USER MANUAL

ProDAQ LXI Instruments

ProDAQ 6100

LXI Function Card Carrier

PUBLICATION NUMBER: 6100-XX-UM-1030

bustec

Bustec Production, Ltd.
Bustec House, Shannon Business Park, Shannon, Co. Clare, Ireland
Tel: +353 (0) 61 707100, FAX: +353 (0) 61 707106

Table of Contents

Table of Figures

Reference Documents

Title	Number
Bustec VISA User Manual	8200-XX-UM
ProDAQ Function Card User Manuals	
ProDAQ 6210 Rack-mount Tray Assembly Instructions	6210-XX-AI

Glossary

DAC	-	Digital-to-Analog Converter
FIFO	-	First-in First-out Memory
Sample	-	16-bit number representing an analog value
Motherboard	-	ProDAQ Motherboard or Carrier featuring function card slots to install ProDAQ function cards in

Safety

This equipment contains voltage hazardous to human life and safety and is able to inflict personal injury. Disconnect the device from the AC line (mains) before opening the covers as described in chapter 3.4.

To operate this device, use a three-conductor power cord and an power outlet providing protective earth. Do not use a two-conductor extension cord or a three-prong/two-prong adapter.

If you replace the power cord provided, make sure that the replacement is rated for the power consumption stated in the specifications.

Do not position the device so that it is difficult to operate the disconnecting device.

If the equipment is used in a manner not specified by the manufacturer, its safety may be impaired.

Waste Electrical and Electronic Equipment (WEEE)

This product complies with the WEEE Directive 2002/96/EC marking requirement. The affixed product label indicates that you must not discard this electrical product in domestic household waste.

Product Category: Monitoring and Control Instrumentation

To return unwanted products, contact Bustec Ltd.

1. Introduction

1.1. Overview

The ProDAQ 6100 LXI Function Card Carrier provides access to up to four ProDAQ Function Cards through a standard Gigabit LAN interface. Its small form factor and connectivity combined with the flexibility of the ProDAQ function cards lets you create multifunction instruments where your application needs them. Simply connect the ProDAQ 6100 to your network and use the standard software drivers or the embedded Web server to integrate it into your application.

The ProDAQ 6100 is a fully compliant LXI Class B device (Class A with the optional LXI Trigger Interface), providing a standardized Gigabit LAN interface, a synchronization API and support for the IEEE 1588 protocol. The IEEE 1588 interface allows devices to execute triggered functions equivalent to those available over GPIB and with similar or better timing accuracy.

The optional trigger interface allows connecting LXI devices by a physical daisy chain or star configuration. It is based on an 8-channel multipoint LVDS signaling system that allows devices to be sources and/or receivers of trigger and clock signals. Using a Wired-OR configuration allows multiple devices to respond to a trigger signal or share a common clock.

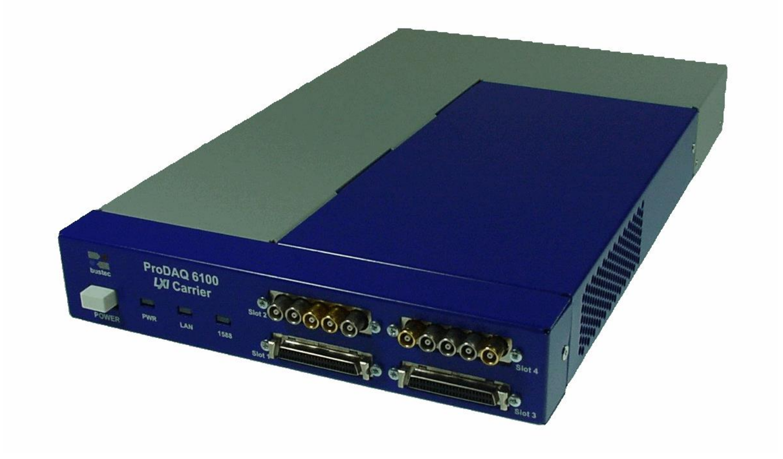


Figure 1 - ProDAQ 6100 LXI Function Card Carrier

1.2. Features

1.2.1. IEEE 1588 Enabled System-on-a-chip CPU

The ProDAQ 6100 utilizes the Freescale QUICC Engine™ technology and Enhanced Triple Speed Ethernet Controller (eTSEC), which are the first Freescale communications interfaces to optimize IEEE 1588 PTP in hardware. These new interfaces time-stamp Ethernet packets at the physical/datalink layer the instant they enter or leave the device. This results in the highest possible timing accuracy for PowerQUICC® devices that implement the increasingly popular IEEE 1588 protocol.

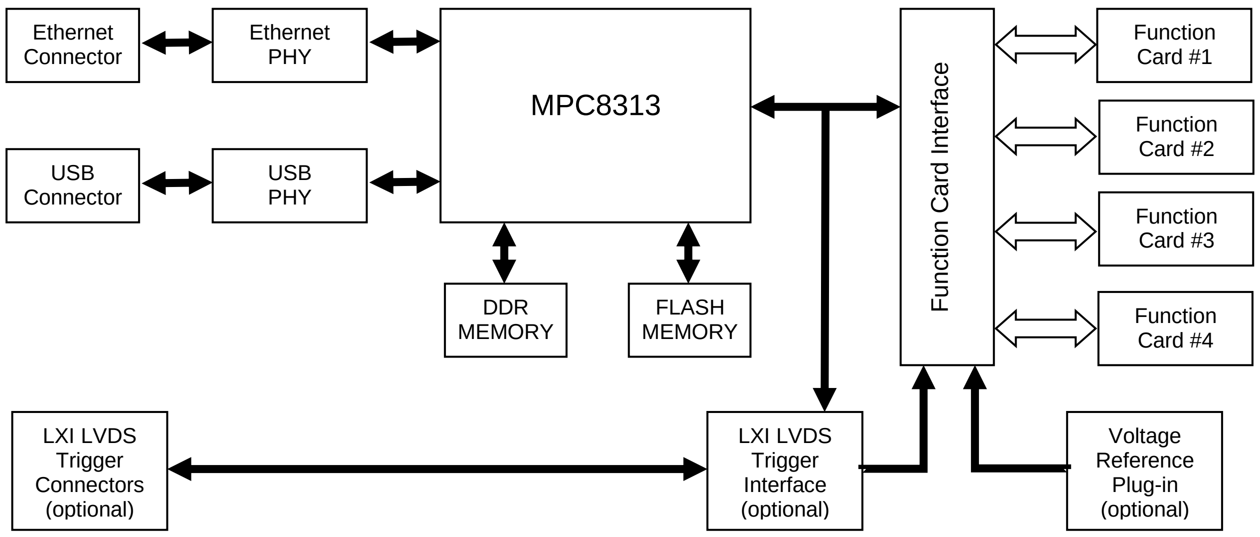


Figure 2 - ProDAQ 6100 Block Diagram

The Freescale MPC8313 PowerQUICC II Pro Communications Processor combines a powerful e300 PowerPC core with a complete set of peripherals and interfaces, such as a DDR2 memory interface, a PCI local bus, a high-speed USB interface and a 10/100/1000 Ethernet controller.

1.2.2. Function Card Interface

The ProDAQ 6100 can host up to four of the standard ProDAQ function cards. Each function card is connected to the carrier board via the function card interface, which provides a separate interface to each function card consisting of

- A 16-bit multiplexed address/data bus
- Up to two trigger in lines
- Up to two trigger out lines
- A voltage reference bus
- Supply voltages

The address/data bus allows the ProDAQ 6100 access to the internal registers and memory areas of the function card. Each function card occupies an address space of 64 kWords. Due to the separate interfaces, the ProDAQ 6100 can access simultaneously several function cards (Word, Double-word or Quad-word access).

The trigger in/out lines can be routed either to other function cards, the optional LXI trigger bus or the CPU for processing. The trigger routing facility in the function card interface utilizes a separate trigger node per trigger sink, where each node can be enabled to receive triggers from each trigger source available in the system. In this way one-to-one, n-to-one, one-to-n as well as n-to-n connections are possible. Additional nodes allow to route triggers to the processor for interrupt generation or the internal distribution of clock signals.

1.2.3. IEEE1588 Precision Time Protocol

The ProDAQ 6100 implements the precision time protocol according to IEEE 1588-2008 (version 2) for clock synchronization. It can operate as both master and slave and allows the internally generated clock of 100 MHz to be conditioned down to an accuracy of 30 ns. The synchronized clock can be routed to the function cards to be used as clock for the data acquisition and generation.

1.2.4. Voltage Reference Plug-in

The ProDAQ 6100 allows installing a ProDAQ 3202 High-precision Voltage Reference Plug-in module. The reference voltage generated by the plug-in module is distributed internally via the voltage reference bus to each function card. ProDAQ function cards utilize this reference voltage in their input stages to allow an end-to-end calibration without disconnection from the device under test. The voltage reference bus is accessible as well via a dual 2mm test jack connector on the rear. This allows recalibrating the ProDAQ 3202 Voltage Reference Plug-in without removing it from the ProDAQ 6100, or, if no plug-in is installed, to feed an external reference voltage into the system.

1.2.5. LXI Trigger Bus

The ProDAQ 6100 features an optional LXI (LAN eXtensions for Instrumentation) trigger bus on the rear panel of the instrument. The LXI Trigger Bus consists of eight shielded twisted-pair wires that can be used to distribute M-LVDS signals between groups of devices connected in either a daisy-chain, star, or hybrid-star configuration. The bus consists of two identical ports connected in parallel. The bus can be used to distribute high-speed event signals, clocks or similar between devices.

1.2.6. High-voltage Option

The ProDAQ 6100 can be equipped with two different power supply options. The standard version supplies function card with digital and analog supply voltages of -2V, -5.2V, +5V, ±12V and ±15V. The optional high-voltage version adds another supply voltage of ±24V, which is necessary for some function cards to allow generating wide-range output signals or driving ICP sensors.

