PCIe + PTP Time and Frequency Processor

PCIe-1000 PTP Clock Card

User Manual

Revision A

Product CD 098-00445-000

Notices

Copyright

Due to continued product development this information may change without notice. If you find any errors in the documentation, please report them to us in writing. Symmetricom, Inc. does not warrant that this document is error-free.

Limited Product Warranty

Hardware – Symmetricom warrants, for a period of twelve (12) months from Symmetricom's date of shipment, the Goods shall be free from defects in design, material, and workmanship under normal use and service, and shall conform to and perform substantially in accordance with Symmetricom's published specifications in effect at the time of shipment. Symmetricom further warrants that the Goods shall be free and clear of all liens and encumbrances and shall have good and valid title at the time of shipment by Symmetricom. This warranty shall survive inspection, acceptance, and payment by Buyer. Symmetricom does not warrant that the operation of the Goods shall be uninterrupted or error free or meet Buyer's intended use or purpose. Symmetricom's warranty does not cover failures caused by acts of God, including electrical or environmental conditions; abuse, negligence, accident, damage in transit; or improper site preparation.

This warranty shall be null and void in the event (i) Buyer or any third party repairs or attempts repair of the Goods without Symmetricom's advance written authorization; or (ii) defects are the result of repairs, modifications, alterations, improper or inadequate maintenance by Buyer or third party; or (iii) damages to said Goods are caused by Buyer or third party-supplied software, interfacing or supplies; or (iv) of improper use (including termination of non-certified third party equipment on Symmetricom's proprietary interfaces and operation outside of the product's specifications) by Buyer or third party; or (v) the Goods are shipped to any country other than that originally specified in the Buyer's purchase order.

Buyer's sole remedy for a breach of the foregoing Goods warranty, whether express or implied, how-soever arising, shall be as set forth in this Section. Goods not meeting the foregoing warranty during the warranty period shall be repaired or replaced, at Symmetricom's option, upon return of such Goods to Symmetricom's factory; provided, however, that Buyer has first obtained a return materials authorization ("RMA") number from Symmetricom authorizing such return. Buyer may obtain an RMA number by logging onto Symmetricom's website www.symmetricom.com. Buyer shall place the RMA number on the exterior packaging of all returns.Buyer shall be responsible for the shipping costs to ship the Good to Symmetricom and Symmetricom shall pay for shipping costs to return the repaired or replacement Good to Buyer. Repaired or replaced portion of the Good shall be warranted for the remainder of the unused warranty term or for ninety (90) days from shipment, whichever is longer.

Software Media - Symmetricom warrants that the accompanying media shall be free from defects in materials and workmanship under normal use for a period of ninety (90) days from date of shipment. The physical media warranty does not apply to defects arising from misuse, theft, vandalism, fire,

water, acts of God or other similar perils. Symmetricom shall not be liable for any damages caused by the Buyer's failure to fulfill its responsibilities as stated above. Buyer's sole and exclusive remedy and Symmetricom's entire liability for a breach of the foregoing warranty shall be for Symmetricom, at its option to replace the Software media, or if unable to replace the Software media, then to refund the license fee paid for the Software.

THE FOREGOING WARRANTY SHALL BE THE ONLY WARRANTY WITH RESPECT TO THE SUBJECT MATTER HEREOF AND SHALL BE IN LIEU OF ALL OTHER WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, ANY IMPLIED WARRANTIES OF TITLE, MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON INFRINGEMENT HOWSOEVER ARISING. WHERE LEGISLATION IMPLIES IN THIS AGREEMENT ANY CONDITION OR WARRANTY AND THAT LEGISLATION VOIDS OR PROHIBITS PROVISIONS IN A CONTRACT WHICH EXCLUDE OR MODIFY THE OPERATION OF THAT CONDITION OR WARRANTY, THE CONDITION OR WARRANTY IS DEEMED TO BE INCLUDED IN THIS AGREEMENT. HOWEVER, SYMMETRICOM'S LIABILITY FOR BREACH OF THE CONDITION OR WARRANTY WILL BE LIMITED AT SYMMETRICOM'S OPTION TO REPLACE OR REPAIR THE GOODS. TO THE EXTENT ANY OF THE FOREGOING LIMITED REMEDY FINALLY FAILS ITS ESSENTIAL PURPOSE, SYMMETRICOM'S TOTAL LIABILITY TO BUYER FOR SUCH BREACH SHALL BE LIMITED TO THE ACTUAL PRICE PAID BY BUYER FOR THE DEFECTIVE GOODS AND SUBJECT TO THE "LIMITATION OF LIABILITY" SECTION.

NOTE: Symmetricom's GPS positioning products for navigation are an AID TO NAVIGATION only and MUST be used in conjunction with normal navigation practices

Limitation of Liability

The remedies provided herein are the Buyer's sole and exclusive remedies. In no event or circumstances will Symmetricom be liable to Buyer for indirect, special, incidental or consequential damages, including without limitation, loss of revenues or profits, business interruption costs, loss of data or software restoration, or damages relating to Buyer's procurement of substitute products or services. Except for liability for personal injury or property damage arising from Symmetricom's negligence or willful misconduct, in no event will Symmetricom's total cumulative liability in connection with any order hereunder or Symmetricom's Goods, from all causes of action of any kind, including tort, contract, negligence, strict liability and breach of warranty, exceed the total amount paid by Buyer hereunder. SOME JURISDICTIONS DO NOT ALLOW CERTAIN LIMITATIONS OR EXCLUSIONS OF LIABILITY, SO THE ABOVE LIMITATIONS OR EXCLUSIONS MAY NOT APPLY TO ALL BUYERS.

Contact Information

Government and Enterprise Business Unit

Symmetricom, Inc.

2300 Orchard Parkway

San Jose, CA 95131-1017

Sales and Technical Support

To order any accessory, contact the Symmetricom Sales Department. See http://www-.symmetricom.com/company/contact-information/sales-offices for sales support contact information.

If you encounter any difficulties installing or using the product, contact Symmetricom Global Services:

U.S.A. Call Center

Symmetricom, Inc.

2300 Orchard Parkway

San Jose, CA 95131-1017

Toll-free in North America: 888-367-7966

Telephone: 408-428-7907

Fax: 408-428-7998

E-mail: support@symmetricom.com

Internet: www.symmetricom.com

Europe, Middle East, and Africa (EMEA) Call Center:

Symmetricom Global Services EMEA

Symmetricom Europe GmbH

Altlaufstrasse 42

85635 Hoehenkirchen-Siegertsbrunn Germany

Telephone: +49 700 3288 6435

Fax: +49 8102 8961 533

E-mail: emeasupport@symmetricom.com

Internet: www.symmetricom.com

Revision History

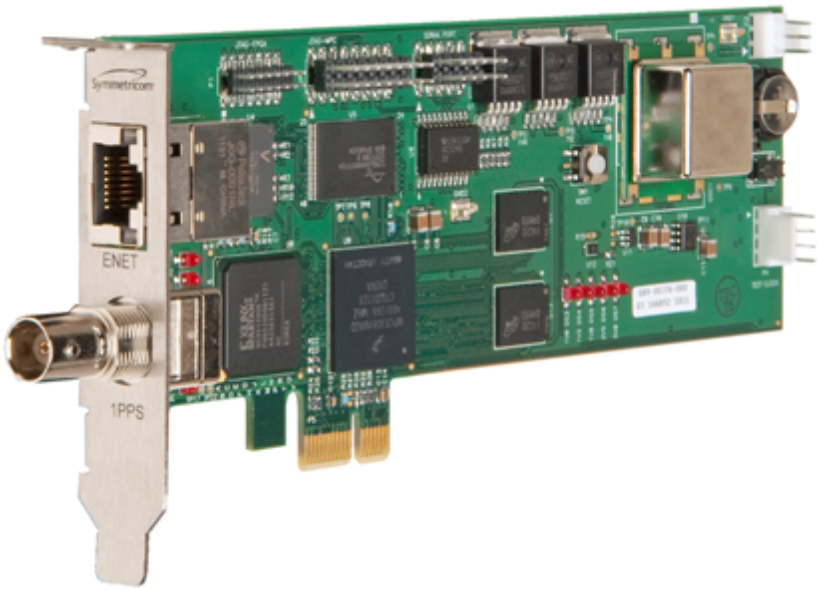
Revision	Date	Description
A	29 March, 2012	First published.

This page intentionally left blank

Table of Contents

This page intentionally left blank

The PCIe-1000 PTP Clock Card



Introduction

The Symmetricom® PCIe-1000 PTP Clock Card is referred through this manual as the PCIe-1000 card. It is also referred to in other literature as SyncPoint PCIe-1000.

The PCIe-1000 card provides ultra high availability time with sub-microsecond accurate time stamps for programs running in Linux. The IEEE 1588 based PCIe-1000 has been optimized to be resilient to network related time errors and provides nanosecond caliber time as needed to a Linux program in under a microsecond.

The PTP synchronized PCIe-1000 synchronizes to a PTP grandmaster, such as a SyncServer, over a network. Networks can introduce arrival time jitter of critical timing packets at the PTP slave, called packet delay variation (PDV). To overcome PDV the PCIe-1000 deploys state-of-the art filtering and servo algorithms, accommodates increased packet exchange rates, and includes a high performance OCXO oscillator as standard. The net result is an extremely accurate clock that is very resilient to network impairments.

The PCIe-1000 includes a 1 PPS output signal with a convenient BNC connector that is useful to compare the time on the card to that of the master. This is useful in adjusting the PTP parameters for optimal time transfer accuracy over the network. The 1 PPS is also useful to synchronize adjacent network devices or probes that accept a 1 PPS input.

Once synchronized, the PCIe-1000 provides time to the host machine by either writing the time directly to a host memory location or by responding to requests for time over the PCIe bus.

Applications accessing the hosted memory time location can read the time in excess of one million times per second and retrieve monotonically advancing time.

The low profile PCIe-1000 with its Gigabit Ethernet interface is readily adaptable to modern networks and servers while also consuming a minimal amount of power. With its x1 PCIe connection the PCIe-1000 fits in a wide variety of server types with a minimal power footprint.

Integration of the PCIe-1000 is straight forward using the included Linux driver and control interface. The included source code means those using the very latest Linux version can be compatible without having to wait for any proprietary software to be upgraded.

Speed and Accuracy

The PCIe-1000 has been optimized for an environment where high-speed, low-latency access to time is just as important as the precision and accuracy of the time itself. The PCIe-1000 can continuously write time to host memory. This time is available in two formats, either Major.Minor or a total nano-seconds counter value.

The advantage of reading the time in total nanoseconds is that in a single 64 bit memory read the entire time can be consumed by an application. (This time can be converted to Major.Minor format later if desired.) This technique reliably provides monotonically advancing, sub-microsecond accurate time with every time read at rates beyond a million times per second. This technique is also more accurate and reliable than using software that reads time from a clock card to steer the local Linux clock from which an application reads the time. This is due to the instability of the local Linux clock versus the high stability of the PCIe-1000 clock.

Reading Major.Minor time is a two step operation where Major (integer seconds) is read from one location and Minor (fractional seconds) from another.

Alternatively, the host can initiate a read over the PCIe bus, but this will be subject to PCIe bus delays that can be up to 3 microseconds.

PTP Enhancements to Overcome PDV

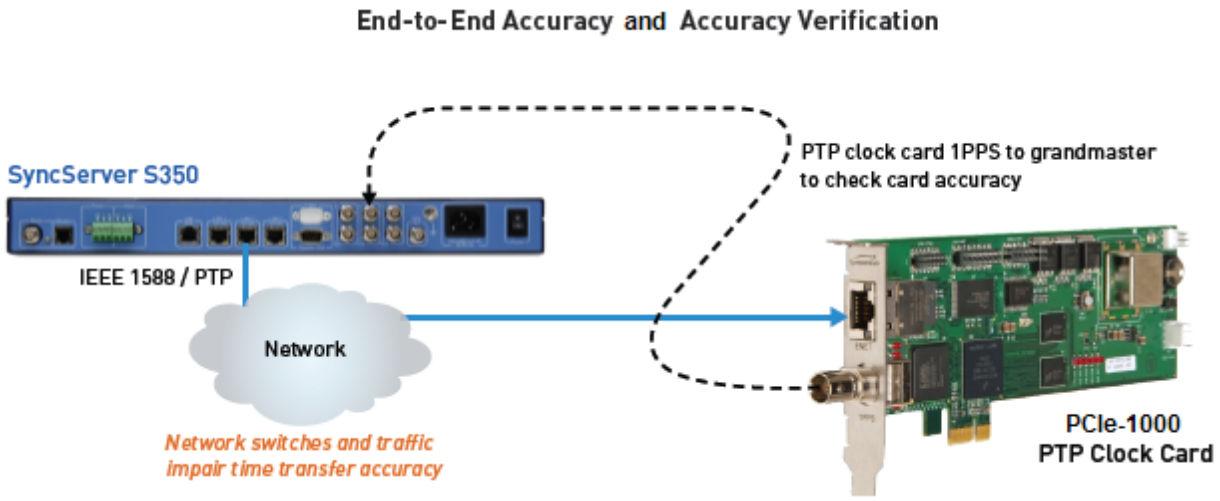
Aside from support of the IEEE 1588 Default Profile, the PCIe-1000 supports increased timing packet exchange rates. Decades of Symmetricom timing expertise is incorporated in the filtering and servo technology in the PCIe-1000 to take advantage of these increased packet rates useful in overcoming PDV. As a result, PTP packets can transit more standard network switches and related packet queues while the PCIe-1000 still maintains a very high degree of accuracy. In many cases the need for expensive PTP-enabled switches such as boundary clocks and transparent clocks can be eliminated.

