIFO memory.
- **Clear Table** button: When clicked, all of the data currently in the table is erased.
- **Clear FIFO** button: When clicked, any data points currently stored in the FIFO memory will be erased.
- **Save Data** button: When clicked, the data in the table can be saved. A .csv file is generated by default.

DEVICE INFO TAB

If the **Device Info** tab is clicked, information regarding the selected instrument will be displayed including the versions of the SFP, firmware, and hardware.

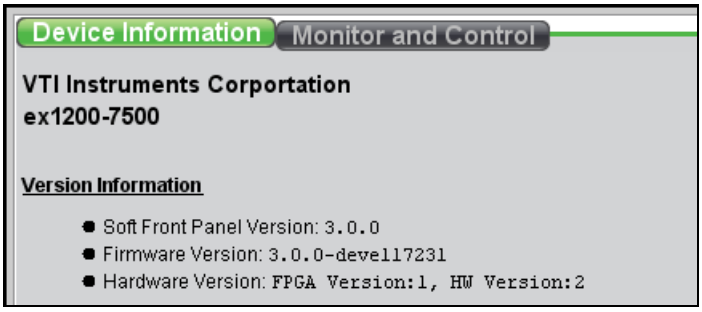


FIGURE 4-10: DEVICE INFORMATION WEB PAGE

SECTION 5

THEORY OF OPERATION

INTRODUCTION

This section provides an overview of how the EX1200-7500 works. The figure below shows illustrates this process.

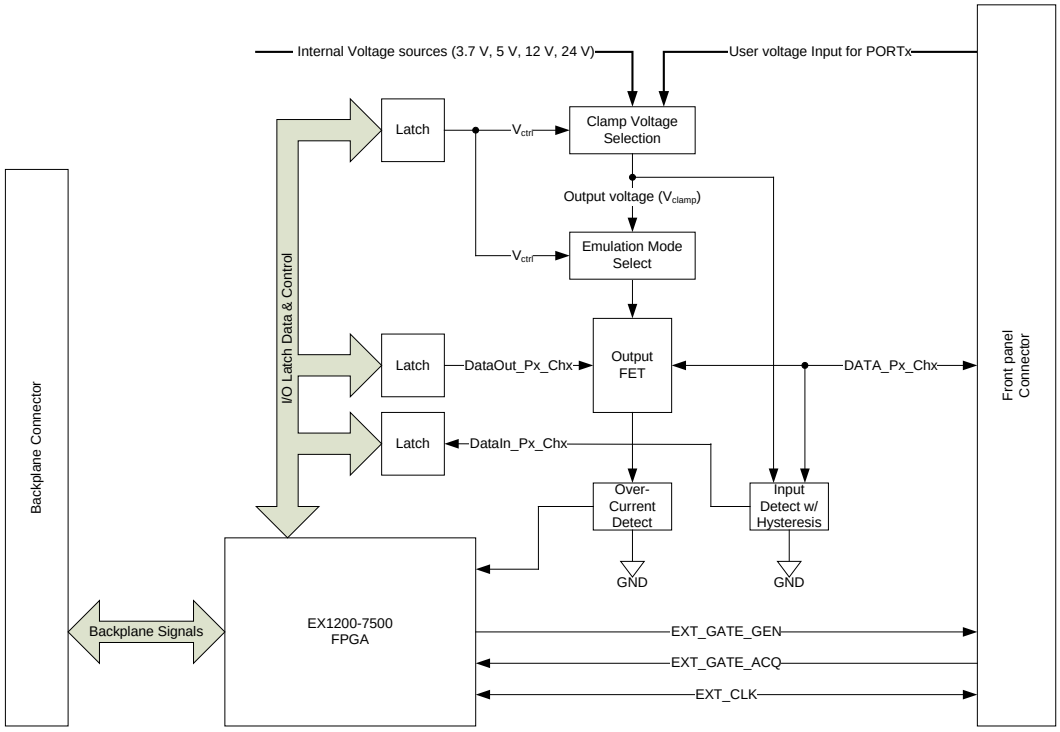


FIGURE 5-1: EX1200-7500 BLOCK DIAGRAM

OVERVIEW

The EX1200-7500 supports a static mode where user-driven software commands execute reading/writing data channels on a per port basis. Ports can also be grouped together as up to four continuous ports (0 through 3 and 4 through 7) for reading and writing. Each port can be programmed for direction and polarity. The default direction is input and default polarity is normal. Ports also need a voltage reference before data can be read or written; this voltage is used for pulling up open drain data lines and setting reference voltage for threshold detection in input path. No voltage should be applied on data lines before voltage selection and during voltage configuration process. User can select a voltage from 3.3 V, 5 V, 12 V or 24 V provided on the module or supply a voltage through front panel connector, to be used as V_{CLAMP}. By default, no voltage is selected and data lines are floating. There is also an option of configuring a port's voltage setting as either TTL or Low Voltage emulation. In TTL emulation setting, open drain lines

are pulled up to 5 V and have a sourcing capability of 4 mA, whereas in Low Voltage (LV) option the pull-up voltage is 3.3 V with 4 mA drive strength.

When doing read and writes, data written to a port set for output updates data on that port, data written to a port set for input does not have any effect. Data read from a port always reads current status of data pins (whether driven by card or by user). Time stamp is maintained on reads.