2. Specifications

2.1. Available Versions

Versions	6100-AA LXI Carrier 6100-AB LXI Carrier with HV Power Supply Option 6100-BA LXI Carrier with LXI Trigger Interface 6100-BB LXI Carrier with LXI Trigger Interface and HV Power Supply Option

2.2. Function Card Interface

Number of Slots	4
Supported card configurations	• four single-wide function cards **or** • two double-wide function cards **or** • one double-wide and two single-wide function cards
Interface Width	16/32 bit
Interface Speed	Up to 120 MB/s

2.3. Network Interface

Type	10/100/1000 Base-T (IEEE 802.3 compliant)
Auto-MDIX	yes
Connector Type	RJ45
Protocols	TCP/IP, HTTP, VXI-11, IEEE1588, mDNS

2.4. LXI Device Capabilities

LXI Class Compliance	6100-AA: LXI Class B 6100-AB: LXI Class B 6100-BA: LXI Class A 6100-BB: LXI Class A
LXI Version	1.3
Programmatic Interface	Embedded Web Interface VXI-11 Device, Instrument Discovery IVI/VXIplug&play compatible driver

2.5. IEEE 1588

Version	IEEE 1588-2008
Profile	Compatible with LXI IEEE 1588 Profile 1.0
Clock Class	Ordinary clock
Time Source	Internal oscillator
Timestamp Accuracy	10 ns

2.6. LXI Trigger Interface (Optional)

No. Trigger Line	8
Type	M-LVDS, LXI Class A Trigger Bus compatible
Input Trigger Pulse Width	20 ns minimum
Connector	25-pin Molex Micro-D

2.7. Environmental Specifications

Temperature	0 °C to +50 °C (operational) -40 °C to +70 °C (storage only)
Humidity	5% - 95% (non-condensing)

2.8. Power Supply

Input Voltage	85-265 VAC, 47-63 Hz
Current Consumption	1.6 A max.
Fuses	2A Slow-Blow

3. Getting Started

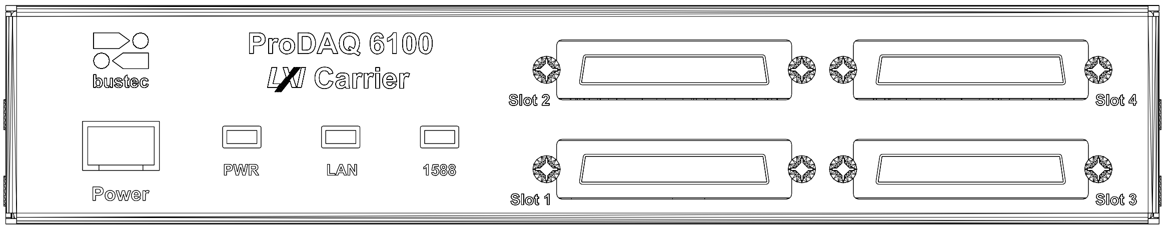
3.1. Unpacking and Inspection

The ProDAQ 6100 is shipped in an antistatic package to prevent any damage from electrostatic discharge (ESD). Proper ESD handling procedures must always be used when packing, unpacking or installing any ProDAQ device or ProDAQ function card:

- Ground yourself via a grounding strap or similar, e.g. by holding to a grounded object.
- Discharge the package by touching it to a grounded object, before removing the module from the package.
- Remove the ProDAQ module from its carton, preserving the factory packaging as much as possible.
- Inspect the ProDAQ module for any defect or damage. Immediately notify the carrier if any damage is apparent.

3.2. Front Panel Switches and Indicators

The ProDAQ 6100 features the following switches and indicators on its front panel:
The "Power"- switch can be used to toggle the ProDAQ 6100 between standby and operation



mode. The "PWR" indicator shows the mode by toggling between orange (standby) and green (operation). To switch off the ProDAQ 6100 completely, use the power switch integrated into the IEC inlet on the rear panel.

The "LAN" indicator is a bi-color LED. It is used to identify the ProDAQ 6100 via its WEB interface or its Soft Front Panel and to indicate a LAN failure. When used for LAN identification, it flashes green, while a LAN failure is indicated by a constant red light. During normal operation, it shows a constant green light.

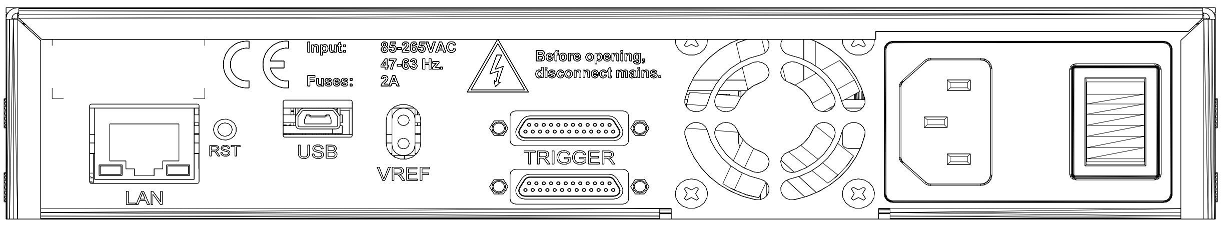
The "1588" indicator shows the status of the clock used for the Precision Time Protocol. If it is off, the ProDAQ 6100 is neither master nor slave and uses the internal clock. If is shows a constant green light, the ProDAQ 6100 is in slave mode and its clock is synchronized with a master clock it is connected to. If it is flashing green once every second, it is master to other devices, but not the grandmaster and when it is flashing once every two seconds, it is master and also grandmaster. A failure is indicated by a constant red light.

The ProDAQ 6100 can host up to four singlewide or two doublewide ProDAQ function cards. Single wide function cards can be installed in either of the four function card positions, while double wide function cards need to be installed either in function card position one and three or two and four.

3.3. Rear Panel Switches and Connectors

3.3.1. ProDAQ 6100 V1

The rear panel of the ProDAQ 6100 V1 features the following switches and connectors:

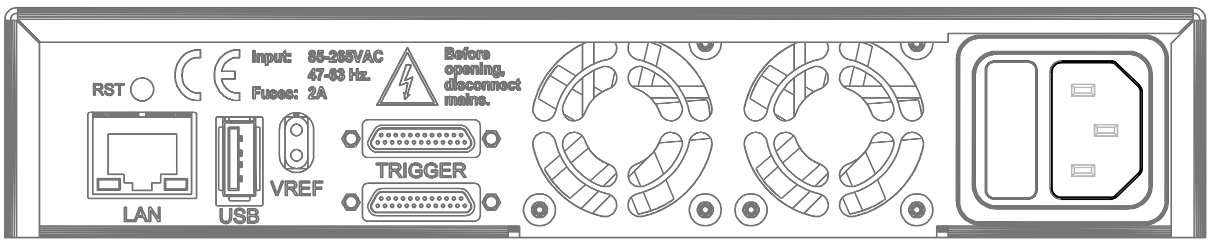


LAN	RJ45 10/100/1000 BASE-TX port with Auto-MDIX
USB	USB 2.0 OTG port
VREF	Dual 2mm test socket. When a ProDAQ 3202 Voltage Reference Plug-in module is installed in the ProDAQ 6100, it can be used to monitor the voltage reference output. If no plug-in is installed, it can be used to feed in a calibration voltage to be used on the ProDAQ function cards installed in the ProDAQ 6100.
TRIGGER	Dual LXI LVDS trigger port (optional)

The IEC Inlet can be used to connect the ProDAQ 6100 to the mains. It also features the master on/off switch and the main fuses.

3.3.2. ProDAQ 6100 V2

The rear panel of the ProDAQ 6100 V1 features the following switches and connectors:



LAN	RJ45 10/100/1000 BASE-TX port with Auto-MDIX
USB	USB 2.0 host port
VREF	Dual 2mm test socket. When a ProDAQ 3202 Voltage Reference Plug-in module is installed in the ProDAQ 6100, it can be used to monitor the voltage reference output. If no plug-in is installed, it can be used to feed in a calibration voltage to be used on the ProDAQ function cards installed in the ProDAQ 6100.
TRIGGER	Dual LXI LVDS trigger port (optional)

The IEC Inlet can be used to connect the ProDAQ 6100 to the mains. It also features the main fuses.

3.4. Installing ProDAQ Function Cards

If the ProDAQ 6100 is bought together with the ProDAQ function cards, the function cards will be pre-installed in the company to your specification. If you want to install additional cards or exchange installed cards, use the following disassembling/assembling procedure.

WARNING

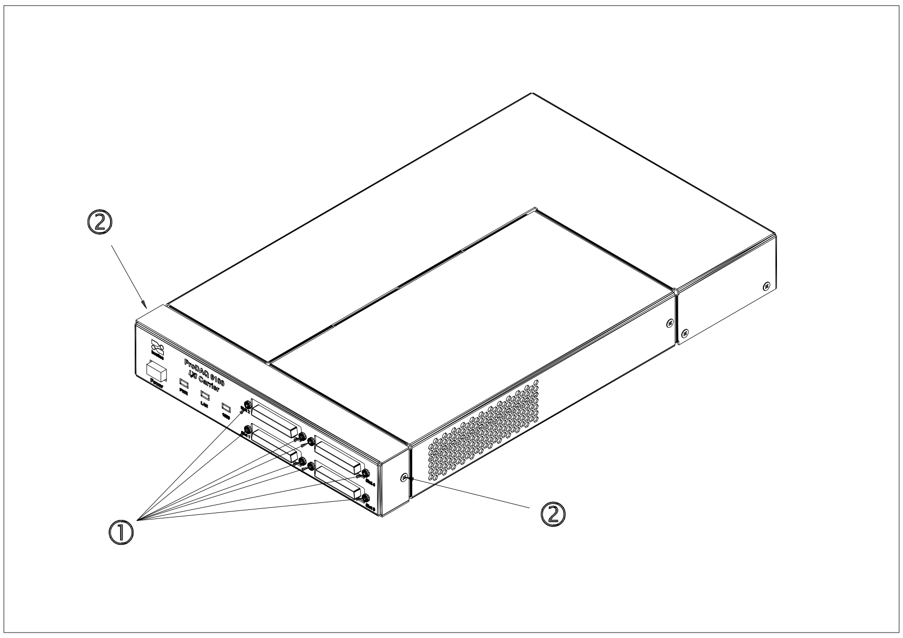
Disconnect the ProDAQ 6100 from the mains before opening the enclosure!