Linux Software Driver Details

A simple interface to the PCIe-1000 is all that is necessary to configure the clock card. In fact, the PCIe-1000 can automatically receive many of its configuration parameters directly from the PTP grandmaster.

The Linux driver to interface the PCIe-1000 is straight forward as well. Driver support for the Host PC Linux drivers and sample applications code is provided to greatly assist integration into the customer's environment.

PCIe-1000 and the SyncServer S350



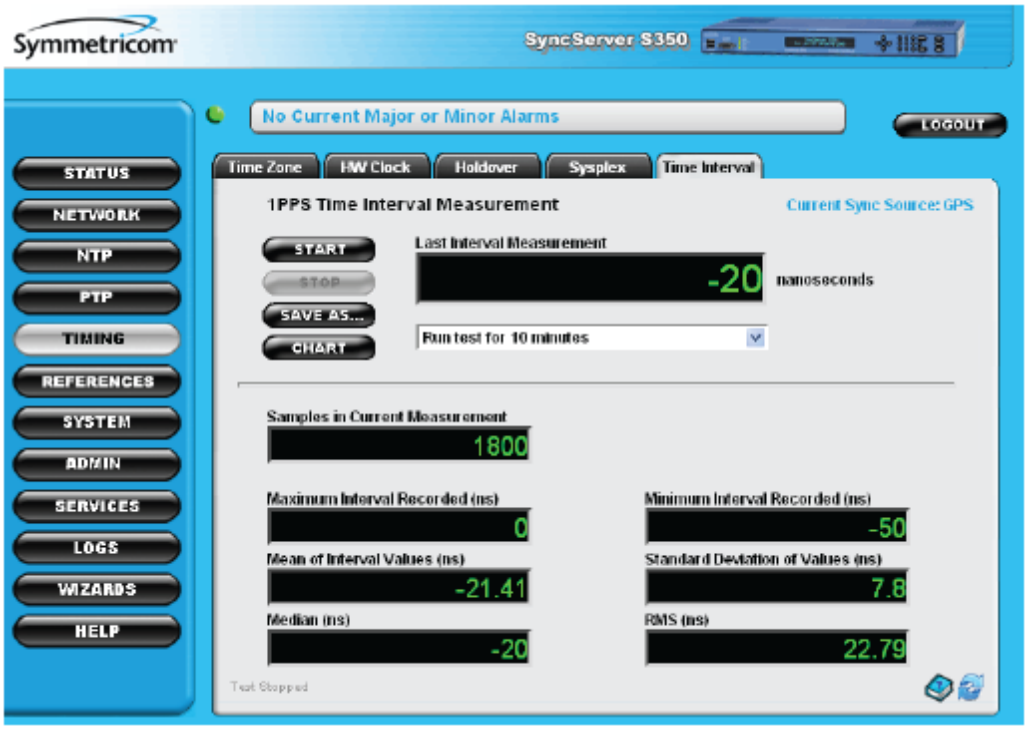
Closed Loop Accuracy Verification

While the PCIe-1000 conforms to the IEEE 1588 standard and will interoperate with any other PTP master that also conforms, there are advantages to working with the Symmetricom SyncServer® S350. The S350 has a built-in Time Interval measurement function that provides real time offset statistics of an externally supplied 1PPS signal. By taking the 1PPS output from the PCIe-1000 and connecting it to the S350, the S350 can measure the exact time difference between the two clocks down to 5 nanoseconds.

Real-Time Measurement Statistics and Charting

The PTP slave accuracy statistics relative to the S350 are computed and displayed in real-time via S350 and displayed in the web interface. The sample count, max, min, mean, standard deviation, median and RMS of the measurement set is continuously updated on the instrument-like web interface throughout the duration of the test.

Data can be charted in the web interface in real-time either as a time series or histogram. The chart continues to update as the measurements are made or charting can be paused and viewed statically while measurements are made in the background. The charting is very flexible with line, scatter, column or histogram formats with a variety of viewing options. Measurements can be performed unattended and for specified periods of time. Once a test is completed the data can be saved to a local text file.



Real World Accuracy Measurements

Once the baseline accuracy has been verified via crossover cable and the 1PPS loopback from the PCIe-1000 to the S350, network elements and traffic can be added and the effect on time accuracy at the PCIe-1000 can be directly measured. If the accuracy has degraded due to network impairments the packet rate can be increased as appropriate to optimize the time accuracy for that network/traffic configuration. This 1PPS loopback accuracy testing is very useful to measure the detrimental effects of switches and data traffic as well as the effectiveness of boundary clocks or transparent clocks.

Product Contents

This product consists of:

- PCIe-1000 PTP Clock Card
- Standard height and low-profile cover plates
- One year warranty
- PCIe User's Guide CD
- Linux SDK/Driver software CD

Getting Started

To successfully install and run the PCIe-1000 bus card, perform the following:

1. "Installation" on page 6
2. "Installing Drivers and Demo Program" on page 7
3. Use the "The Demo Program to Test the PCIe-1000 PTP Clock Card" on page 9
4. Use the Linux API library to develop application programs

General PCIe-1000 PTP Clock Card Information

The PCIe-1000 is a PCIe compliant Add-in card. It supports PCIe GEN1 single lane. The hardware has following key features:

- 32-bit microprocessor (Freescale MPC8303 running at 400 MHz)
- 512MB of DDR RAM. This memory is used as program storage and execution memory for the software running on the processor.
- 8MB NOR Flash ROM for storage of the boot loader.
- 512MB of NAND Flash for storage of operating system.
- FPGA used for the generation of a 1PPS signal.
- 10/100/1000BaseT PHY chip and magnetics for connection to RJ-45 Category 5 cable. This is the port used for PTP Sync and Management traffic.
- 100MHz OCXO precision oscillator.

Installation

General

This section contains installation instructions and information regarding operating the PCIe-1000 card.

Installation of PCIe-1000 cards is simpler than other bus architectures due to the following factors:

- Geographical addressing, which eliminates the need for DIP switches and jumpers normally required to select a “base address” or interrupt level for plug-in modules.
- Auto configuration that allows the host computer to read the device ID, and other configuration information directly from the PCI Configuration Registers.
- The PCIe-1000 is shipped with software suitable for use with Linux. This includes drivers for low-level access, as well as a demo software program for accessing the card.

Installation is as easy as choosing a vacant x1 PCIe slot, plugging in the Symmetricom PCIe-1000 PTP clock card and installing the device driver. Be sure to consult the user documentation that came with your particular workstation for any specific installation instructions. In addition, to protect the card, use good ESD protection practices when installing the card.

Installing the PCIe-1000 PTP Clock Card in the Host Computer

- Unpack the card and carefully inspect it for shipping damage. Report any damage to the carrier immediately.
- With the computer's power turned OFF, install and secure the card in an empty x1 PCIe card slot.Note, you cannot hot-swap the PCIe-1000 card, power has to be turned off for installing or replacing this card. If a different front panel needs to be attached to the bus card, follow the instructions below.
- After installing the PCIe-1000 card connect power to the PC. All LEDs on the bus card will be lit.
- The bus card program is loaded from flash storage and runs. A single LED will slowly flash indicating correct operation of the PCIe-1000 card.

Changing the Bus Card Front Panel

The PCIe-1000 card is shipped with a front panel attached to it. A second front panel is included as an alternative. If the alternative front panel option is required, use the following procedure. Note, the following tool will be required to change the front panel:

- Phillip's #1 screwdriver

Standard ESD precautions should be followed when handling the bus card printed circuit board.

Procedure for Changing the Bus Card Front Panel

The PCIe-1000 card is attached to the front panel in two places, with two Phillip's #1 screws. To change the front panel, do the following:

1. Remove the two Phillip's #1 screws that secures the PCIe-1000 card directly to the front panel.
2. Set the front panel aside and pick up the alternative front panel.
3. Tighten the two Phillip's #1 screws connecting the PCIe-1000 card directly to the front panel.

Minimum System Requirements

- PC with one free x1 PCIe slot
- 25 MB disk space

Installing Drivers and Demo Program

The PCIe-1000 PTP clock card Linux Developer's API is designed to provide development of applications which access features of the PCIe-1000 bus card.

To successfully operate the PCIe-1000 card, ensure that a late version of either a 32 bit or 64 bit Linux operating system is installed. In addition to the 32 bit or 64 bit Linux operating system, the following software modules need to be installed:

- Boost version 1.44 or greater
- GCC
- G++
- Gawk
- Terminal Window

Software Installation

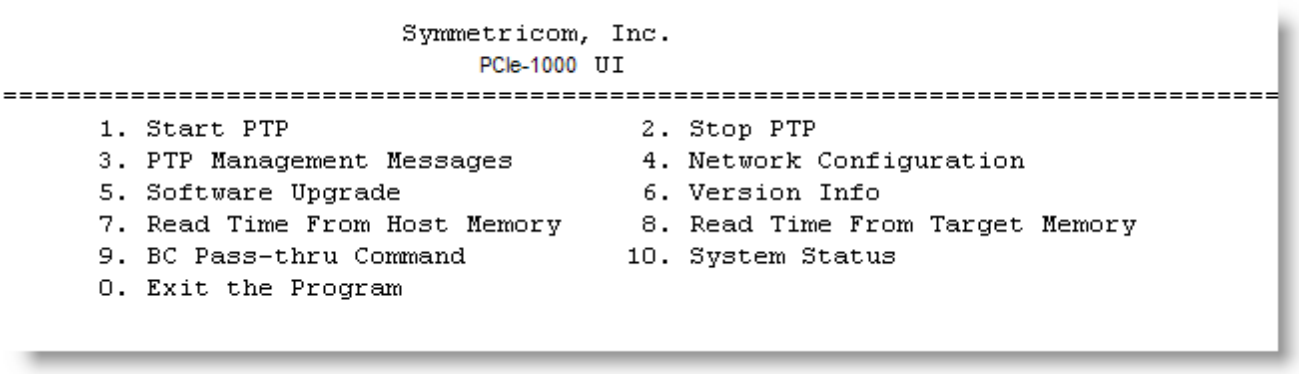
To develop Linux based applications for the Symmetricom PCIe-1000 card, you must install the Linux Software Drivers included on the PCIe-1000 product CD.

1. With the PCIe-1000 card installed, apply power to the host PC or workstation.
2. When the Linux OS has fully loaded, navigate to the User folder.
3. Create and name a new folder, for example, “PCIe-1000”.
4. Load the PCIe-1000 PTP Clock Card CD into a CD drive on the host PC.
5. Copy the file “**PCIe-1000-SourceRelease-1.0y.gz**” (where **"y"** is the build number) into the new folder “PCIe-1000”.
6. In the new folder, run gunzip on the “PCIe-1000-SourceRelease-1.0y.gz” file, then run the tar command: “**tar xvzf PCIe-1000-SourceRelease-1.0y.gz**”
7. The files are unpacked in a newly created folder named "build".
8. Change directories to the "build" folder, and type “**./configure**”.
9. At the prompt, type “**make**” to compile the driver and demo program.

10. At the prompt, type "**sudo make install**" to install the PCIe + PTP bus card driver and demo programs.
11. At the prompt, type "**ls**" to list the files and note that the "examples" folder with the demonstration programs has been created.
12. At the prompt, type "./**examples/cli_demo/cli_demo**" to run the command line interface (CLI) demonstration program.

The Demo Program to Test the PCIe-1000 PTP Clock Card

When the demonstration program launches, it opens with the main menu as follows:



See the "User Interface" on page 15 for information on each user interface option.

See also the Demonstration Program (cli_demo) included with the software for the code examples of many of the API commands.

Checking out the PCIe-1000 card

Checking out the PCIe-1000 card is a two step process. The associated grandmaster clock needs to be configured, and then PCIe-1000 card configured so that the two devices interact correctly.

Configuring a PTP Grandmaster

The model of Grandmaster serving time to the Symmetricom PCIe-1000 card installed in your PC will vary. As an example, the following section gives the settings used for the Symmetricom SyncServer.

Settings for the Symmetricom IEEE-1588 Grandmaster

The settings needed in the IEEE 1588 Grandmaster PTP to connect to the Symmetricom PCIe-1000 card installed in your PC are as follows.

- PTP set to Master
- Transport Protocol: UDP
- Sync Interval: 32 packets/1 second
- Delay Mechanism: E2E
- Packet TTL: 6
- E2E Delay Interval 32 pkt/s 1 sec, when using the P2P delay mechanism
- P2P Delay Interval 32 pkt/s 1 sec, when using the E2E delay mechanism
- Priority 1: 128
- Priority 2: 128
- Domain Number: 0
- Mean Announce Message Transmit Interval: 2 Sec
- Announce receipt timeout Multiplier: 3

For more information on terms and descriptions for IEEE 1588 Grandmaster PTP configuration settings, see "Appendix A: IEEE-1588 PTP Grandmaster Configuration " on page 59

Also see "Appendix B: Best Master Algorithm" on page 65

Important note: When the Grandmaster is reconfigured after the initial set up, all PCIe-1000 cards need to have PTP restarted so as to synchronize with the Grandmaster. See "Starting and Stopping PTP" on page 16

Configuring the PCIe-1000 card

With the PCIe-1000 UI displayed on the host PC:

1. Select 2 to stop PTP
2. Select 4 and then 2 to set the static IP address, then 3 to set the connection type to 1 for cross-over cable.
3. Select 1 to start PTP (so that the card can read the GM setting)
4. Select 10 to see that PTP is indeed running
5. Select 3 and then 14 to set the domain, then 15 to enable the port

Also see "Appendix B: Best Master Algorithm" on page 65

Reading Time

Some information on how time is accessed between the host computer and the target PCIe slave card will be helpful before reading the next sections on the UI Interface and Developing Programs.