When a port's direction is output, data is driven by the DIO on the corresponding channels and no user voltage should be applied on those channels. When direction is input, data lines are pulled up to V_{CLAMP} using 100 kΩ resistor and user can drive their data on the channels with a V_{IN} range of zero to V_{CLAMP}.

TTL and LV emulation voltage selection is only applicable when port's direction is output. If TTL or LV emulation voltage is selected for a port and user changes the direction of that port to input, the card automatically switches to 5 V or 3.3 V, respectively, with high impedance pull-up. When direction is changed back to output, TTL or LV configuration is restored. Similarly if direction is input and TTL/LV voltage is selected, 5 V/3.3 V with high impedance pull-up is applied.

Data in output register(s) of concerned port is reset (set to all F's) after reset and any operation that involves voltage configuration, as listed below:

1) When card is reset. All ports are affected.
2) A port's direction is changed when its voltage is set for TTL/LV emulation. Only concerned port is affected.
3) Port voltage configuration is done. Only concerned port is affected.

Data in output registers is retained even if direction is changed. So for example, when an output port that is driving AA on data lines is made input, all its channels are pulled up so that user data can be driven but when the direction is changed back to output, AA gets driven, given that polarity and voltage setting didn't change.

Polarity settings are applicable irrespective of direction. On output channels, changing the polarity flips the state of data lines and further writes are done according to polarity selection (for example, if polarity is configured as reversed then writing 1 to a channel writes a 0 to it and vice versa). Read back of data, whether driven by card when direction is output or driven by user when direction is input, also checks for polarity and if polarity is reversed, 1 is read as 0, otherwise as 1 and 0 is read as 1, otherwise as 0.

CONFIGURATION SECTION

This section of circuit is used to select reference voltage (V_{CLAMP}) for ports. Each port has a set of programmatically controlled switches that let the user select a voltage from 3.3 V, 5 V, 12 V, 24 V and User supplied voltage, which can be anywhere from 3.3 V to 60 V. In addition, LV and TTL emulation modes can be selected which use 3.3 V and 5 V as V_{CLAMP} respectively, and insert a low impedance pull-up resistor in the path to source 4 mA.

OUTPUT SECTION

The output section of each channel consists of an N-channel MOSFET, pull-up resistors and over current sense circuitry. When a 1 is driven on channel, the MOSFET is turned off and data line is pulled-up to V_{CLAMP} using appropriate pull-up resistor (330 Ω for LV emulation, 500 Ω for TTL emulation and 100 kΩ otherwise). The rising voltage level at data channel exhibits RC charging with a rise time of approximately 70 µs with 100 kΩ pull-up resistor and about 600 ns with low impedance pull-up in TTL and LV voltage configurations. When a 0 is driven on channel, the MOSFET is turned on and data line is connected to ground via a current sense resistor of 0.3 Ω. The falling edge is much sharper with a maximum fall time of 300 ns approximately. When the

port is set as input, the data lines are pulled up using 100 kΩ resistor. Please refer to electrical characteristics for output voltage levels with different load resistance.

The load resistor and pull-up resistor form a voltage divider off of V_{CLAMP} and determine output voltage, thus output voltage decreases with decreasing load resistance. The minimum load resistance that can be connected on channels is 500 Ω for LV emulation, 750 Ω for TTL emulation and 150 kΩ otherwise.

The data lines have a current sinking capacity of 300 mA. If the current through MOSFET increases above limited value, the over current protection circuit turns off all FETs in the concerned port. The over current condition must last for greater than 12.8 µs before over current protection circuit is activated. This is done to prevent false over current events due to transients. It takes approximately 20 µs from first occurrence of the over current event before the FETs are truned off. The over current circuit will activate with 330 mA hold current and 475 mA trip current, where hold current is the maximum current circuit will allow without tripping and trip current is the minimum current required to trip the circuit. The absolute maximum limit on sinking current is 500 mA. This is true when only one card in a system. If multiple cards are installed, then the current through channels MUST be adjusted such that total current is no more than 32 A. Staying within these limits is user's responsibility. Over current readings from ports will be latched and can be read by user. When they reset over current condition then port will be enabled and data last written to port will be applied on channels.

INPUT SECTION

The input section converts the high or low voltage reading on the data channels to 1 or 0 that can be read by the user. The port's V_{CLAMP} voltage is used as a reference for threshold detection. If voltage on data line is greater than 40% (range spreads to 37% to 42% across different voltage levels) of port's V_{CLAMP} voltage, 1 is read and if voltage on data line is less than 13% (10% to15% range) of port's V_{CLAMP} voltage, 0 is read. Hysteresis loop is implemented on input comparison meaning that on rising voltage, level has to cross higher threshold (VIH) to record a 1 and on falling voltage, level has to cross lower threshold (VIL) to record a 0. In between VIH and VIL the digital state of data channel reading stays the same. Please refer to the recommended operating conditions for input threshold values.

The input section also has a fly-back protection diode. The port's reference voltage V_{CLAMP} is routed to the cathode of a fly-back protection diode, whose anode is then connected to the associated port's data line. Every data line has a diode installed to suppress transients in case of voltage overshooting. The current through diode must be limited as indicated in absolute maximum ratings.

INDEX