WARNING

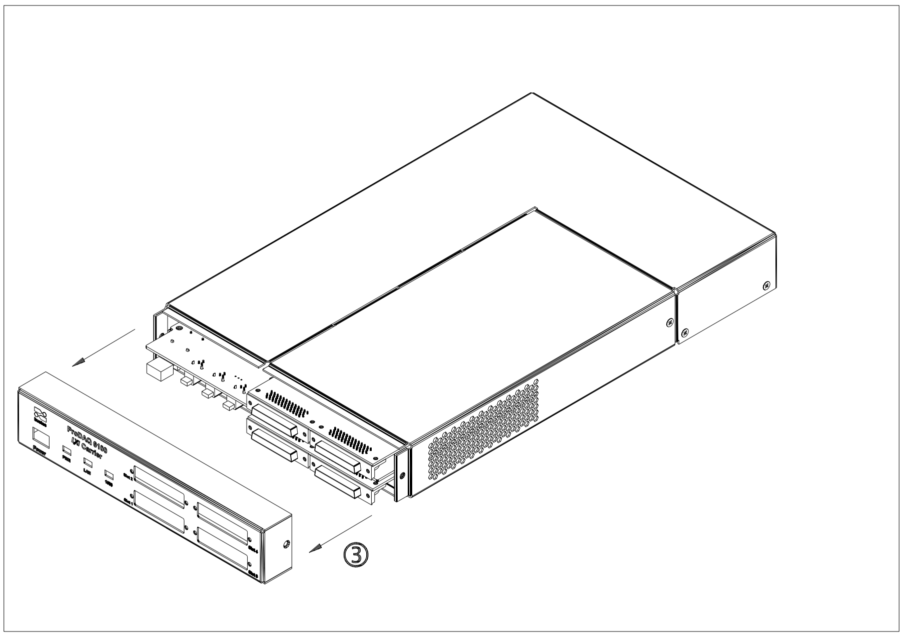
Proper ESD handling procedures must always be used when packing, unpacking or installing any ProDAQ device or ProDAQ function card. Ground yourself via a grounding strap or similar, e.g. by holding to a grounded object and discharge the package by touching it to a grounded object, before removing the module from the package.

3.4.1. Opening the ProDAQ 6100 Enclosure

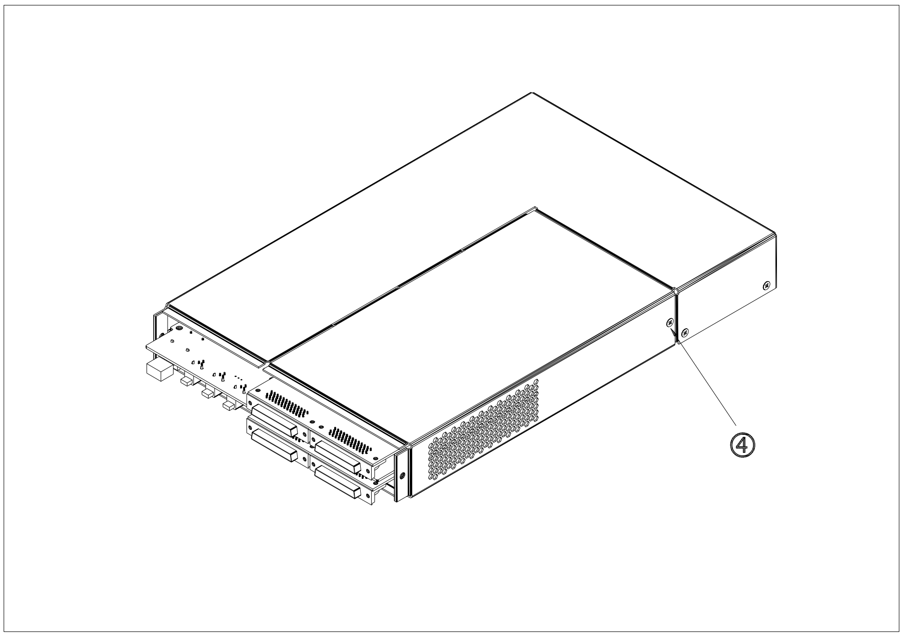
Remove the up to eight M2.5x6mm Panhead screws (①) attaching the front bezel to the function cards (If there is no function card installed in a slot and a blanking panel is used to cover the front bezel opening, do not remove it screws before detaching the front bezel). Then remove the two M3x6mm Countersunk screws (②) attaching the front bezel to the enclosure.



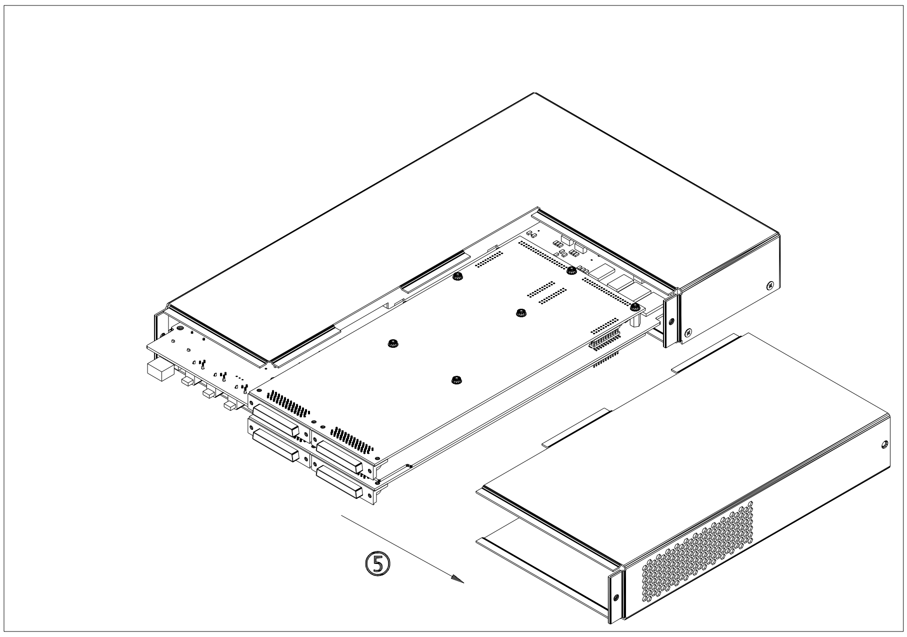
Slide the front bezel off (③) as shown below:



Remove the M3x6mm Countersunk screw (④) attaching the function card cover to the enclosure:



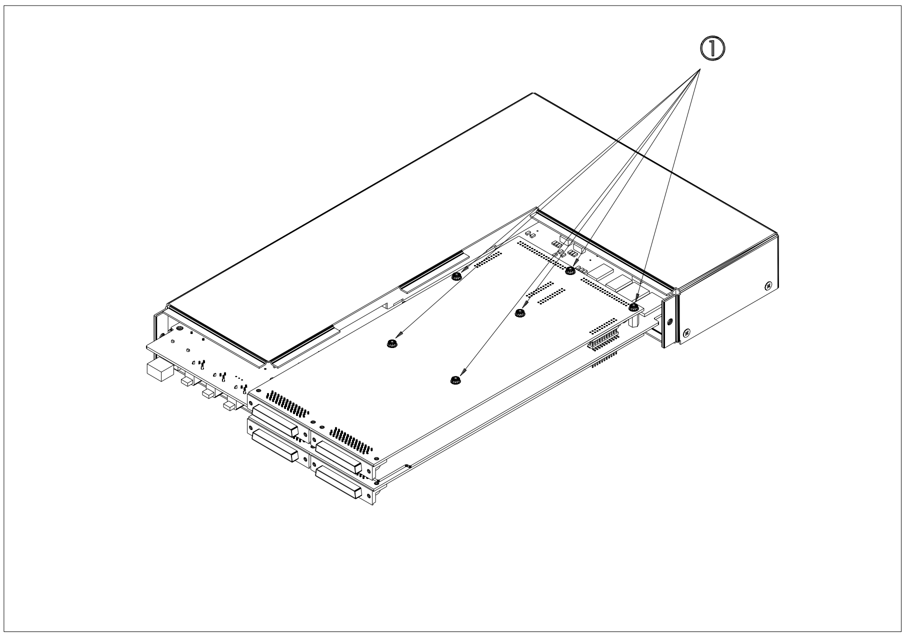
Slide the function card cover off (⑤) as shown below:



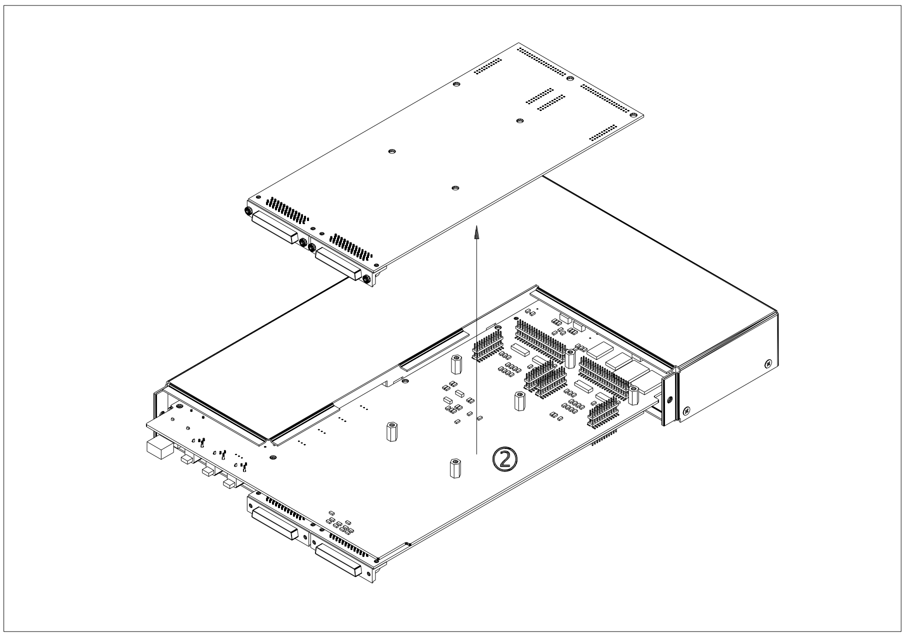
3.4.2. Removing a ProDAQ Function Card

The ProDAQ Function Cards are mounted inside the ProDAQ 6100 directly on the main PCB. The function cards positions two and four are located on top of the PCB and the positions one and three below. The function cards are mounted face down, e.g. the front-panel connectors as well as the motherboard connectors are underneath the PCB when mounted.

If you need to remove an installed function card before installing a new one, remove the three M3x6mm screws (①) mounting them to the base board (six M3x6mm screws for a double wide function card.