The PCIe card also known as the target, stores time in two forms:

1. A counter made up of a 64 bit register. The counter gives the time in nanoseconds. Because it is a homogenous 64 bit counter, its contents are referred to as binary time. The letters "Bin" appear in the API command.
2. The counter value computed in whole seconds (Major Time), and the remainder nanoseconds (Minor Time). The major time is obtained by dividing the counter time by 1 billion. The major time is therefore the major part of the binary time in whole seconds. The remainder is the minor part of the binary time as a total of nanoseconds.

The 64 bit counter on the PCIe slave card is steered remotely by the designate grandmaster clock over the network.At the same time on the PCIe slave card, the value of the steered register is computed in Major Time and Minor Time. Both the steered register value and Major/Minor Time can be read over the PCIe bus as register "Reg" API command. API commands reading time from the target PCIe slave card include "Reg" in their API commands. For example, the API command "bcReadRegBinTime" on page 52.

In addition to time being stored on the PCIe slave card, the values of the steered 64 bit register and Major/Minor Time are placed into memory on the host computer once every 200 nanoseconds. In this manner,there is therefore some time latency between getting time from the host memory as opposed to reading time over the PCIe bus.

Several considerations:

- The host memory values are refreshed once every 200 nanoseconds.
- Additional time is required to compute the Major/Minor Time from the Counter Time.
- Reading time from the target counter is the most accurate time because of the previous two considerations.

Atomic, non-atomic time

Atomic time is a term usually used to refer to an official time scale that is based on atomic clocks.

Examples are UTC, UTC(NIST), UTC(USNO), TAI, GPST, etc.

There are many non-Atomic times such as UT, which is based on the rotation of the earth, and Ephemeris time, which is based on position of the earth in its orbit around the sun.

Sync and NoSync

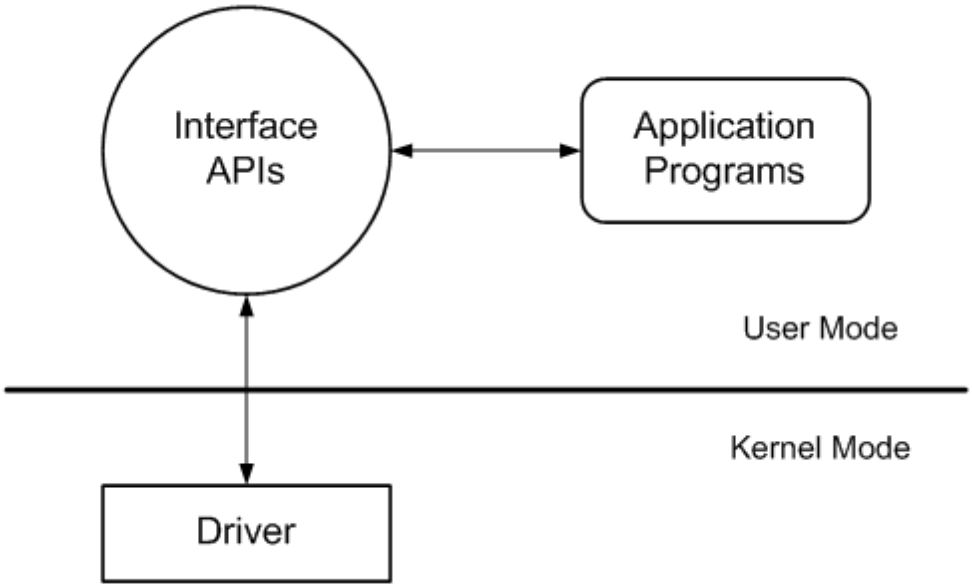
“NoSync”, means that the operation is non-atomic. A memory read "Mutex: " on page 67 is not used while reading the time information. If the user call the “NoSync” API functions consecutively, there is no guaranteed that the last read will have updated information. It could be reading the same time as the previous read.

To safeguard against this, the user will have to add detection themselves to determine if the time read is updated.

As for the API's without the "NoSync", a memory read mutex is used (atomic).This ensures that once the function returns the read time, it is updated and valid.

Developing Programs using the API

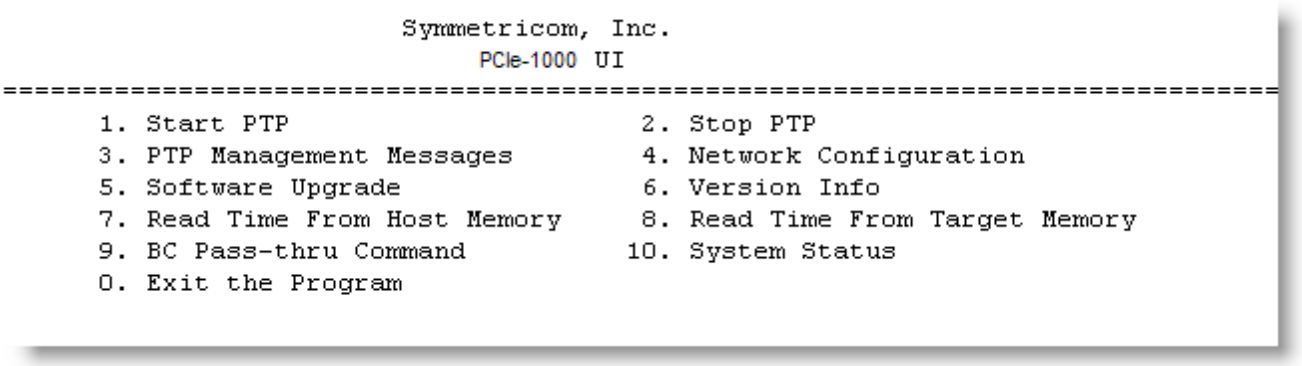
The Developer's API is designed to provide an interface to the PCIe-1000 PTP clock card in the Linux OS environment. The Example/Demo program provides sample code to exercise the PCIe-1000 card. It was developed using discrete functions for each operation, allowing the developer to clip any useful code. The code included can be viewed following the next section.



The interface API library provides an abstraction layer between the application programs and the device driver. This allows Symmetricom to advance the Time and Frequency Processor hardware features while protecting your investment in application development. Symmetricom will maintain interface API compatibilities between the current product release and future product releases.

User Interface

The PCIe-1000 user face (UI) is a simple text based UI. From the UI main menu, the user is able to interact with the PCIe-1000 card by selecting any of the ten options shown in the following image.



Starting and Stopping PTP

1. Starting PTP

```
                    Symmetricom, Inc.
                        PCIe-1000 UI
================================================================================
      1. Start PTP                          2. Stop PTP
      3. PTP Management Messages            4. Network Configuration
      5. Software Upgrade                   6. Version Info
      7. Read Time From Host Memory         8. Read Time From Target Memory
      9. BC Pass-thru Command              10. System Status
      0. Exit the Program
```

When the PTP is already stopped, selecting option 1 will return the following:

```
********************************************************************************
                              PTP
********************************************************************************
PTP Started Successfully!
********************************************************************************
```

See option 10 to determine whether PTP is running or stopped.

When the PTP is already running such as when the PC is powered up, selecting option 1 will return the following:

```
********************************************************************************
                              PTP
********************************************************************************
bcStartPtp: Start PTP failed.
********************************************************************************
```

See option 10 to determine whether PTP is running or stopped.

For more information on this command, "bcStartPtp" on page 38

2. Stopping PTP

When the PTP is already running, electing option 2 will return the following:

```
*******************************************************************************
                                PTP
*******************************************************************************
PTP Stopped Successfully!
*******************************************************************************
```

For more information on this command, "bcStopPtp" on page 38

3. PTP Management Messages

```
                        Symmetricom, Inc.
                            PCIe-1000 UI
================================================================================
      1. Start PTP                          2. Stop PTP
      3. PTP Management Messages            4. Network Configuration
      5. Software Upgrade                   6. Version Info
      7. Read Time From Host Memory         8. Read Time From Target Memory
      9. BC Pass-thru Command              10. System Status
      0. Exit the Program
```

The PTP management messages provide vital system information. Select option 3 from the main menu to get the PTP management menu.

```
================================================================================
                          PTP Management Messages
================================================================================
       1. Get Clock Description
       2. Get User Description               3. Set User Description
       4. Get Default Data Set               5. Get Current Data Set
       6. Get Parent Data Set                7. Get Time Properties Data Set
       8. Get Port Data Set
       9. Get Priority1                     10. Set Priority1
      11. Get Priority2                     12. Set Priority2
      13. Get Domain                        14. Set Domain
      15. Enable Port                       16. Disable Port
       0. Back to main menu


      Select Option:
```

3.1 Get Clock Description

Select option 1 for the clock description. Here is an example display:

```
clockType: 0x8800; phyProt: Ethernet using UDP/IPv4
phyAddr: 04:00:00:00:00:0A; netProt: 1; netAddr: 192.168.0.100
manuId: 00 A0 69; prodDesc: Symmetricom;IEEE1588-2008 Norfolk Card;
revData: V0.01.01; UserDesc: Symmetricom IEEE1588-2008 Norfolk Card
ProfId: 001B19000100
********************************************************************************
```

For more information, see the API command "bcGetClockDesc" on page 39

3.2 Get User Description

Select option 2 to get the user description. Here is an example display:

```
********************************************************************************
                          User Description
********************************************************************************
Description: Symmetricom IEEE1588-2008 Norfolk Card
********************************************************************************
```

For more information, see the associated API command; "bcGetUserDesc" on page 39

3.3 Set User Description

Select option 3 to change the user description. This is a text field for entering the new user description.

For more information, see the associated API command: "bcSetUserDesc" on page 39

3.4 Get Default Data Set

Select option 4 to get the default data set. Here is an example display:

```
                          Default Data Set
********************************************************************************
so_tsc: 0x03; nPorts: 1, prio1: 128; clkClass: 0xFF
clkAccu: 0xFE; scalLogV: 18944; prio2: 128; clkId: 04:00:00:FF:FE:00:00:0A; dom: 0
********************************************************************************
```

For more information, see the associated API command:"bcGetDefaultDataSet" on page 40

3.5 Get Current Data Set

Select option 5 to get the current data set. Here is an example display:

```
*********************************************************************************
                              Current Data Set
*********************************************************************************
stepsRemoved: 0; offFromMaster: 0; meanPathDelay: 1163591680
*********************************************************************************
```

For more information, see the associated API command:"bcGetCurrentDataSet" on page 40

3.6 Get Parent Data Set

Select option 6 to get the parent data set. Here is an example display:

```
                              Parent Data Set
*********************************************************************************
parentPortId: 04:00:00:FF:FE:00:00:0A-0; stat: 0x37
offSLogV: 65535; phyChgRate: 2147483647; GMPrio1: 128; GMClkClass: 0xFF
GMClkAccu: 0xFE; GMScalLogV: 18944; GMPrio2: 128; GMId: 04:00:00:FF:FE:00:00:0A
*********************************************************************************
```

For more information, see the associated API command: "bcGetParentDataSet" on page 40

3.7 Get Time Properties Data Set

Select option 7 to get the time properties data set. Here is an example display:

```
*********************************************************************************
                          Time Properties Data Set
*********************************************************************************
utcOffset: 34; flag: 0x00; timeSrc: 0xA0
*********************************************************************************
```

For more information, see the associated API command: "bcGetTimePropertiesDataSet" on page 40

3.8 Get Port Data Set

Select option 8 to get the port data set. Here is an example display:

```
                              Port Data Set
*********************************************************************************
portId: 04:00:00:FF:FE:00:00:0A-1; state: 4
minDelayReqInt: -5; peerMeanPDelay: 0, announceInt: 1, annReceiptTimeout: 3
syncInt: -5; delayMech: 0x01; minPDelayReqInt: -5, reservedVerNum: 0x02
*********************************************************************************
```

For more information, see the associated API command: "bcGetPortDataSet" on page 41

3.9 Get Priority1

Select option 9 to get the priority 1 information. Here is an example display:

```
*******************************************************************************
                            Priority 1
*******************************************************************************
priority: 128
*******************************************************************************
```

For more information, see the associated API command: "bcGetPriority1" on page 41

3.10 Set Priority1

Select option 10 to set the priority 1 information. Here is an example display:

```
       Select Option: 10
New Priority1 (0 - 255):
```

For more information, see the associated API command: "bcSetPriority1" on page 41

3.11 Get Priority2

Select option 11 to get the priority 2 information. Here is an example display:

```
*******************************************************************************
                            Priority 2
*******************************************************************************
priority: 128
*******************************************************************************
```

For more information, see the associated API command: "bcGetPriority2" on page 42

3.12 Set Priority2

Select option 12 to set the priority 2 information. Here is an example display:

```
       Select Option: 12
New Priority2 (0 - 255):
```

For more information, see the associated API command: "bcSetPriority2" on page 42

3.13 Get Domain

Select option 13 to get the domain information. Here is an example display:

```
*******************************************************************************
                            Domain
*******************************************************************************
Domain Number: 0
*******************************************************************************
```

For more information, see the associated API command: "bcGetDomain" on page 42

3.14 Set Domain

Select option 14 to set the domain information. Here is an example display:

```
      Select Option: 14
New Domain (0 - 255):
```

For more information, see the associated API command: "bcSetDomain" on page 43

3.15 Enable Port

Select option 15 to enable the port. Here is an example display:

```
      Select Option: 15
MANAGMENT ERROR STATUS: tlv: 2; length: 30; errId: 0xfffe; Id: 0x200d
                        displayData: Port is not disabled
```

For more information, see the associated API command: "bcCmdEnablePort" on page 43