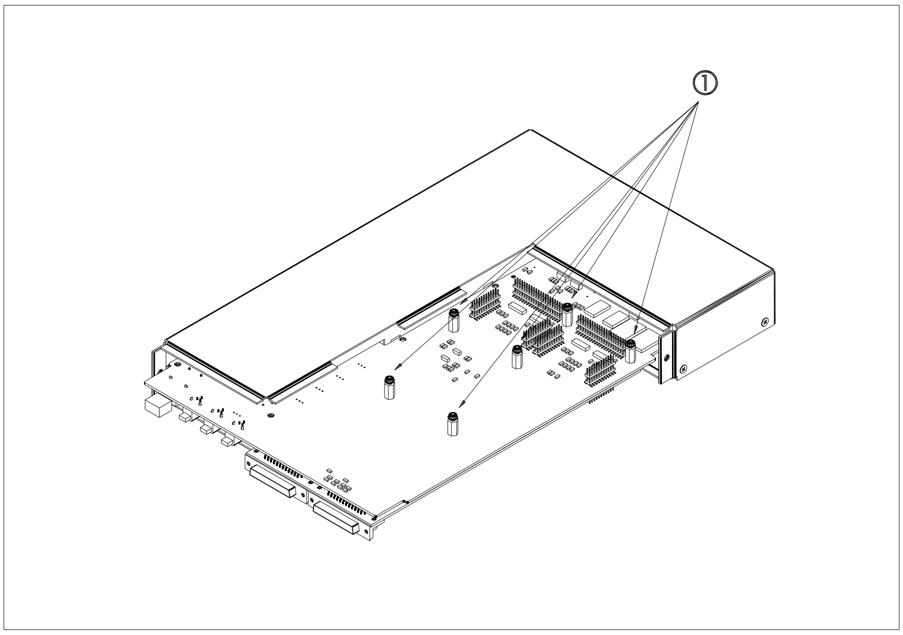
Remove the function card by pulling it (②) straight and evenly upward (or downward for a function card mounted on the bottom of the main PCB). Do not tilt the function card when doing so as it



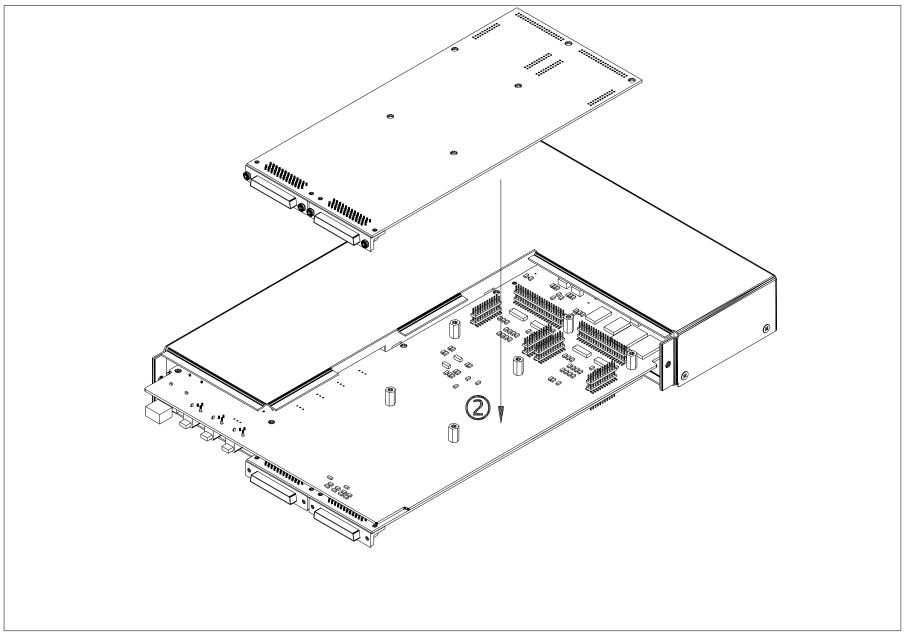
might damage the connectors connecting it to the ProDAQ 6100 PCB.

3.4.3. Installing a ProDAQ Function Card

To install a ProDAQ Function Card into the ProDAQ 6100 LXI Function Card Carrier, you must first remove the front bezel and the function card cover as shown previously (see paragraph 3.4.1 Opening the ProDAQ 6100 Enclosure). Make sure that the M3x6mm screws and washers are removed from the PCB standoffs (①):

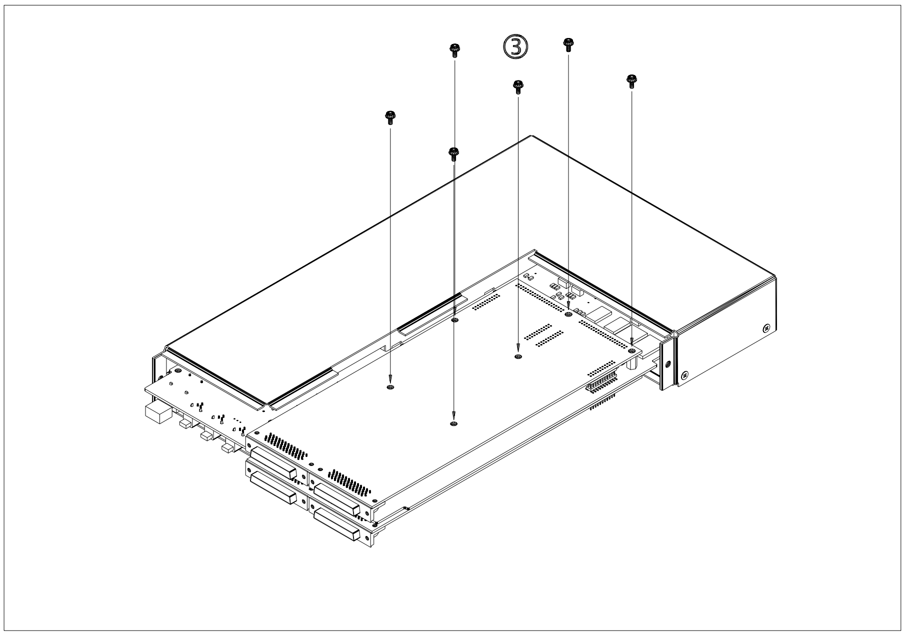


Position the function card over the function card slot you want to install it to (②), carefully aligning the connectors connecting it to the ProDAQ 6100 PCB and push it down until it seats fully onto the



standoffs of the ProDAQ 6100 PCB:

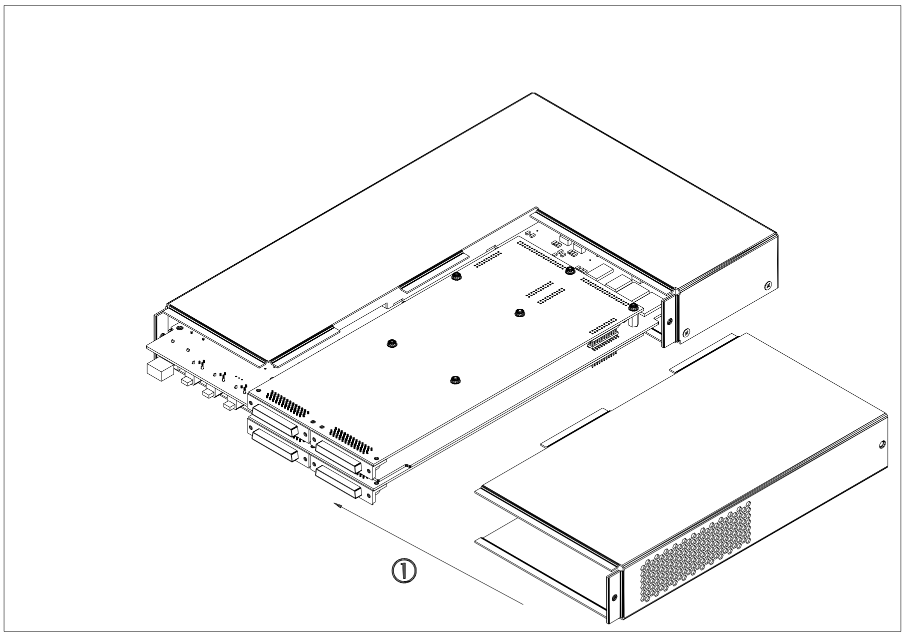
Use three M3x6mm panhead screws and washers (③) to attach the function card to the ProDAQ



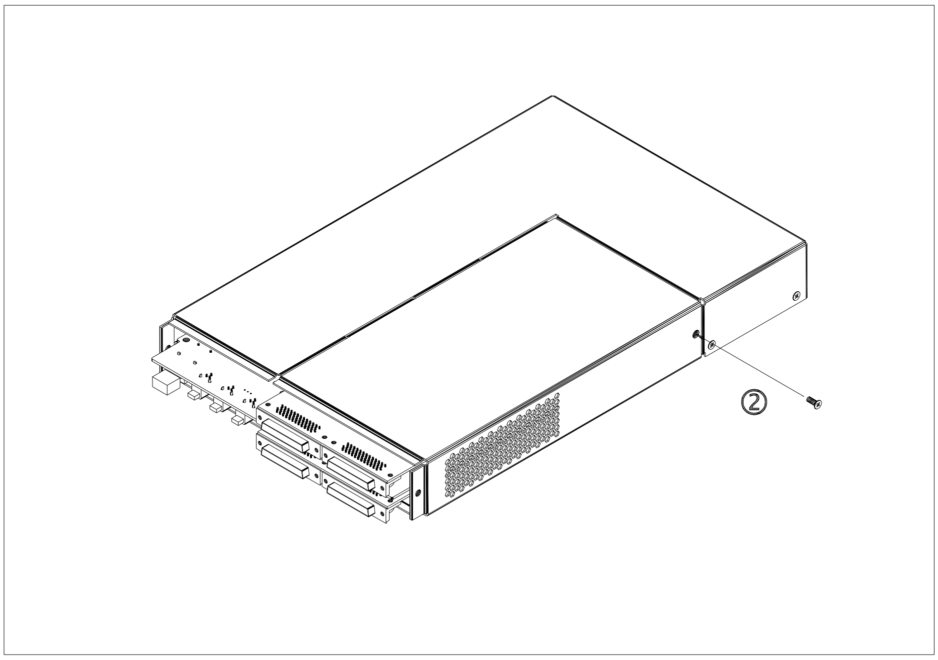
6100 PCB (six screws and washers for a double wide function card):

3.4.4. Closing the ProDAQ 6100 Enclosure

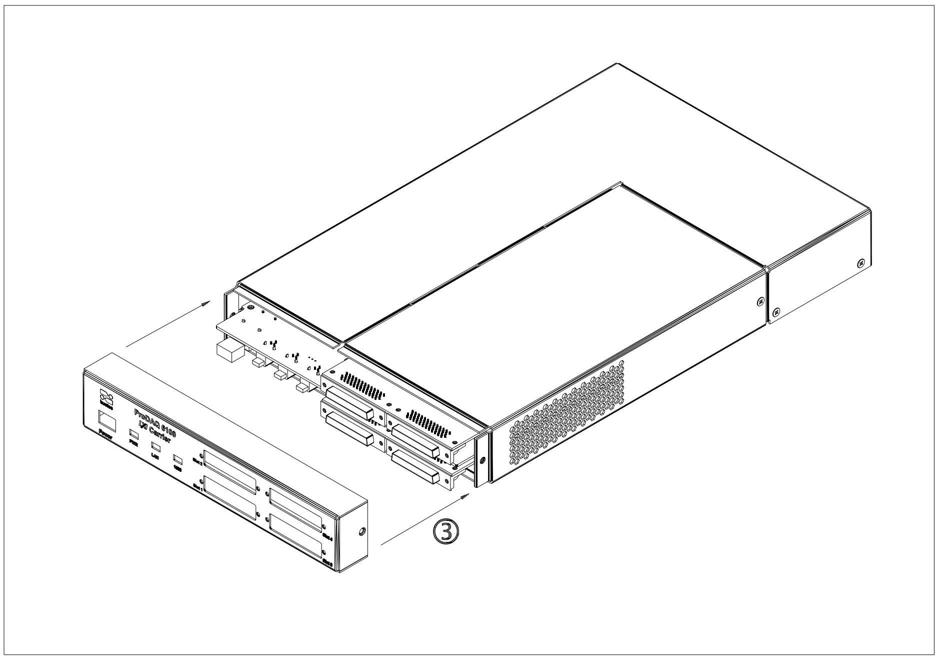
To close the enclosure after installing or removing a ProDAQ function card, first slide back on the function card cover (①):



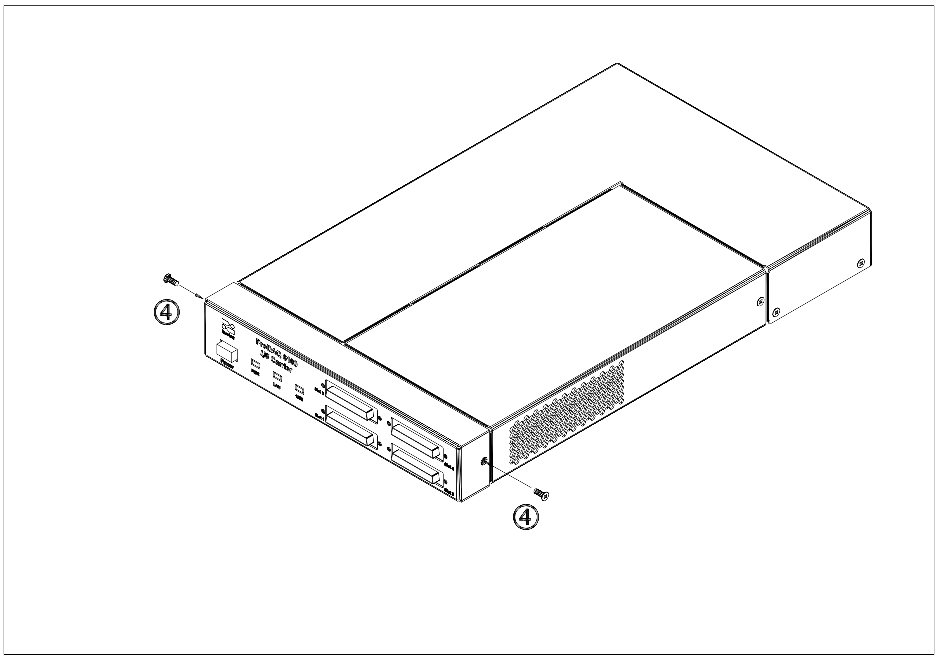
and attach it with a M3x6mm Torx screw to the enclosure:

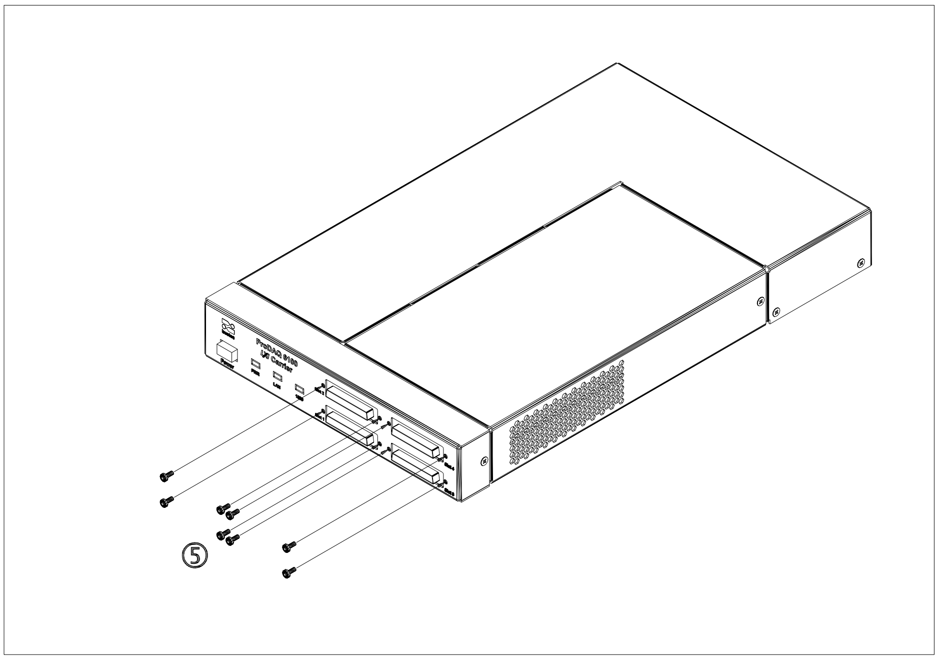


Make sure that the cutouts for the function card connectors in the front bezel are properly opened or covered by filler panels to match the installed function cards. Slide the front bezel back on (③)



and attach it to the enclosure by two M3x6 countersunk screws





Attach the function cards to the front bezel using up to eight M2.5x6 panhead screws.

Warning

Using other screw types or lengths may permanently damage the instrument and/or cause electrical shorts!

3.5. Preparing the ProDAQ 6100 for first use

The ProDAQ 6100 can be used on a tabletop or mounted on a ProDAQ 6210 tray in a standard 19" rack.

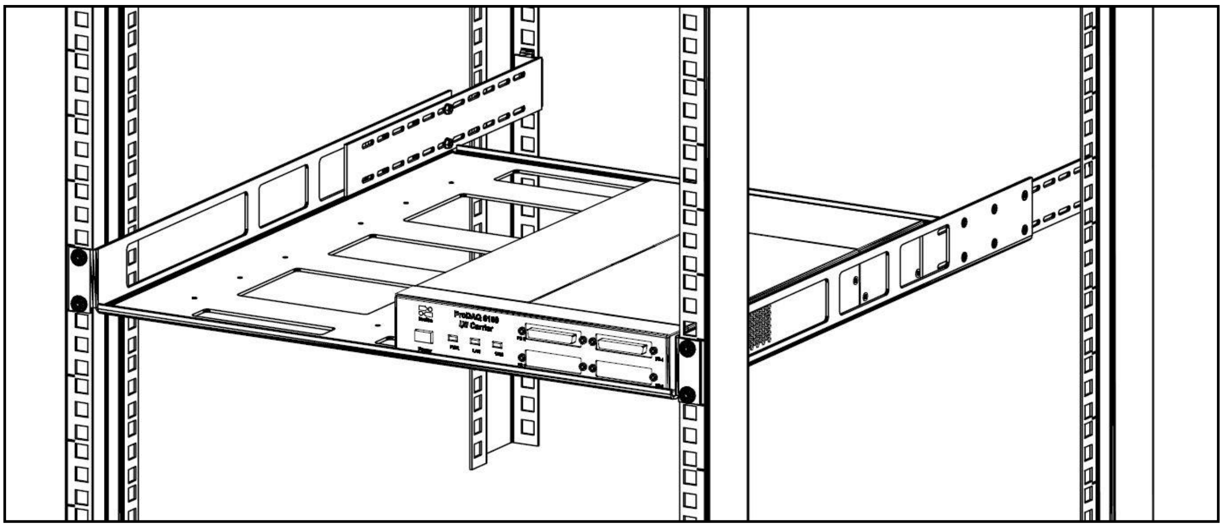


Figure 3 - ProDAQ 6210 Rack-mount Tray

Please refer to the ProDAQ 6210 assembly instructions which can be downloaded from the Bustec web site (see https://www.bustec.com/support/manuals/))for details on how to install the tray and the ProDAQ 6100 into it..

Once the unit is in its place, connect the ProDAQ 6100 to your network via a 10/100/1000-Mbit switch or directly to your host computer using a CAT5e Ethernet cable, connect the IEC inlet on the rear to a mains outlet and turn the device on.

During the configuration of the network port the LAN LED on the front blinks green. Once the port is configured, the LAN LED shows either a constant green light showing that the network setup was finished successfully and the ProDAQ 6100 is now accessible via the LAN or a constant red indicating a failure.

3.5.1. Network Considerations

By default the ProDAQ 6100 uses DHCP to configure its network interface. If no DHCP server is found in the network, it will attempt to obtain a network address using AutoIP. AutoIP addresses are allocated from the reserved range 169.254.0.0 -169.254.255.255. The ProDAQ 6100 will first try to use the address 169.254.x.y, where <x> and <y> are the two last octets of the devices MAC address. If the address is already in use, a new pair of <x> and <y> will be generated using a random number generator. By using the embedded web interface, the ProDAQ 6100 can also be configured to use a static IP address.

Note

The usage of AutoIP is only recommended for direct connections between the ProDAQ 6100 and a host computer.

If there is a dynamic DNS server available in the network, the instrument can be also accessed via its hostname. The default hostname is:

ProDAQ6100-<*serial number*>.<*domain*>

Where:

<*serial number*>	is the 8-digit serial number of the device,
<*domain*>	is defined by the Dynamic DNS server.

When using a multicast DNS tool, the domain is set to “local”. The host name can be also statically configured via the embedded web interface.

To reset the instruments network settings to the default configuration, press the recessed LAN reset button on the rear panel (see 3.3: Rear Panel Switches and Connectors).

3.5.2. Required Software

To operate not only the ProDAQ 6100 LXI Function Card Carrier itself, but also the ProDAQ Function Cards installed in it, you will need to install the Bustec VISA library on your computer. It can be downloaded from the Bustec web site at https://www.bustec.com/support/drivers/. The Bustec VISA library can be installed stand-alone, providing access to LXI, VXI, LAN-or USB based instruments; or in parallel to VISA libraries of other vendors.