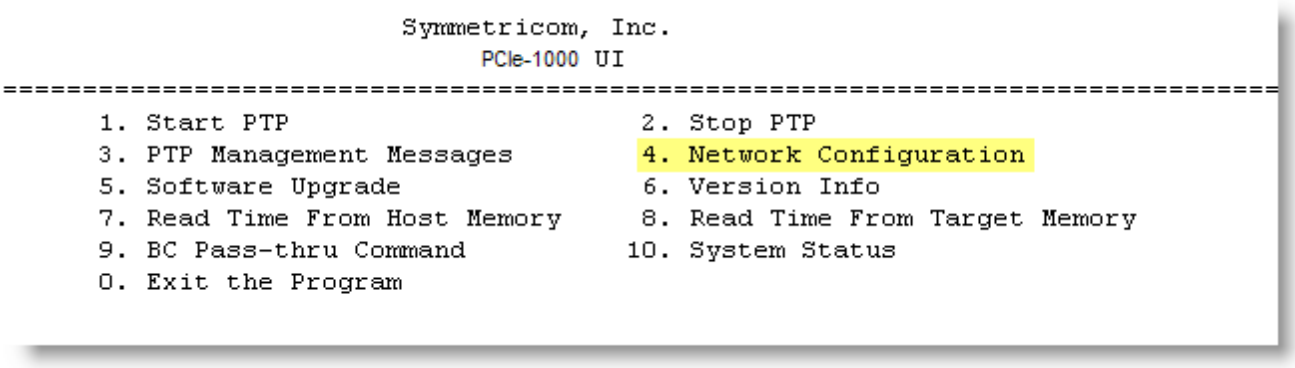
3.16 Disable Port

Select option 16 to disable the port. Here is an example display:

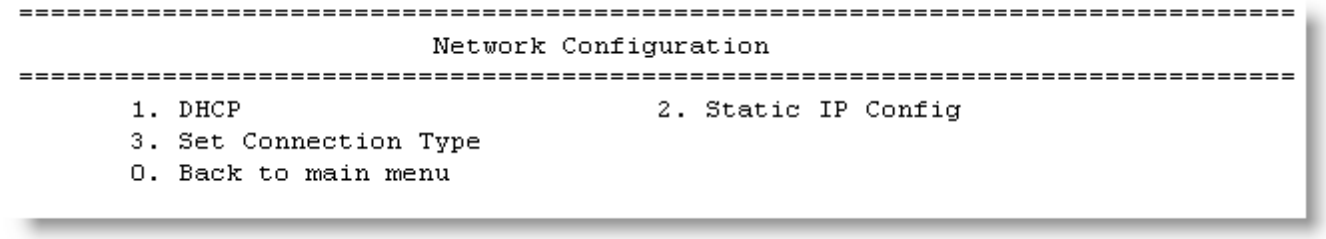
```
NO MESSAGE BACK!!!
```

For more information, see the associated API command: "bcCmdDisablePort" on page 43

4. Network Configuration



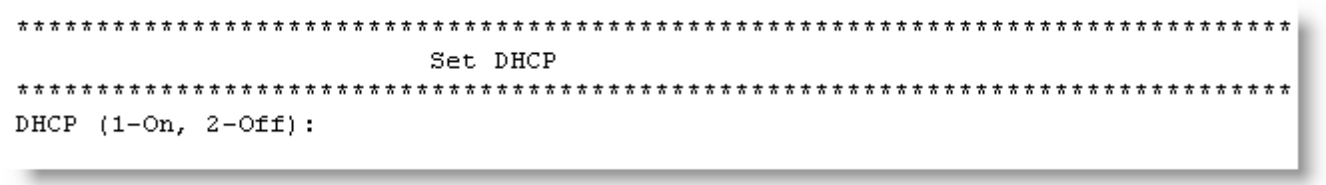
Select option 4 from the main menu to display the network Configuration menu as follows:



To find out more select from the following:

4.1 Set DHCP

Select option 1 from the network Configuration menu to turn DHCP On or Off



For more information, see the associated API command: "bcSetDHCP" on page 46

4.2 Static IP Config

Select option 2 from the network Configuration menu to set up the static IP configuration. The following display has no values allocated to the parameters. In practice a value would be entered for each of the parameters shown in the display below.

```
*******************************************************************************
                          Set Network Configuration
*******************************************************************************
Port IP Address:
Netmask:
Gateway IP Address:
Update Network Settings? [y/n]:
```

For more information, see the associated API command: "bcSetNetworkConfig" on page 46

4.3 Set Connection Type

Select option 3 from the network Configuration menu to set the connection type.

Enter:

1.For a crossover cable 32 Pkt/sec.

2.For a switch, 32 Pkt/sec.

3. For a switch,1 Pkt/sec.

At the prompt for calibration values enter the following:

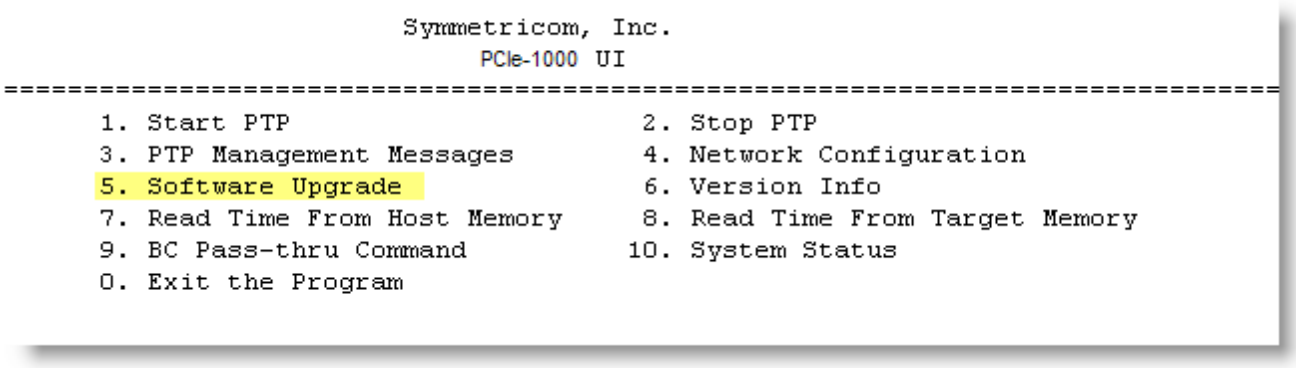
Crossover 32 pkt/sec	PPS100BTCal = 535	PPS1000BTCal = 0
Switch 32 pkt/sec	PPS100BTCal = 3770	PPS1000BTCal = 6750
Switch 1 pkt/sec	PPS100BTCal = 3770	PPS1000BTCal = 6750

In the follwing example, selection 2 has been made

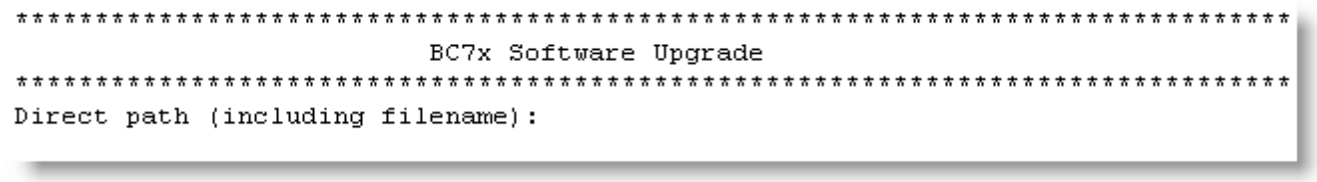
```
*******************************************************************************
                          Set Connection Type
*******************************************************************************
        Connect Type (1-Crossover 32 Pkt/sec, 2-Switch 32 Pkt/sec, 3-Switch 1 Pkt/Sec): 2
```

For more information, see the associated API command: "bcSetNetworkConnectType" on page 46

5. Software Upgrade

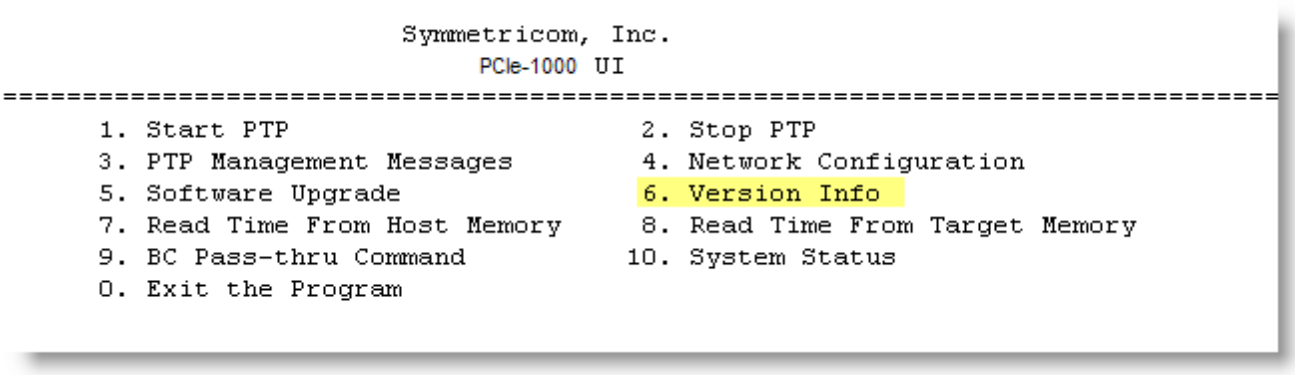


Select option 5 from the main menu to upgrade the PCIe card software. Enter the path to the upgrade software.



For more information, see the associated API command: "bcSoftwareUpdate" on page 49

6. Version Information



Select option 6 from the main menu to obtain system version information.

Note, the following display is an example only. Your version information will most likely be different.

For more information, see the associated API command: "bcGetVersion" on page 45

Also see:

- "bcGetModel" on page 44
- "bcGetSerialNumber" on page 45

7. Read Time From Host Memory

```
                        Symmetricom, Inc.
                            PCIe-1000 UI
================================================================================
      1. Start PTP                          2. Stop PTP
      3. PTP Management Messages            4. Network Configuration
      5. Software Upgrade                   6. Version Info
      7. Read Time From Host Memory         8. Read Time From Target Memory
      9. BC Pass-thru Command              10. System Status
      0. Exit the Program
```

For additional information, "Reading Time" on page 12

Select option 7 from the main menu to obtain read time from host memory display.

```
================================================================================
                          Read Time From Host Memory
================================================================================
       1. Read Binary Time                   2. Read Binary Time-No Sync
       3. Read Binary Time-Counter           4. Read Binary Time-Counter No Sync
       0. Return To Main Menu
```

To lean more, select from the following:

7.1 Read Binary Time

Select option 1 to read binary time from the Host Memory.

```
********************************************************************************
                          Time From Host Memory
********************************************************************************
Major Time[2724] Minor Time[987072710]
********************************************************************************
```

For more information, see the associated API command: "bcReadBinTime" on page 50

7.2 Time From Host Memory-No Sync

Select option 2 to read binary time from the host memory-no sync.

```
*********************************************************************************
                          Time From Host Memory-No Sync
*********************************************************************************
Major Time[2744] Minor Time[11802360]
*********************************************************************************
```

For more information, see the associated API command: "bcReadBinTimeNoSync" on page 50

7.3 Time From Host Memory-Counter

Select option 3 to read binary time from the host memory-counter.

```
*********************************************************************************
                          Time From Host Memory-Counter
*********************************************************************************
Counter [2761771692980]
*********************************************************************************
```

For more information, see the associated API command: "bcReadBinCounter" on page 51

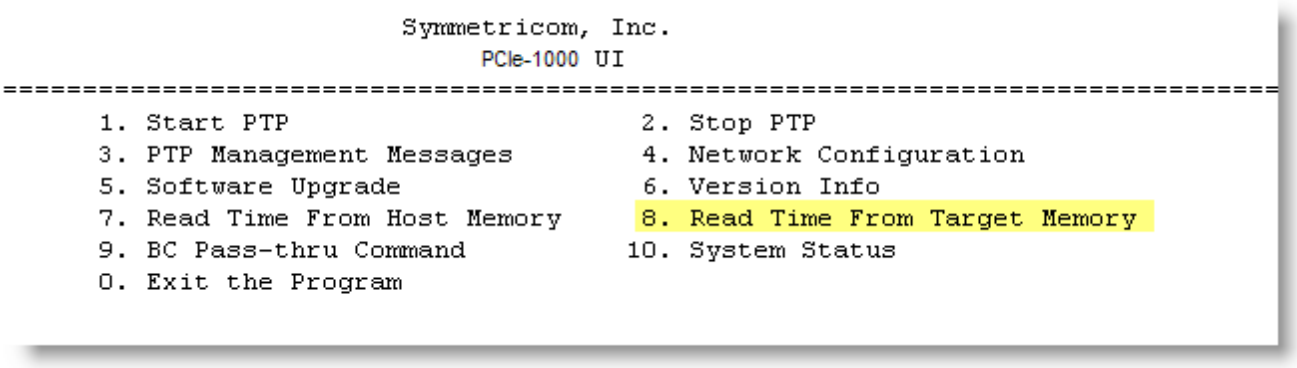
7.4 Time From Host Memory-Counter-No Sync

Select option 4 to read binary time from the host memory-counter-no sync.

```
*********************************************************************************
                          Time From Host Memory-Counter-No Sync
*********************************************************************************
Counter [2796219235940]
*********************************************************************************
```

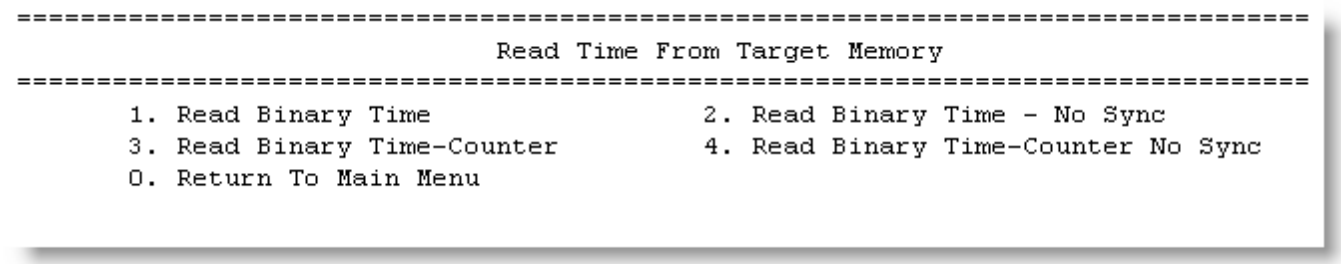
For more information, see the associated API command: "bcReadBinCounterNoSync" on page 52

8. Read Time From Target Memory



For additional information, "Reading Time" on page 12

Select option 8 from the main menu to obtain Read Time from Target Memory display.



To lean more, select from the following:

8.1 Read Binary Time From Target Memory

Select option 1 to read binary time from the target memory.

```
********************************************************************************
                          Time From Target Memory
********************************************************************************
Major Time[2849] Minor Time[995444600]
********************************************************************************
```

For more information, see the associated API command: "bcReadRegBinTime" on page 52

8.2 Read Binary Time From Target Memory-No Sync

Select option 2 to read binary time from the target memory with no sync.

```
********************************************************************************
                          Time From Target Memory-No Sync
********************************************************************************
Major Time[2874] Minor Time[347051950]
********************************************************************************
```

For more information, see the associated API command: "bcReadRegBinTimeNoSync" on page 52

8.3 Read Binary Time From Target Memory-Counter

Select option 3 to read binary time-counter from the target memory counter.

```
********************************************************************************
                          Time From Target Memory-Counter
********************************************************************************
Counter [2891819993120]
********************************************************************************
```

For more information, see the associated API command: "bcReadRegBinCounter" on page 54

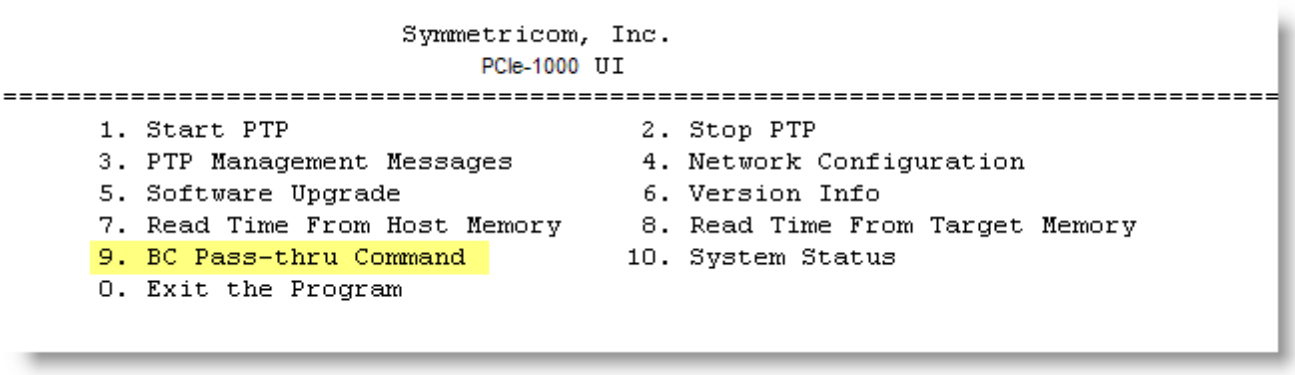
8.4 Read Binary Time From Target Memory-Counter No Sync

Select option 4 to read binary time-counter from the target memory counter with no sync.

```
*******************************************************************************
                          Time From Target Memory-Counter-No Sync
*******************************************************************************
Counter [2910684605730]
*******************************************************************************
```

For more information, see the associated API command: "bcReadRegBinCounterNoSync" on page 54

9. BC Pass-thru Command



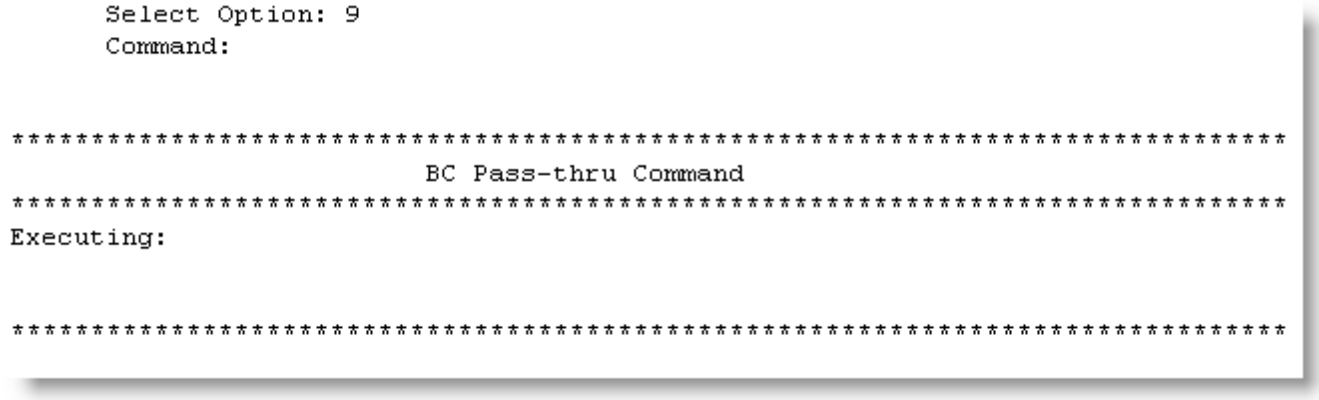
The PCIe-1000 is running its own Linux OS. The pass-thru command allows the host user to issue shell commands (through the API) to the PCIe-1000. It's mostly for debugging purpose. For example, if you want to issue a "ls" command to the PCIe-1000, you would choose "9. BC Pass-thru Command" from the cli_demo program, then enter "ls".

Alternatively, if the user doesn't want to use cli_demo, they can issue this command through the API:

BOOL bcExecCli(BC_PCI_HANDLE hPCI, PCHAR pCmdStr);

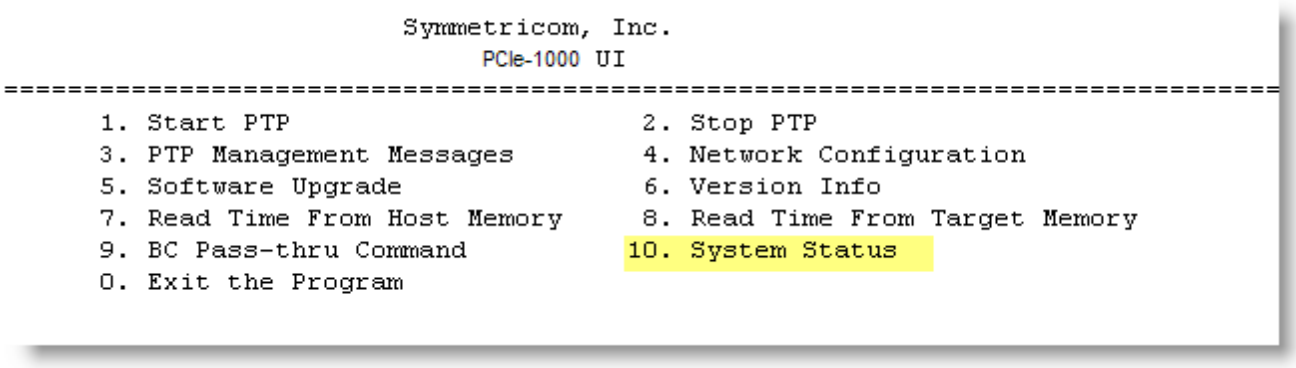
See the **bcExecCli** command in the "Developing Programs using the API" on page 14

Select option 9 from the main menu to enter a pass-thru command. At the command prompt, enter the command and click **Enter**.

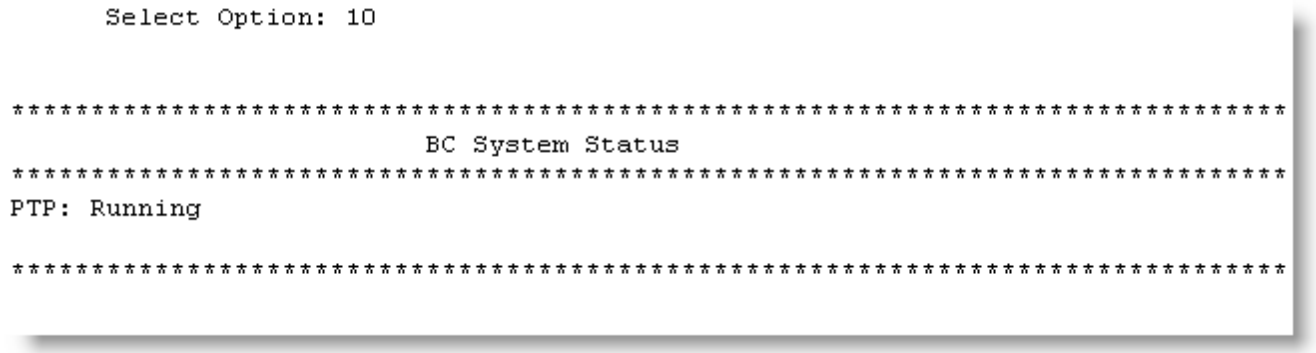


For more information, see the associated API command: "bcExecCli" on page 44

10. System Status



Select option 10 from the main menu to display the system status. Use this option to determine if PTP is running or stopped.



For more information, see the associated API commands: "Functions for system status" on page 43

Linux API Library Definitions

The interface library provides functions for each of the programming packets supported by the PCIe-1000 card.

PCIe-1000 API Functions

To learn more, select from the following:

Functions to open/close/initialize the device

To learn more, select from the following:

bcOpenPCI	
Syntax	BC_PCI_HANDLE bcOpenPCI(INT nDev);
Description	This function opens the bcpci device and memory maps the device's corresponding PCI BAR resources and DMA area. The _tagBC_PCI_HANDLE structure is malloced and its reference is returned to the caller. This function should be called before any other API functions can be executed. The BC Error String is also reset to "blank".
Parameters	nDev (IN) – Device number for the PCIe-1000 device
Return value	BC_PCI_HANDLE - Pointer to _tagBC_PCI_HANDLE structure corresponding to given device number.
Remarks	

bcClosePCI	
Syntax	VOID bcClosePCI(BC_PCI_HANDLE hPCI);
Description	This function closes the bcpci device associated with given BC_PCI_HANDLE. Unmaps corresponding PCI BAR resources and DMA area. All memory read mutexes and file descriptors are destroyed and freed. The BC Error String is also reset to "blank".
Parameters	hPCI (IN) - PCI handle for the PCIe-1000 device
Return value	TRUE - function succeeded FALSE - an error occurred
Remarks	It is recommended bcInit be called right after bcOpenPCI.

bcInit	
Syntax	BOOL bcInit(BC_PCI_HANDLE hPCI)
Description	This function establishes the communication between host and PCIe-1000.
Parameters	hPCI (IN) - PCI handle for the PCIe-1000 device
Return value	None
Remarks	

Function to report last error

bcGetLastErrorString	
Syntax	PCHAR bcGetLastErrorString();
Description	This function retrieves the last PCIe-1000 error string. After a PCIe-1000 API is called, which returned an error code, caller can use this function to retrieve the error string. Error string is reset to blank upper calls to bcOpenPCI and bcClosePCI.
Parameters	None
Return value	PCHAR - Pointer to error string.
Remarks	

Functions to configure PTP

To learn more, select from the following:

bcStartPtp	
Syntax	BOOL bcStartPtp(BC_PCI_HANDLE hPCI);
Description	This function starts the PTP process on PCIe-1000. PTP process must be started prior to calling subsequent PTP management message APIs.
Parameters	hPCI (IN) - PCI handle for the PCIe-1000 device
Return value	TRUE - function succeeded FALSE - an error occurred (including if PTP process has already started)
Remarks	

bcStopPtp	
Syntax	BOOL bcStopPtp(BC_PCI_HANDLE hPCI);
Description	This function stops the PTP process on PCIe-1000. Subsequent PTP management message APIs will fail while PTP process is not running.