For more details regarding the installation and use of the Bustec VISA library, please refer to the Bustec VISA user manual.

Note

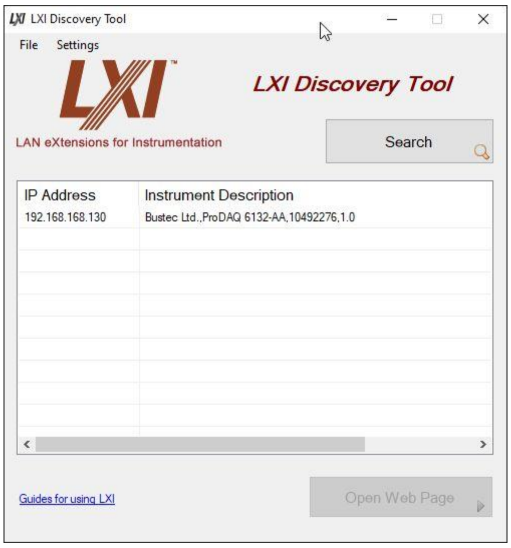
Make sure that the access of the VISA library, its tools, the function card soft front panels or your application is not blocked by the firewall of the host computer.

3.5.3. Network Discovery

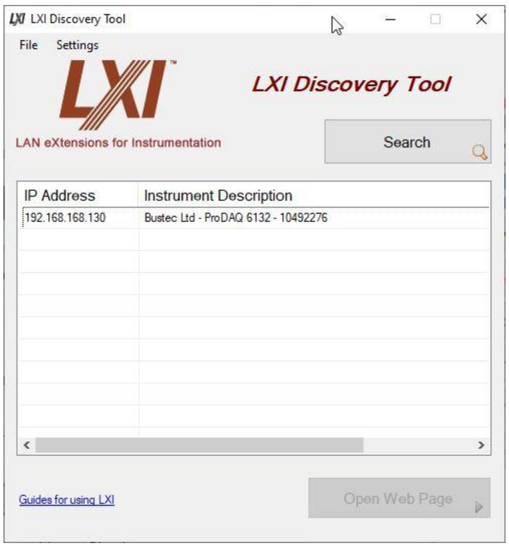
Once the ProDAQ 6100 is connected to a network and was assigned an IP address, there are several tools and ways to discover and access it.

LXI Discovery Tool

The LXI Discovery Tool is a standalone tool provided by the LXI Consortium. It uses the VXI-11 protocol or mDNS service discovery to discover LXI instruments in your LAN subnet. mDNS service recovery requires to install the Apple Bonjour Printer Services.



IP Address	Instrument Description
192.168.168.130	Bustec Ltd.,ProDAQ 6132-AA,10492276,1.0



IP Address	Instrument Description
192.168.168.130	Bustec Ltd - ProDAQ 6132 - 10492276

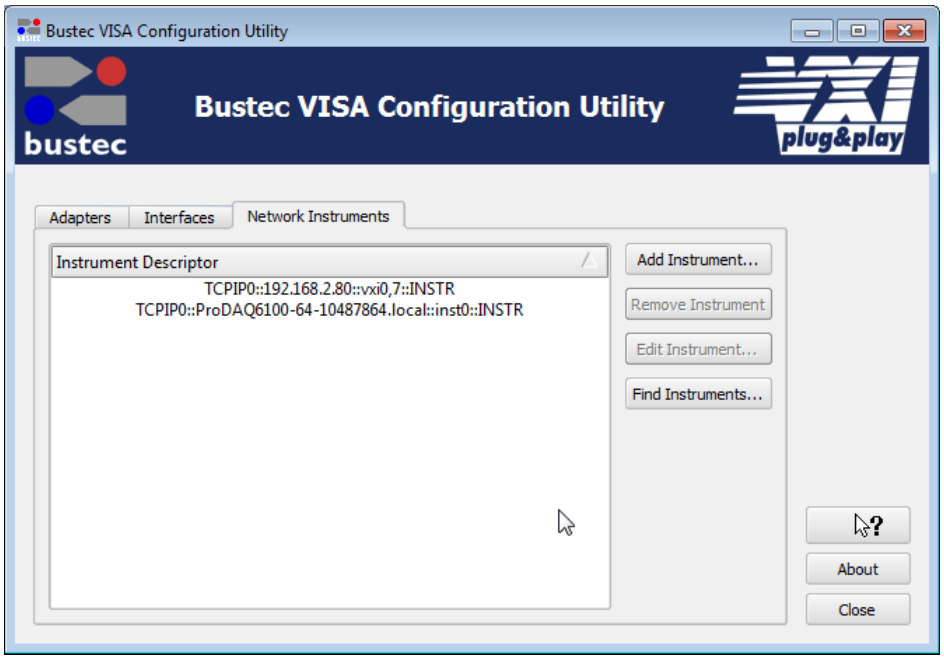
Figure 4 - LXI Discovery Tool

When discovery via the VXI11 protocol is used, each discovered device is queried using the "*IDN?" SCPI command. The device then responds with manufacturer, model, serial number, and firmware version as shown in the left screen shot. If mDNS service discovery is used, the mDNS service name is used as instrument description, as shown in the screen shot on the right.

For more information and downloading the LXI discovery tool, visit the web site of the LXI Consortium at https://lxistandard.org/Resources/LXIDiscoveryTool.aspx.

Bustec VISA

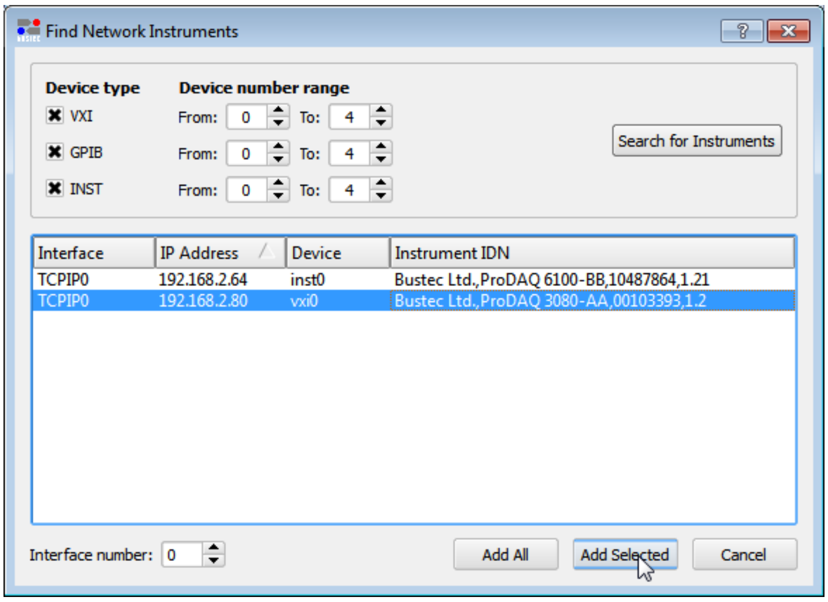
The Bustec VISA library contains a configuration tool which allows you to search, identify and configure network instruments. To view already known network instruments or add new, select the "Network Instruments" tab in the configuration utilities main dialog. All known network instruments will be displayed in the "Instrument Descriptor" list in the main dialog.



Instrument Descriptor
TCPIP0::192.168.2.80::vxi0,7::INSTR
TCPIP0::ProDAQ6100-64-10487864.local::inst0::INSTR

Figure 5 – Configuring Network Instruments

To search for network instruments located in your subnet, select the “Find Instruments ...” button to the right of the list of instrument descriptors. This opens the “Find Network Instruments” dialog, where you can set search parameters and perform the search.



Device type	From	To
☒ VXI	0	4
☒ GPIB	0	4
☒ INST	0	4

Interface	IP Address	Device	Instrument IDN
TCPIP0	192.168.2.64	inst0	Bustec Ltd.,ProDAQ 6100-BB,10487864,1.21
TCPIP0	192.168.2.80	vxi0	Bustec Ltd.,ProDAQ 3080-AA,00103393,1.2

Interface number: 0

Figure 6 - Searching for Network Instruments

To be found, the network instruments must comply to the VXI-11 standard and provide internal “VXIn” (VXI-11.1), “GPIBn” (VXI-11.2) or “INSTn” (VXI-11.3) interfaces and they must be able to respond to the “*IDN?” query. By checking/unchecking the boxes in front of the entries in the interface list the search can be limited to a particular type or set of types of interfaces. By setting the interface number ranges the search is limited to the selected range of interface numbers.

After selecting the “Search for Instruments” button, the utility uses network broadcasts to search for the instruments. Each instrument is then sent an “*IDN?” query and the results are shown in the “Find Network Instruments” dialog.

After a successful search, select the “Add All” button below the list to add all network instruments found to the list of known instruments or select instruments from the list and the “Add Selected” button to add only the instruments selected. The “Cancel” button allows you to close the dialog without adding any instrument.

If you run the Bustec VISA Agent you can discover the ProDAQ 6100 using mDNS discovery. The Bustec VISA Agent is a background task that monitors hot-plug events for ProDAQ VXIbus Interfaces and mDNS services published by LXI Devices. The agent can be accessed via its icon in the system tray.

If you select the icon, a popup menu is shown which allows access to the different applications included in the Bustec VISA installation and shows discovered network devices.