bcStopPtp	
Parameters	hPCI (IN) - PCI handle for the PCIe-1000 device
Return value	TRUE - function succeeded FALSE - an error occurred
Remarks	

bcGetClockDesc	
Syntax	BOOL bcGetClockDesc(BC_PCI_HANDLE hPCI, PBC_CLOCK_DESC pClock-Desc);
Description	This function retrieves the PTP clock description. Refer to IEEE 1588-2008 specification document for details of fields from BC_CLOCK_DESC type.
Parameters	hPCI (IN) - PCI handle for the PCIe-1000 device pClockDesc (OUT) - Pointer to BC_CLOCK_DESC structure to store PTP clock description info.
Return value	TRUE - function succeeded FALSE - an error occurred
Remarks	

bcGetUserDesc	
Syntax	BOOL bcGetUserDesc(BC_PCI_HANDLE hPCI, PCHAR pDesc);
Description	This function retrieves the PTP user description. Caller must allocate char array to hold user description string.
Parameters	hPCI (IN) - PCI handle for the PCIe-1000 device pDesc (OUT) - Pointer to character array to store user description.
Return value	TRUE - function succeeded FALSE - an error occurred
Remarks	

bcSetUserDesc	
Syntax	BOOL bcSetUserDesc(BC_PCI_HANDLE hPCI, PCHAR pDesc);
Description	This function sets the PTP user description provided by the caller.
Parameters	hPCI (IN) - PCI handle for the PCIe-1000 device pDefaultDataSet (OUT) - Pointer to BC_DEFAULT_DATASET structure to store PTP default dataset info.
Return value	TRUE - function succeeded FALSE - an error occurred
Remarks	

bcGetDefaultDataSet	
Syntax	BOOL bcGetDefaultDataSet(BC_PCI_HANDLE hPCI, PBC_DEFAULT_DATASET pDefaultDataSet);
Description	This function retrieves the PTP Default Data Set. Refer to IEEE 1588-2008 specification document for details of fields from BC_DEFAULT_DATASET type.
Parameters	hPCI (IN) - PCI handle for the PCIe-1000 device pParentDataset (OUT) - Pointer to BC_PARENT_DATASET structure to store PTP parent dataset info.
Return value	TRUE - function succeeded FALSE - an error occurred
Remarks	

bcGetCurrentDataSet	
Syntax	BOOL bcGetCurrentDataSet(BC_PCI_HANDLE hPCI, PBC_CURRENT_DATASET pCurrentDataset);
Description	This function retrieves the PTP Current Data Set. Refer to IEEE 1588-2008 specification document for details of fields from BC_CURRENT_DATASET type.
Parameters	hPCI (IN) - PCI handle for the PCIe-1000 device pCurrentDataset (OUT) - Pointer to BC_CURRENT_DATASET structure to store PTP current dataset info.
Return value	TRUE - function succeeded FALSE - an error occurred
Remarks	

bcGetParentDataSet	
Syntax	BOOL bcGetParentDataSet(BC_PCI_HANDLE hPCI, PBC_PARENT_DATASET pParentDataset);
Description	This function retrieves the PTP Parent Data Set. Refer to IEEE 1588-2008 specification document for details of fields from BC_PARENT_DATASET type.
Parameters	hPCI (IN) - PCI handle for the PCIe-1000 device pDesc (IN) - Pointer to user description string
Return value	TRUE - function succeeded FALSE - an error occurred
Remarks	

bcGetTimePropertiesDataSet	
Syntax	BOOL bcGetTimePropertiesDataSet(BC_PCI_HANDLE hPCI, PBC_TIME_

bcGetTimePropertiesDataSet	
	PROPERTIES_DATASET pTimePropDataset);
Description	This function retrieves the PTP Time Properties Data Set. Refer to IEEE 1588-2008 specification document for details of fields from BC_TIME_PROP-ERTIES_DATASET type.
Parameters	hPCI (IN) - PCI handle for the PCIe-1000 device pTimePropDataset (OUT) - Pointer to BC_TIME_PROPERTIES_DATASET structure to store PTP time properties dataset info.
Return value	TRUE - function succeeded FALSE - an error occurred
Remarks	

bcGetPortDataSet	
Syntax	BOOL bcGetPortDataSet(BC_PCI_HANDLE hPCI, PBC_PORT_DATASET pPort-Dataset);
Description	This function retrieves the PTP Port Data Set. Refer to IEEE 1588-2008 specification document for details of fields from BC_PORT_DATASET type.
Parameters	hPCI (IN) - PCI handle for the PCIe-1000 device pPortDataset (OUT) - Pointer to BC_PORT_DATASET structure to store PTP port data-set info.
Return value	TRUE - function succeeded FALSE - an error occurred
Remarks	

bcGetPriority1	
Syntax	BOOL bcGetPriority1(BC_PCI_HANDLE hPCI, PUINT priority1);
Description	This function retrieves PTP priority1 value, which should be the value of BC_ DEFAULT_DATASET.priority1 member of the default dataset.
Parameters	hPCI (IN) - PCI handle for the PCIe-1000 device priority1 (IN) - Priority1 value to set.
Return value	TRUE - function succeeded FALSE - an error occurred
Remarks	

bcSetPriority1	
Syntax	BOOL bcSetPriority1(BC_PCI_HANDLE hPCI, UINT priority1);
Description	This function sets PTP priority1 value, which should be the value of BC_DEFAULT_ DATASET.priority1 member of the default dataset.

bcSetPriority1	
Parameters	hPCI (IN) - PCI handle for the PCIe-1000 device priority1 (IN) - Priority1 value to set.
Return value	TRUE - function succeeded FALSE - an error occurred
Remarks	

bcGetPriority2	
Syntax	BOOL bcGetPriority2(BC_PCI_HANDLE hPCI, PUINT priority2);
Description	This function retrieves PTP priority2 value, which should be the value of BC_DEFAULT_DATASET.priority2 member of the default dataset.
Parameters	hPCI (IN) - PCI handle for the PCIe-1000 device priority2 (OUT) - Pointer to PUINT to store priority2 info.
Return value	TRUE - function succeeded FALSE - an error occurred
Remarks	

bcSetPriority2	
Syntax	BOOL bcGetPriority2(BC_PCI_HANDLE hPCI, PUINT priority2);
Description	This function retrieves PTP priority2 value, which should be the value of BC_DEFAULT_DATASET.priority2 member of the default dataset.
Parameters	hPCI (IN) - PCI handle for the PCIe-1000 device priority2 (OUT) - Pointer to PUINT to store priority2 info.
Return value	TRUE - function succeeded FALSE - an error occurred
Remarks	

bcGetDomain	
Syntax	BOOL bcGetDomain(BC_PCI_HANDLE hPCI, PUINT domain);
Description	This function retrieves PTP domain number, which should be the value of BC_DEFAULT_DATASET.domainNumber member of the default dataset.
Parameters	hPCI (IN) - PCI handle for the PCIe-1000 device domain (OUT) - Pointer to PUINT to store domain number.
Return value	TRUE - function succeeded FALSE - an error occurred
Remarks	

bcSetDomain	
Syntax	BOOL bcSetDomain(BC_PCI_HANDLE hPCI, UINT domain);
Description	This function sets PTP domain number, which should be the value of BC_DEFAULT_DATASET.domainNumber member of the default dataset.
Parameters	hPCI (IN) - PCI handle for the PCIe-1000 device domain (IN) - Domain value to set
Return value	TRUE - function succeeded FALSE - an error occurred
Remarks	

bcCmdEnablePort	
Syntax	BOOL bcCmdEnablePort(BC_PCI_HANDLE hPCI);
Description	This function issues an enable port command to PTP process. In an ordinary and boundary clock, the receipt of an enable port message shall cause the DESIGNATED_ENABLED event to occur. Refer to IEEE 1588-2008 specification document for details.
Parameters	hPCI (IN) - PCI handle for the PCIe-1000 device
Return value	TRUE - function succeeded FALSE - an error occurred
Remarks	

bcCmdDisablePort	
Syntax	BOOL bcCmdEnablePort(BC_PCI_HANDLE hPCI);
Description	This function issues an enable port command to PTP process. In an ordinary and boundary clock, the receipt of an enable port message shall cause the DESIGNATED_ENABLED event to occur. Refer to IEEE 1588-2008 specification document for details.
Parameters	hPCI (IN) - PCI handle for the PCIe-1000 device
Return value	TRUE - function succeeded FALSE - an error occurred
Remarks	

Functions for system status

bcGetSystemStatus	
Syntax	BOOL bcGetSystemStatus(BC_PCI_HANDLE hPCI, PBC_SYSTEM_STATUS pSysStatus);
Description	This function retrieves the PCIe-1000 device's system status.
Parameters	hPCI (IN) - PCI handle for the PCIe-1000 device

bcGetSystemStatus	
	pSysStatus (OUT) - Pointer to PBC_SYSTEM_STATUS structure to store system status information.
Return value	TRUE - function succeeded FALSE - an error occurred
Remarks	Currently, only PTP status is available. More will be added in future releases.

Functions to execute CLI on PCIe-1000

bcExecCli	
Syntax	BOOL bcExecCli(BC_PCI_HANDLE hPCI, PCHAR pCmdStr);
Description	This function allows caller to execute a local shell command on the PCIe-1000 device. It is essentially a pass-thru command which can be used for debugging (For example: Concatenate system log files on the PCIe-1000 device by issuing the command string "cat /var/log/messages").
Parameters	hPCI (IN) - PCI handle for the PCIe-1000 device pCmdStr (IN) - Pointer to command string to be executed.
Return value	TRUE - function succeeded FALSE - an error occurred
Remarks	

Functions for querying manufacturing data

To learn more, select from the following:

bcGetModel	
Syntax	BOOL bcGetModel(BC_PCI_HANDLE hPCI, PCHAR pModel);
Description	This function queries PCIe-1000 model number. Memory is allocated by caller. The input char array should be 32 bytes or more.
Parameters	hPCI (IN) - PCI handle for the PCIe-1000 device pModel (OUT) - Pointer to char array to hold model number.
Return value	TRUE - function succeeded FALSE - an error occurred
Remarks	

bcGetSerialNumber	
Syntax	BOOL bcGetSerialNumber(BC_PCI_HANDLE hPCI, PCHAR pSerial);
Description	This function queries PCIe-1000 serial number. Memory is allocated by caller. The input char array should be 32 bytes or more.
Parameters	hPCI (IN) - PCI handle for the PCIe-1000 device pSerial (OUT) - Pointer to char array to hold serial number.
Return value	TRUE - function succeeded FALSE - an error occurred
Remarks	

bcGetVersion	
Syntax	BOOL bcGetVersion(BC_PCI_HANDLE hPCI, PBC_VERSION pVersion);
Description	This function queries PCIe-1000 version number. Memory is allocated by caller. The pointer to BC_VERSION must be passed to get version info.
Parameters	hPCI (IN) - PCI handle for the PCIe-1000 device pVersion (OUT) - Pointer to BC_VERSION to hold version info.
Return value	TRUE - function succeeded FALSE - an error occurred
Remarks	

Functions for network configuration

To learn more, select from the following:

bcSetDHCP	
Syntax	BOOL bcSetDHCP(BC_PCI_HANDLE hPCI, BOOL enable);
Description	This function sets DHCP on the PCIe-1000 device. If DHCP is set to FALSE, the stored static IP configuration on the device will be used.
Parameters	hPCI (IN) - PCI handle for the PCIe-1000 device enable (IN) - TRUE to enable DHCP. FALSE to disable DHCP.
Return value	TRUE - function succeeded FALSE - an error occurred
Remarks	

bcSetNetworkConfig	
Syntax	BOOL bcSetNetworkConfig(BC_PCI_HANDLE hPCI, PBC_NET_CONFIG config);
Description	This function allows caller to configure network settings for the PCIe-1000 device. The port IP, netmask, and gateway IP is set and the corresponding network interface is brought back up. If DHCP was enabled, it will be disabled after this call and the static IP settings will take effect.
Parameters	hPCI (IN) - PCI handle for the PCIe-1000 device. config (IN) - Pointer to BC_NET_CONFIG structure containing static network configuration info.
Return value	TRUE - function succeeded FALSE - an error occurred
Remarks	

bcSetNetworkConnectType	
Syntax	BOOL bcSetNetworkConnectType(BC_PCI_HANDLE hPCI, eBC_NET_CONN_TYPE connectType, INT pps100BTCal, INT pps1000BTCal);
Description	This function allows caller to set the network connection between the PCIe-1000 device and PTP grandmaster. Set connection type: • BC_CROSSOVER_32_PKT_PER_SEC - if PCIe-1000 is connected to the grandmaster through a crossover cable at 32 packets per second of sync interval.

bcSetNetworkConnectType	
	• BC_SWITCH_32_PKT_PER_SEC - if PCIe-1000 is connected to the grandmaster through a switch at 32 packets per second of sync interval. • BC_SWITCH_1_PKT_PER_SEC - if PCIe-1000 is connected to the grand-master through a switch at 1 packets per second of sync interval.
Parameters	• hPCI (IN) - PCI handle for the PCIe-1000 device • connectType (IN) - Connection type between PCIe-1000 and PTP grand-master. • pps100BTCal (IN) - PPS 100 base T calibration parameter. • pps1000BTCal (IN) - PPS 1000 base T calibration parameter.
Return value	TRUE - function succeeded FALSE - an error occurred
Remarks	

Functions to control 1PPS output

To learn more, select from the following:

bcGetPPSOutputEnable	
Syntax	BOOL bcGetPPSOutputEnable(BC_PCI_HANDLE hPCI, PBOOL enable);
Description	This function retrieves current status of PPS output enable.
Parameters	hPCI (IN) - PCI handle for the PCIe-1000 device enable (OUT) - Pointer to BOOL of PPS out enable. TRUE is PPS output enabled. FALSE is PPS output disabled.
Return value	TRUE - function succeeded FALSE - an error occurred
Remarks	

bcSetPPSOutputEnable	
Syntax	BOOL bcSetPPSOutputEnable(BC_PCI_HANDLE hPCI, BOOL enable);
Description	This function sets the PPS output enable.
Parameters	hPCI (IN) - PCI handle for the PCIe-1000 device enable (IN) - Value to set the PPS out enable. TRUE is PPS output enabled. FALSE is PPS output disabled.
Return value	TRUE - function succeeded FALSE - an error occurred
Remarks	

bcGetPPSOutputMode	
Syntax	BOOL bcGetPPSOutputMode(BC_PCI_HANDLE hPCI, PBC_PPS_OUTPUT mode);
Description	This function retrieves current status of PPS output mode.
Parameters	hPCI (IN) - PCI handle for the PCIe-1000 device mode (OUT) - Pointer to BC_PPS_OUTPUT enum of PPS output mode. BC_PPS_60US_PULSE mode if PPS output is set to 60usec pulse.

bcGetPPSOutputMode	
	BC_PPS_50PERCENT_DUTY mode if PPS output is set to 50% duty cycle.
Return value	TRUE - function succeeded FALSE - an error occurred
Remarks	

bcSetPPSOutputMode	
Syntax	BOOL bcSetPPSOutputMode(BC_PCI_HANDLE hPCI, BC_PPS_OUTPUT mode);
Description	This function sets the PPS output mode.
Parameters	hPCI (IN) - PCI handle for the PCIe-1000 device mode (IN) - BC_PPS_OUTPUT enum of PPS output mode. BC_PPS_60US_PULSE mode sets PPS output to 60usec pulse. BC_PPS_50PERCENT_DUTY mode sets PPS output to 50% duty cycle.
Return value	TRUE - function succeeded FALSE - an error occurred
Remarks	

Functions to update software

bcSoftwareUpdate	
Syntax	BOOL bcSoftwareUpdate(BC_PCI_HANDLE hPCI, PCHAR pUpdateStr);
Description	This function allows caller to perform software update of the PCIe-1000 device.
Parameters	hPCI (IN) - PCI handle for the PCIe-1000 device pUpdateStr (IN) - Pointer to upgrade character string. Update string should include direct path and filename. (For example: http://www.symmetricom.com/filename.tgz)
Return value	TRUE - function succeeded FALSE - an error occurred
Remarks	

Functions to read time

To learn more, select from the following:

Read binary time from the host memory

bcReadBinTime	
Syntax	BOOL bcReadBinTime(BC_PCI_HANDLE hPCI, PUINT pMajor, PUINT pMinor);
Description	This function performs synchronous read of binary time from the DMA'ed MPC8308 time counter. Synchronize call by using memory read mutex on 32-bit systems. On 64 bit platform, the read is atomic. The returned binary time is in the UTC time scale.
Parameters	hPCI (IN) - PCI handle for the PCIe-1000 device pMajor (OUT) - Major part of the binary time (seconds) pMinor (OUT) - Minor part of the binary time (nanoseconds)
Return value	TRUE - function succeeded FALSE - an error occurred
Remarks	

bcReadBinTimeNoSync	
Syntax	BOOL bcReadBinTimeNoSync(BC_PCI_HANDLE hPCI, PUINT pMajor, PUINT pMinor);
Description	This function performs nonatomic read of binary time from the DMA'ed MPC8308 time counter. Because of its nonatomic nature, caller should insure the validity of each read by comparing subsequent read values. The returned binary time is in the UTC time scale.
Parameters	hPCI (IN) - PCI handle for the PCIe-1000 device pMajor (OUT) - Major part of the binary time (seconds) pMinor (OUT) - Minor part of the binary time (nanoseconds)
Return value	TRUE - function succeeded FALSE - an error occurred
Remarks	

bcReadPTPBinTime	
Syntax	BOOL bcReadPTPBinTime(BC_PCI_HANDLE hPCI, PUINT pMajor, PUINT pMinor);
Description	This function performs synchronous read of binary time from the DMA'ed MPC8308 time counter. Synchronize call by using memory read mutex on 32-bit systems. On 64 bit platform, the read is atomic. The returned binary time is in the same time scale as the grandmaster to which this board currently synchronizes.
Parameters	hPCI (IN) - PCI handle for the PCIe-1000 device pMajor (OUT) - Major part of the binary time (seconds) pMinor (OUT) - Minor part of the binary time (nanoseconds)
Return value	TRUE - function succeeded FALSE - an error occurred
Remarks	

bcReadPTPBinTimeNoSync	
Syntax	BOOL bcReadPTPBinTimeNoSync(BC_PCI_HANDLE hPCI, PUINT pMajor, PUINT pMinor)
Description	This function performs nonatomic read of binary time from the DMA'ed MPC8308 time counter. Because of its nonatomic nature, caller should insure the validity of each read by comparing subsequent read values. The returned binary time is in the same time scale as the grandmaster to which this board currently synchronizes.
Parameters	hPCI (IN) - PCI handle for the PCIe-1000 device pMajor (OUT) - Major part of the binary time (seconds) pMinor (OUT) - Minor part of the binary time (nanoseconds)
Return value	TRUE - function succeeded FALSE - an error occurred
Remarks	

Read binary time counter from the host memory

bcReadBinCounter	
Syntax	BOOL bcReadBinCounter(BC_PCI_HANDLE hPCI, PUINT64 pCounter);
Description	This function performs synchronous read of 64-bit binary time counter from the DMA'ed MPC8308 time counter. Synchronize call by using memory read mutex on 32-bit systems. On 64 bit platform, the read is atomic. The epoch of the counter is the same as the epoch of the grandmaster to which this board currently synchronizes.
Parameters	hPCI (IN) - PCI handle for the PCIe-1000 device pCounter (OUT) - Pointer to 64-bit time counter value (nanoseconds)

bcReadBinCounter	
Return value	TRUE - function succeeded FALSE - an error occurred
Remarks	

bcReadBinCounterNoSync	
Syntax	BOOL bcReadBinCounterNoSync(BC_PCI_HANDLE hPCI, PUINT64 pCounter);
Description	This function performs nonatomic read of 64-bit binary time counter from the DMA'ed MPC8308 time counter. Because of its nonatomic nature, caller should insure the validity of each read by comparing subsequent read values. The epoch of the counter is the same as the epoch of the grandmaster to which this board currently synchronizes.
Parameters	hPCI (IN) - PCI handle for the PCIe-1000 device pCounter (OUT) - Pointer to 64-bit time counter value (nanoseconds)
Return value	TRUE - function succeeded FALSE - an error occurred
Remarks	

Read binary time from the target registers

bcReadRegBinTime	
Syntax	BOOL bcReadRegBinTime(BC_PCI_HANDLE hPCI, PUINT pMajor, PUINT pMinor);
Description	This function performs synchronous read of binary time from the MPC8308 time count registers through the PCIe bus. Synchronize call using memory read mutex. The returned binary time is in the UTC time scale.
Parameters	hPCI (IN) - PCI handle for the PCIe-1000 device pMajor (OUT) - Major part of the binary time (seconds) pMinor (OUT) - Minor part of the binary time (nanoseconds)
Return value	TRUE - function succeeded FALSE - an error occurred
Remarks	

bcReadRegBinTimeNoSync	
Syntax	BOOL bcReadRegBinTimeNoSync(BC_PCI_HANDLE hPCI, PUINT pMajor, PUINT pMinor);
Description	This function performs nonatomic read of binary time from the MPC8308 time count registers through the PCIe bus. Because of its nonatomic nature, caller should insure the validity of each read by comparing subsequent read values. The

bcReadRegBinTimeNoSync	
	returned binary time is in the UTC time scale.
Parameters	hPCI (IN) - PCI handle for the PCIe-1000 device pMajor (OUT) - Major part of the binary time (seconds) pMinor (OUT) - Minor part of the binary time (nanoseconds)
Return value	TRUE - function succeeded FALSE - an error occurred
Remarks	

bcReadPTPRegBinTime	
Syntax	BOOL bcReadPTPRegBinTime(BC_PCI_HANDLE hPCI, PUINT pMajor, PUINT pMinor);
Description	This function performs synchronous read of binary time from the MPC8308 time count registers through the PCIe bus. Synchronize call using memory read mutex. The returned binary time is in the same time scale as the grandmaster to which this board currently synchronizes.
Parameters	hPCI (IN) - PCI handle for the PCIe-1000 device pMajor (OUT) - Major part of the binary time (seconds) pMinor (OUT) - Minor part of the binary time (nanoseconds)
Return value	TRUE - function succeeded FALSE - an error occurred
Remarks	

bcReadPTPRegBinTimeNoSync	
Syntax	BOOL bcReadPTPRegBinTimeNoSync(BC_PCI_HANDLE hPCI, PUINT pMajor, PUINT pMinor);
Description	This function performs nonatomic read of binary time from the MPC8308 time count registers through the PCIe bus. Because of its nonatomic nature, caller should insure the validity of each read by comparing subsequent read values. The returned binary time is in the same time scale as the grandmaster to which this board currently synchronizes.
Parameters	hPCI (IN) - PCI handle for the PCIe-1000 device pMajor (OUT) - Major part of the binary time (seconds) pMinor (OUT) - Minor part of the binary time (nanoseconds)
Return value	TRUE - function succeeded FALSE - an error occurred
Remarks	

Read binary time counter from the target registers

bcReadRegBinCounter	
Syntax	BOOL bcReadRegBinCounter(BC_PCI_HANDLE hPCI, PUINT64 pCounter);
Description	This function performs synchronous read of 64-bit binary time counter from the MPC8308 registers through the PCIe bus. Synchronize call using memory read mutex. The epoch of the counter is the same as the epoch of the grandmaster to which this board currently synchronizes.
Parameters	hPCI (IN) - PCI handle for the PCIe-1000 device pCounter (OUT) - Pointer to 64-bit time counter value (nanoseconds)
Return value	TRUE - function succeeded FALSE - an error occurred
Remarks	

bcReadRegBinCounterNoSync	
Syntax	BOOL bcReadRegBinCounterNoSync(BC_PCI_HANDLE hPCI, PUINT64 pCounter);
Description	This function performs nonatomic read of 64-bit binary time counter from the MPC8308 registers through the PCIe bus. Because of its nonatomic nature, caller should insure the validity of each read by comparing subsequent read values. The epoch of the counter is the same as the epoch of the grandmaster to which this board currently synchronizes.
Parameters	hPCI (IN) - PCI handle for the PCIe-1000 device pCounter (OUT) - Pointer to 64-bit time counter value (nanoseconds)
Return value	TRUE - function succeeded FALSE - an error occurred
Remarks	

Read Current UTC Offset

bcReadCurrentUTCOffset	
Syntax	BOOL bcReadCurrentUTCOffset(BC_PCI_HANDLE hPCI, PSHORT pUTCOffset);
Description	This function retrieves the PTP grandmaster's current UTC offset in seconds. The PTP grandmaster runs the PTP time scale (TAI time). The MPC8308 time counter registers follows the PTP grandmaster time. So counter value is in TAI time scale. The current UTC offset is the offset in seconds between the UTC time scale and TAI time scale. The UTC time is the TAI time minus the current UTC offset.
Parameters	hPCI (IN) - PCI handle for the PCIe-1000 device pUTCOffset (OUT) - Pointer to 16-bit time counter value (seconds)
Return value	TRUE - function succeeded

bcReadCurrentUTCOffset	
	FALSE - an error occurred
Remarks	

Specifications

IEEE 1588 / PTP Compliance

- IEEE 1588 2008 (v2)
- Role: PTP Slave
- Default Profile with parameter extensions
- 1-Step or 2-step clock operation
- Multicast
- Hardware time stamping of PTP packets

Typical PTP Slave Performance

- Time Accuracy:
  - **1 PPS Sync/Delay_req rate** 7.9 nanoseconds @ 1 sigma (1σ) +/- 18 nanoseconds to the master via crossover cable
  - **32 PPS Sync/Delay_req rate** 10 nanoseconds @ 1 sigma (1σ) +/- 20 nanoseconds to the master via crossover cable
- Time Stamp Precision/Resolution: 10 nanoseconds

Configurable PTP Parameters

- Transport Protocol: UDP or 802.3
- Delay Mechanism: E2E or P2P
- Sync, E2E Delay and P2P Delay Intervals: 32 packets / 1 sec and 1 packet / 1 sec
- PacketTTL: 1 to 256
- Priority1 and 2: 0 to 255
- Domain Number: 0 to 255
- Mean Announce Message
  - Transmit Interval: 1, 2, 4, 8, 16, 32, 64 seconds
- Announce Receipt Timeout Multiplier: 2,3, 4, 5, 6, 7, 8, 9, 10

Network

- Ethernet:
  - RJ-45 100Base TX, 1000Base-T

Timing I/O

- 1 PPS-out: BNC Rising edge on-time, TTL into 50Ω
- Pulse width: 60 microseconds

Oscillator

- Type: High Performance OCXO
- Stability:
  - Short term:

    +/- 5.00ppb max tempco

    +/- 20.00ppb Max voltage

    +/- 25ppb max
  - Long term:

    +/-1.00ppb daily

    +/-100ppb yearly

    +/- 500 ppb over 10 Years

PCIe Specification

- Single lane PCI Express (PCIe) Interface, PCIe 1.0 compliant
- Power:
  - +3.3V @1.9 A
  - +12V@50 mA

Mechanical

- Size: Low Profile PCIe
- Dimensions:
  - 3.25" inches (8.26 cm) High
  - 6.75 inches" (17.15 cm) Deep

Environmental

- Temperature:
  - Operating: 0ºC to 50ºC
  - Storage: -30ºC to 85ºC
  - Humidity: 5% to 95% non-condensing
- Certifications:
  - FCC, Class B
  - CE(RoHS) 6 of 6