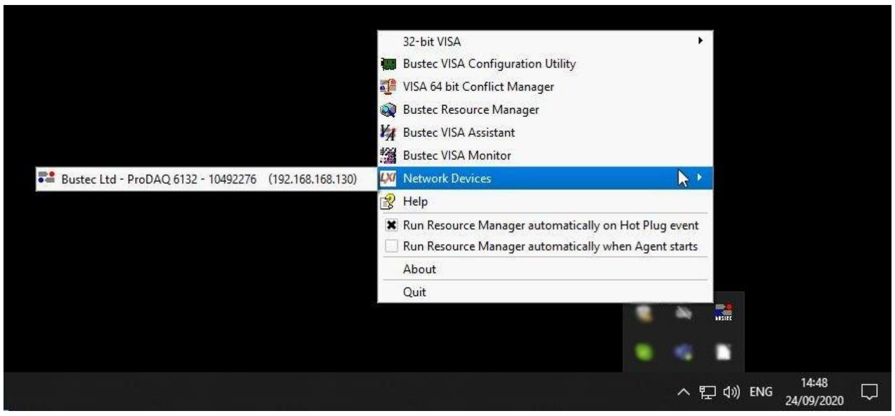

32-bit VISA
Bustec VISA Configuration Utility
VISA 64 bit Conflict Manager
Bustec Resource Manager
Bustec VISA Assistant
Bustec VISA Monitor
Network Devices
Help
Run Resource Manager automatically on Hot Plug event
Run Resource Manager automatically when Agent starts
About
Quit
Bustec Ltd - ProDAQ 6132 - 10492276 (192.168.168.130)
ENG
14:48
24/09/2020

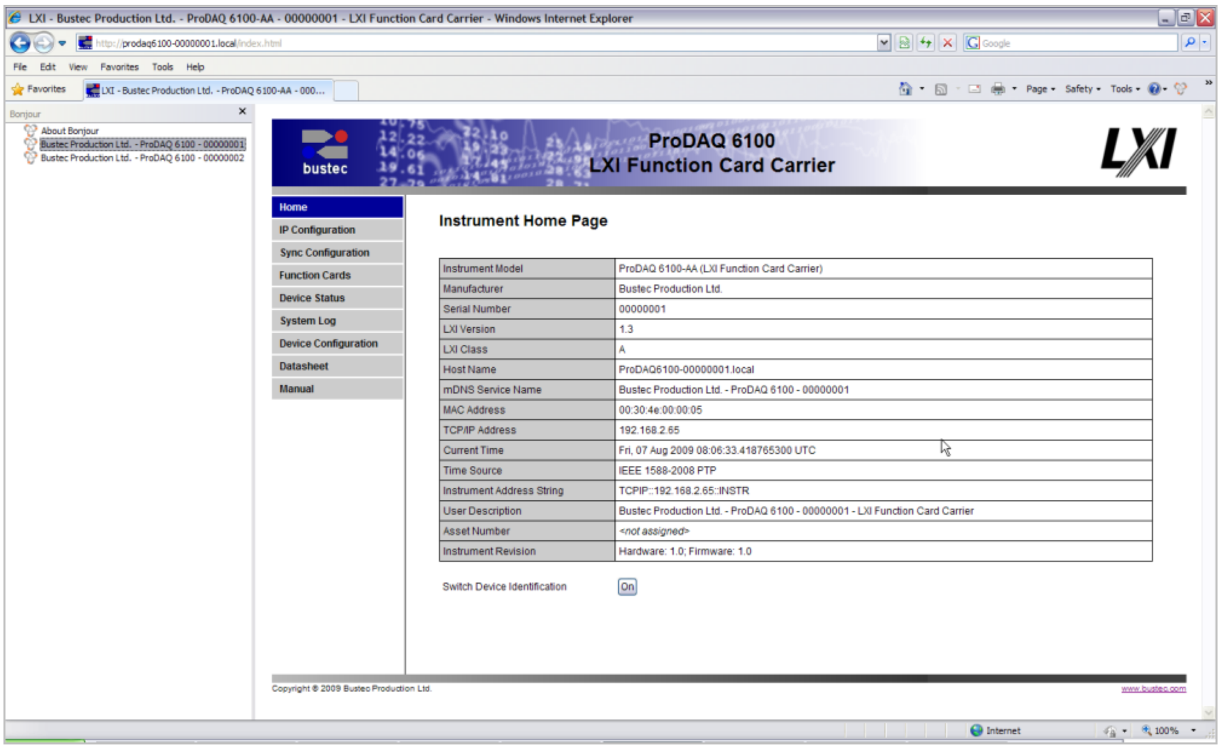
Figure 7 - Bustec Agent Network Device Discovery

As with the LXI Discovery Tool provided by the consortium, the mDNS discovery requires the installation of the Apple Bonjour Printer Services.

For more information, refer to the Bustec VISA User Manual.

3.5.4. Accessing the Instrument

Once the instrument was successfully discovered, you can access and configure the instrument via its web interface by a web browser using the instruments IP or mDNS host name.


LXI - Bustec Production Ltd. - ProDAQ 6100-AA - 00000001 - LXI Function Card Carrier - Windows Internet Explorer
http://prodaq6100-00000001.local/index.html
File Edit View Favorites Tools Help
Favorites
LXI - Bustec Production Ltd. - ProDAQ 6100-AA - 000...
Page Safety Tools
Bonjour
About Bonjour
Bustec Production Ltd. - ProDAQ 6100 - 00000001
Bustec Production Ltd. - ProDAQ 6100 - 00000002
bustec
ProDAQ 6100
LXI Function Card Carrier
LXI
Home
IP Configuration
Sync Configuration
Function Cards
Device Status
System Log
Device Configuration
Datasheet
Manual
Internet
100%

Instrument Home Page

Instrument Model	ProDAQ 6100-AA (LXI Function Card Carrier)
Manufacturer	Bustec Production Ltd.
Serial Number	00000001
LXI Version	1.3
LXI Class	A
Host Name	ProDAQ6100-00000001.local
mDNS Service Name	Bustec Production Ltd. - ProDAQ 6100 - 00000001
MAC Address	00:30:4e:00:00:05
TCP/IP Address	192.168.2.65
Current Time	Fri, 07 Aug 2009 08:06:33.418765300 UTC
Time Source	IEEE 1588-2008 PTP
Instrument Address String	TCPIP::192.168.2.65::INSTR
User Description	Bustec Production Ltd. - ProDAQ 6100 - 00000001 - LXI Function Card Carrier
Asset Number	*<not assigned>*
Instrument Revision	Hardware: 1.0; Firmware: 1.0

Switch Device Identification [On]

Copyright © 2009 Bustec Production Ltd. www.bustec.com

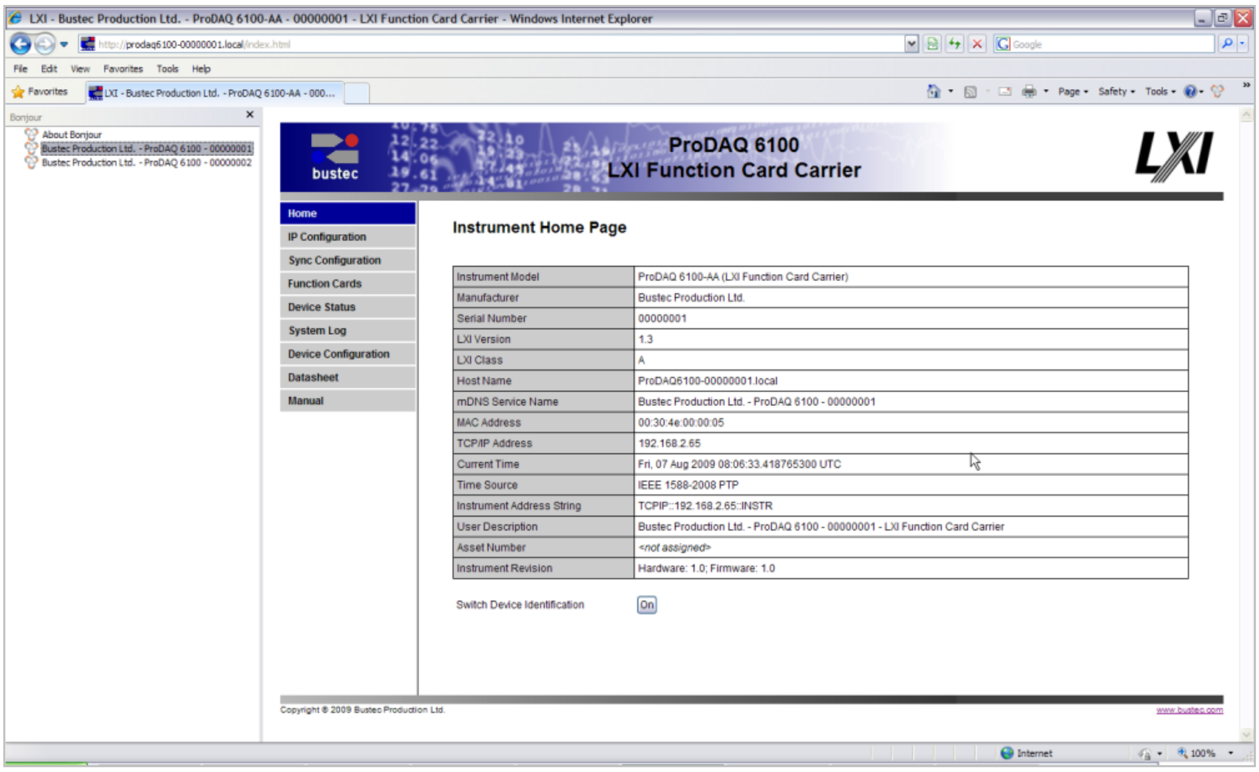
Figure 8 - Access the instrument home page

4. WEB Page Operation

The ProDAQ 6100 features an embedded web server, which allows you to configure and operate the ProDAQ 6100 by using a standard web browser from any host computer in your network. To make use of the complete functionality of the embedded web interface, the browser will need to have its JavaScript support enabled.

4.1. Instrument Home Page

The instrument home page shows general information about the device like model number, manufacturer, serial number, and revisions.