Software

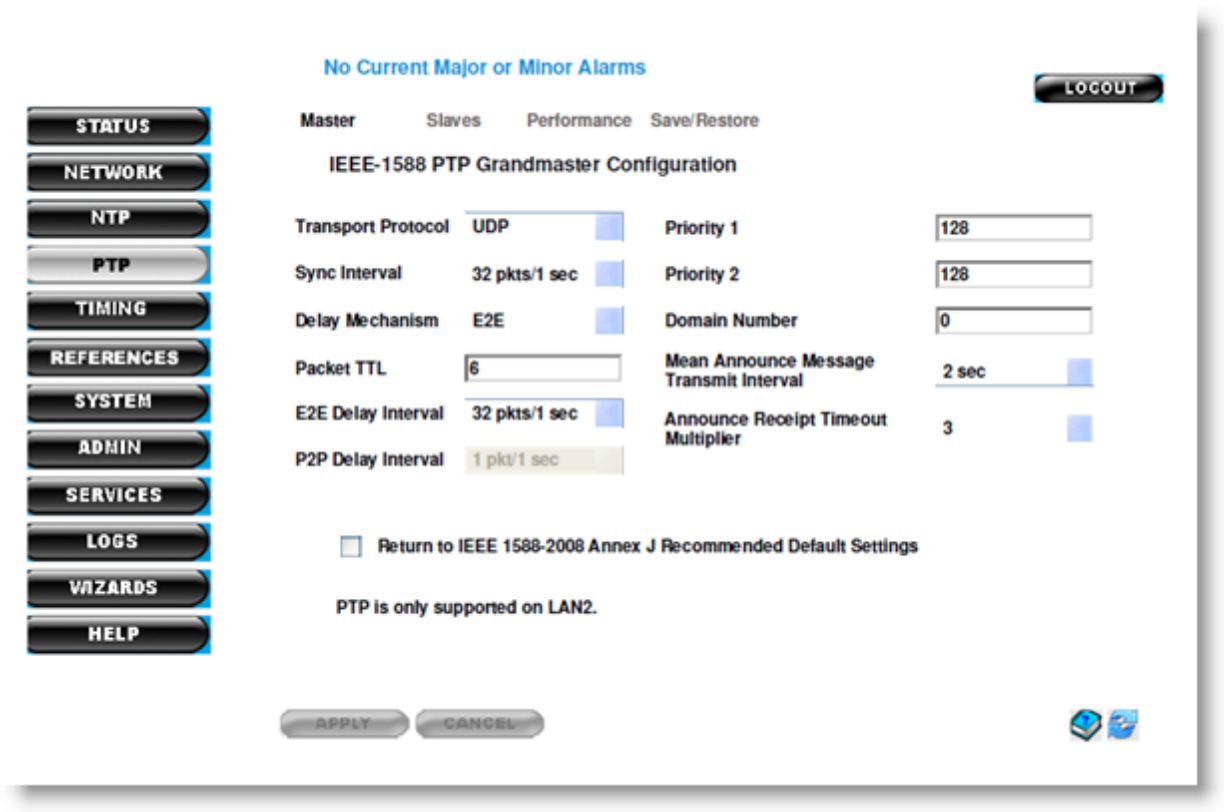
The PCIe-1000 CD includes software for 32/64 bit Linux with:

- Device driver
- API driver
- Sample code for reuse in applications using the provided APIs
- Command line interface with representative API calls to configure, monitor, and read time from card registers and host memory

Product Includes

- PCIe-1000 PTP clock card
- Standard height and low-profile cover plates
- One year warranty
- PCIe User's Guide CD
- Linux Driver software CD

Appendix A: IEEE-1588 PTP Grand-master Configuration



Setting Definitions

PTP - Master

Note: PTP is only supported on LAN2 of the 300 Series SyncServer

This page is used to set up parameters associated with the IEEE-1588 2008 PTP Grandmaster Configuration.

The following is a list of these configuration parameters:

Transport Protocol

Select from:

- UDP
- 802.3

Sync Interval

The Sync Interval is used to specify the mean time interval between successive Sync messages (the syncInterval) when transmitted as multicast messages.

The configurable range is 2-6 to 2+6 (2-6 to 2+6, is 1 second/64-packets to 64-packets/second).

Default is 20, which is 1 packet/second

Select from :

• 64pkt/1 sec

• 32 pkt/1 sec

• 16 pkt/1 sec

• 8 pkt/1 sec

• 4 pkt/1 sec

• 2 pkt/1 sec

• 1 pkt/1 sec

• 1 pkt/2 sec

• 1 pkt/4 sec

• 1 pkt/8 sec

• 1 pkt/16 sec

• 1 pkt/32 sec

• 1 pkt/64 sec

Note: The IEEE 1588-2008 requires the delay_request setting to be = or less than the sync interval. Bear this in mind as this Web UI does not enforce this.

Delay Mechanism

The choices for the Delay Mechanism are:

• E2E (End to End)

• P2P (Peer to Peer)

Packet TTL

(Time to live - TTL)

In this text box you can set the number of router hops up to 256 hops.

The TTL range is 1 to 256, if you enter 0 or >256 a message stating that the value of TTL is out of range will appear.

Note: 1588 multicast packets typically operate at a TTL of 1, changing this value may affect the quality of your network timing.

E2E Delay Interval

The end-to-end E2E Delay Interval controls the number of request packets from the slaves connected to this unit. See "Sync Interval"

When the E2E selection is made at the Delay Mechanism (see above), the following selections are available:

• 64pkt/1 sec

• 32 pkt/1 sec

• 16 pkt/1 sec

• 8 pkt/1 sec

• 4 pkt/1 sec

• 2 pkt/1 sec

• 1 pkt/1 sec

• 1 pkt/2 sec

• 1 pkt/4 sec

• 1 pkt/8 sec

• 1 pkt/16 sec

• 1 pkt/32 sec

• 1 pkt/64 sec

P2P Delay Interval

The peer-to-peer P2P Delay Interval controls the number of request packets from the slaves connected to this unit. See "Sync Interval"

When the P2P selection is made at the Delay Mechanism (see above), the following selections are available:

• 64pkt/1 sec

• 32 pkt/1 sec

• 16 pkt/1 sec

• 8 pkt/1 sec

• 4 pkt/1 sec

• 2 pkt/1 sec

• 1 pkt/1 sec

• 1 pkt/2 sec

• 1 pkt/4 sec

• 1 pkt/8 sec

• 1 pkt/16 sec

• 1 pkt/32 sec

• 1 pkt/64 sec

Priority 1

The priority field affects the result of the Best Master Clock Algorithm. The lower number in this field will win the BMC calculation. The initialization value of priority1 is specified in a PTP profile. Choices are:

• 0 – 255

• Default is 128

Priority 2

The priority field affects the result of the Best Master Clock Algorithm. The lower number in this field will win the BMC calculation. The initialization value of priority2 is specified in a PTP profile. Choices are:

• 0 – 255

• Default is 128

Domain Number

A domain consists of one or more PTP devices communicating with each other as defined by the protocol. A domain defines the scope of PTP message communication, state, operations, data sets, and timescale. PTP devices may participate in multiple domains; however, unless otherwise specified in the standard, the operation of the protocol and the timescale in different domains is independent.

The configurable range is 0 – 255

• Default is 0

Mean Announce Message Transmit Interval

This is the Announce Interval specified in IEEE 1588-2008 and is specified as the mean time interval between successive Announce messages.

Selections available:

• 1 sec

• 2 sec

• 4 sec

• 8 sec

• 16 sec

• 32 sec

• 64 sec

• Default is 2 seconds

Announce Receipt Timeout Multiplier

The value of Announce ReceiptTimeout is an integral multiple of the announceInterval (see section 7.7.3.1 of IEEE 1588-2008).

• The configurable range is 2 to 10 (22 to 210).

• Default is 3

Return to IEEE 1588-2008 Annex J Recommended Default Settings check box

If you are not sure what selections to make, click on this button to get the standard settings. Any slave should support theses settings as these settings meet the specification.

IEEE 1588-2008 Annex J Recommended Default Settings

Transport Protocol UDP

Sync Interval 1 pkt/1 sec

Delay Mechanism E2E

Packet TTL 1

E2E Delay Interval 1 pkt/1 sec

P2P Delay Interval 1 pkt/1 sec

Priority 1 128

Priority 2 128

Domain Number 0

Mean Announce Message Transmit Interval 2 sec

Announce Receipt Timeout Multiplier 3

Important note: When the Grandmaster is reconfigured after the initial set up, all PCIe-1000 bus cards need to have PTP restarted so as to synchronize with the Grandmaster. See "Starting and Stopping PTP" on page 16

Appendix B: Best Master Algorithm

The Best Master Algorithm is how the slave determines which master to use.

Following is the list, in priority order, of the PTP parameters used in the Best Master Algorithm.

The data set comparison algorithm is based on pair wise comparisons of attributes with the following precedence:

1. Priority1: A user configurable designation that a clock belongs to an ordered set of clocks from which a master is selected.
2. ClockClass: An attribute defining a clock's TAI traceability.
3. ClockAccuracy: An attribute defining the accuracy of a clock. IEEE P1588 D2.2 35
4. OffsetScaledLogVariance: An attribute defining the stability of a clock.
5. Priority2: A user configurable designation that provides finer grained ordering among otherwise equivalent clocks.
6. ClockIdentity: A tie-breaker based on unique identifiers. (MAC ID).

In addition to this precedence order, the "distance" measured by the number of boundary clocks between the local clock and the foreign master is used when two Announce messages reflect the same foreign master. The distance is indicated in the stepsRemoved field of Announce messages. This condition can occur in PTP systems with cyclic paths not removed by a protocol outside of PTP. The data set comparison algorithm unambiguously selects one of the two clocks as "better" or as "topologically better. "

Glossary

The following is a glossary of key terms used in the discussion of timing operations: An expanded glossary of terms is available on-line at:

http://www.symmetricom.com/resource/glossary/

Atomic Time: is a term usually used to refer to an official time scale that is based on atomic clocks. Examples are UTC, UTC(NIST), UTC(USNO), TAI, GPST, etc. (also see

BCD: Binary Coded Decimal. Also called packed decimal, this is the representation of each digit of a decimal number by four-bit binary numbers. For example, the number 42 would be encoded as 0100 0010 .

Coordinated Universal Time (UTC): See UTC.

COTS: Commercial Off-The-Shelf products or services that are generally available and not built to customized specifications.

DCLS: Direct Current Level Shift, or digital IRIG.

Discipline: The word discipline, as used in this manual, means to adjust the frequency of the 10 MHz oscillator to track the incoming reference signal.

DPRAM: Dual Port RAM.

Epoch: A reference time or event. Epoch often refers to a one pulse per second event.

Event: An event is defined here as a transition of a digital signal (rising or falling), which can be used to time stamp the event.

Flywheel: To maintain time or frequency accuracy when the reference source has been lost or removed.

GPS: Global Positioning System. Originally designated NAVSTAR (Navigation System with Timing And Ranging), GPS was developed by the US Department of Defense to provide all-weather round-the-clock navigation capabilities for military ground, sea, and air forces.

HW: Hardware.

IRIG: Serial time format standard maintained by the Inter Range Instrumentation Group.

ISA: Industry Standard Architecture; desktop PC adapter board specification.

Jamsync: Is the process of abruptly synchronizing with a time reference, as opposed to gradually adjusting to match up with the time reference.

Major Time: Units of time larger than or equal to seconds.

Malloced: The function malloc is used to allocate a certain amount of memory during the execution of a program.

MHz: A MegaHertz is one million (1,000,000) cycles per second.

Minor Time: Sub-second time to whatever resolution is supported.

MTBF: Mean Time Between Failure, a measure of reliability. The longer the time span between failures, the more reliable the device.

MTTR: Mean Time To Repair.

Mutex:

Short for mutual exclusion object. In computer programming, a mutex is a program object that allows multiple program threads to share the same resource, such as file access, but not simultaneously. When a program is started, a mutex is created with a unique name.

NASA 36: National Aeronautics & Space Administration 1-second BCD 36-bit Time Code.

NIST: National Institute of Standards and Technology, the National Measurement Institute in the United States.

Non-Atomic Time: There are many non-Atomic times such as UT, which is based on the rotation of the earth, and Ephemeris time, which is based on position of the earth in its orbit around the sun. (also see Atomic Time).

OCXO: Oven-Controlled Crystal Oscillator

OS: Operating System.

Packet: A group of bytes conforming to a defined structure. Packets are usually used in bit serial or byte serial data transmissions to allow framing of the transmitted data.

PCI: Peripheral Component Interconnect, a local bus that supports high-speed connection with peripherals. It plugs into a PCI slot on the motherboard.

PCIe: PCI Express, (Peripheral Component Interconnect Express), officially abbreviated as PCIe, is a computer expansion card standard designed to replace the older PCI, PCI-X, and AGP standards. Introduced by Intel in 2004, PCIe is the latest standard for expansion cards that is available on mainstream personal computers. PCI Express is used in consumer, server, and industrial applications, both as a motherboard-level interconnect (to link motherboard-mounted peripherals) and as an expansion card interface for add-in boards. A key difference between PCIe and earlier PC buses is a topology based on point-to-point serial links, rather than a shared parallel bus architecture.

PCISIG: PCI Special Interest Group.

PCM: Pulse Code Modulation.

Periodic: A programmable frequency that is obtained by dividing the TFP reference frequency. Periodics are sometimes referred to as “heartbeats.” PICMG: PCI Industrial Computer Manufacturers Group.

PLL: Phase-Locked Loop.

PPM: parts per million.

PPS: pulse per second.

RAM: Random Access Memory.

Resolution: Resolution of a time code refers to the smallest increment of time, whether it is days, hours, seconds, or other increments.

Strobe: The strobe is a programmable “alarm.” It compares the reference time with a user-programmed time, and outputs a signal when the two values are the same. The signal is indicated by a transition from low to high voltage. The duration of the signal is equal to 1 uSec. The Strobe function is also referred to as Time Compare.

SW: Software.

Sync/NoSync: “NoSync”, means that the operation is non-atomic. A memory read mutex is not used while reading the time information. If the user call the “NoSync” API functions consecutively, there is no guaranteed that the last read will have updated information. It could be reading the same time as the previous read.

To safeguard against this, the user will have to add detection themselves to determine if the time read is updated.

As for the API’s without the “NoSync”, a memory read mutex is used (atomic).This ensures that once the function returns the read time, it is updated and valid.

TCXO: Temperature Compensated Crystal Oscillator

TFP: Time and Frequency Processor is the name given to the bc63x family of products.

USNO: U.S. Naval Observatory, in Washington, D.C., where the atomic clock that serves as the official source of time for the United States is maintained.

UTC: The international time standard is called Universal Coordinated Time or, more commonly, UTC, for “Universal Time, Coordinated”. This ITU standard has been in effect since 1972. UTC is maintained by the Bureau International de l'Heure (BIH), which forms the basis of a coordinated dissemination of standard frequencies and time signals.

VCXO: Voltage-Controlled Crystal Oscillator.

Index

C

D

E

F

G

U

V

2300 Orchard Parkway
San Jose, California 95131-1017
tel: 408.433.0910 fax: 408.428.6960
www.symmetricom.com