LXI - Bustec Production Ltd. - ProDAQ 6100-AA - 00000001 - LXI Function Card Carrier - Windows Internet Explorer
http://prodaq6100-00000001.local/index.html
File Edit View Favorites Tools Help
Favorites
Bonjour
About Bonjour
Bustec Production Ltd. - ProDAQ 6100 - 00000001
Bustec Production Ltd. - ProDAQ 6100 - 00000002
bustec
ProDAQ 6100
LXI Function Card Carrier
LXI
Home
IP Configuration
Sync Configuration
Function Cards
Device Status
System Log
Device Configuration
Datasheet
Manual
On
Copyright © 2009 Bustec Production Ltd.
www.bustec.com
Internet
100%

Instrument Home Page

Instrument Model	ProDAQ 6100-AA (LXI Function Card Carrier)
Manufacturer	Bustec Production Ltd.
Serial Number	00000001
LXI Version	1.3
LXI Class	A
Host Name	ProDAQ6100-00000001.local
mDNS Service Name	Bustec Production Ltd. - ProDAQ 6100 - 00000001
MAC Address	00:30:4e:00:00:05
TCP/IP Address	192.168.2.65
Current Time	Fri, 07 Aug 2009 08:06:33.418765300 UTC
Time Source	IEEE 1588-2008 PTP
Instrument Address String	TCPIP::192.168.2.65::INSTR
User Description	Bustec Production Ltd. - ProDAQ 6100 - 00000001 - LXI Function Card Carrier
Asset Number	*<not assigned>*
Instrument Revision	Hardware: 1.0; Firmware: 1.0

Switch Device Identification

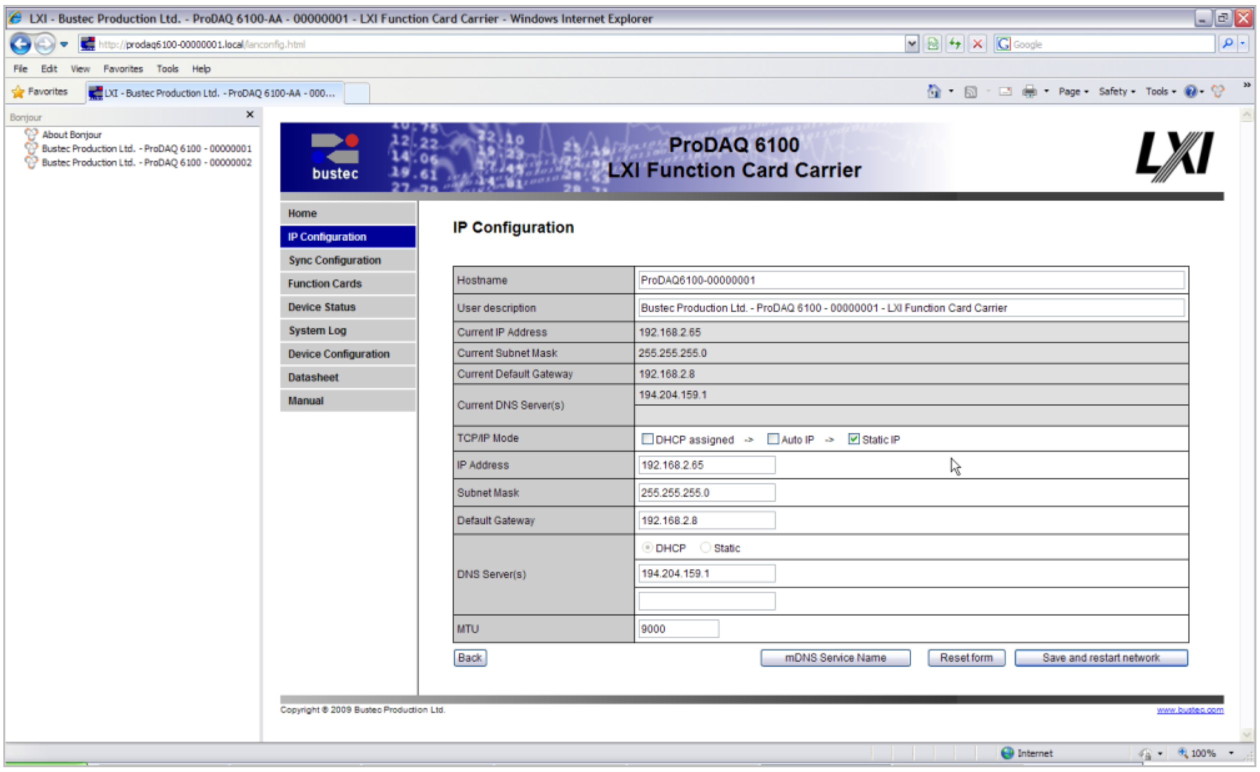
Figure 9 - Instrument Home Page

From here you can navigate to the different categories and pages by using the menu on the left side. For security reasons, all pages except of the instruments home page are protected by username and password, which can be configured on the "Device Configuration" -> "Security Settings" page. Upon delivery, the username is set to "admin" and password to "1234".

The <On/Off> button on the instrument home page allows you to identify physically the ProDAQ 6100 you are connected to. When you click onto the button, the LAN status indicator on the front bezel will start flashing. A second click switches the LAN indicator off again.

4.2. IP Configuration

The IP Configuration Page allows you to change the settings for the ProDAQ 6100's LAN interface.


IP Configuration

Hostname	ProDAQ6100-00000001
User description	Bustec Production Ltd. - ProDAQ 6100 - 00000001 - LXI Function Card Carrier
Current IP Address	192.168.2.65
Current Subnet Mask	255.255.255.0
Current Default Gateway	192.168.2.8
Current DNS Server(s)	194.204.159.1
TCP/IP Mode	☐ DHCP assigned -> ☐ Auto IP -> ☑ Static IP
IP Address	192.168.2.65
Subnet Mask	255.255.255.0
Default Gateway	192.168.2.8
DNS Server(s)	[✓ DHCP] Static
	194.204.159.1
MTU	9000

LXI - Bustec Production Ltd. - ProDAQ 6100-AA - 00000001 - LXI Function Card Carrier - Windows Internet Explorer
http://prodaq6100-00000001.local/lanconfig.html
ProDAQ 6100
LXI Function Card Carrier
bustec
Home
IP Configuration
Sync Configuration
Function Cards
Device Status
System Log
Device Configuration
Datasheet
Manual
Back
mDNS Service Name
Reset form
Save and restart network
Copyright © 2009 Bustec Production Ltd.
www.bustec.com

Figure 10 - IP Configuration Page

The IP Configuration page shows the current settings for the instruments LAN interface and allows you to change and store the following settings:

Hostname User defined hostname for the device (without domain).

Clear this value to revert to factory default.

Note: Multicast DNS domain is always: ".local". Dynamic DNS domain depends on the network configuration.

User Description User defined description of the device – it is displayed on the Home Page along with user defined Asset Number (see Device Configuration).

Clear this value to revert to factory default.

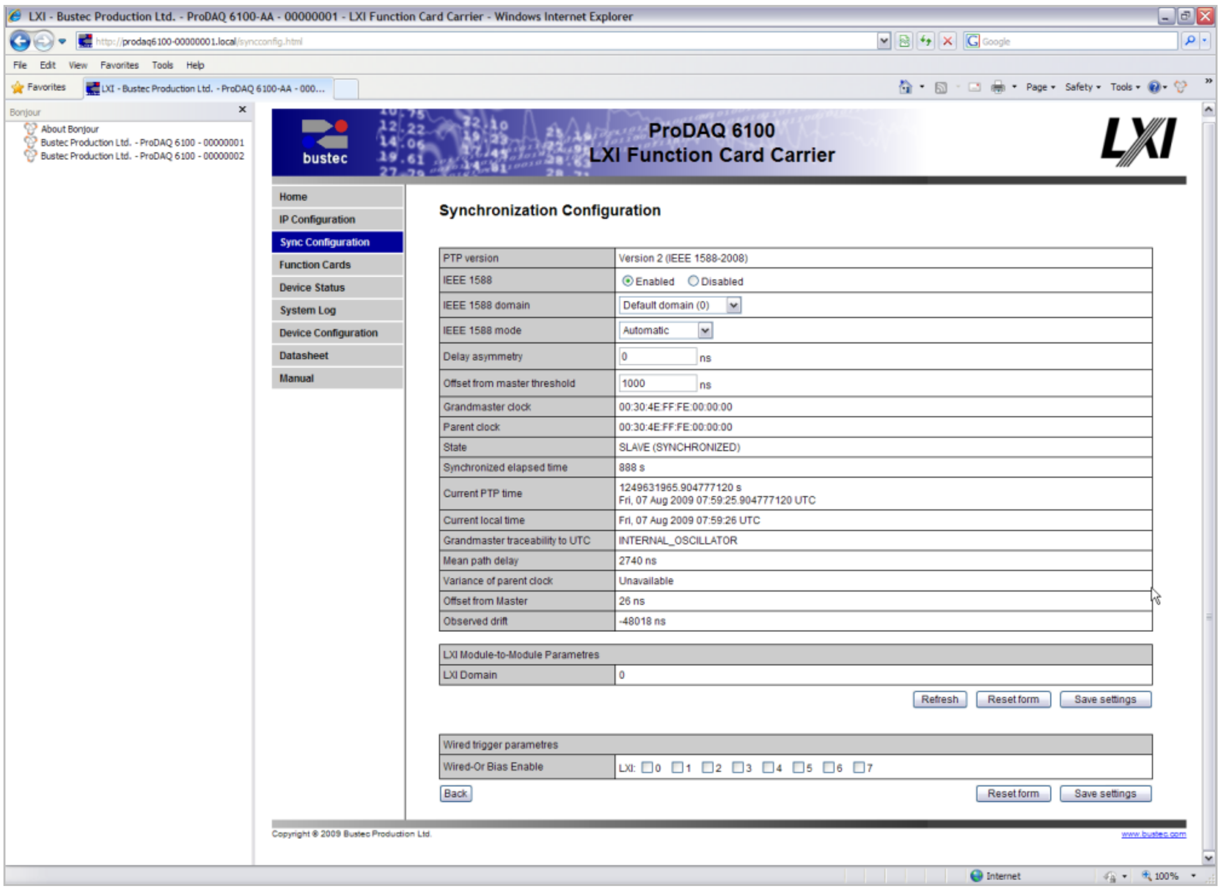
Current IP configuration Displays currently assigned: IP Address, Subnet Mask, Default Gateway and DNS servers.

TCP/IP mode	Specifies whether the device shall use a DHCP server in the network, or AutoIP protocol to automatically obtain the IP configuration, or maybe the static IP configuration defined in the form below. More than one option may be selected. The priority is as follows: DHCP → AutoIP → Static. For example, if DHCP and Static are selected and DHCP fails, the Static configuration is set.
IP Address	If "Static IP" was selected as the TCP/IP mode, this field allows assignment of a static IP address to the ProDAQ 6100s LAN interface.
Subnet mask	If "Static IP" was selected as the TCP/IP mode, this field allows assignment of a static subnet mask address to the ProDAQ 6100s LAN interface.
Default Gateway	If "Static IP" was selected as the TCP/IP mode, this field allows assignment of a static default gateway for the routing of IP packets.
DNS Servers	If "Static IP" was selected as the TCP/IP mode, these two fields allow you to specify the DNS server the ProDAQ 6100 will use for name resolving. If “DHCP” was selected as the TCP/IP mode, then it is possible to select whether the DNS servers’ IP addresses shall be acquired automatically (DHCP) or user-defined (Static).
MTU	Maximum Transmission Unit (MTU) – maximum size (in bytes) of an IP packet that can be transmitted without fragmentation (including IP headers, but excluding headers from lower levels in the protocol stack). The default value for a typical network is 1500 B. It can be defined as high as 9000 B (jumbo frames). For correct interoperation, the whole network must have the same MTU. For best performance, it is recommended to configure the network to work with as high MTU as possible.
mDNS Service Name	User defined name of mDNS services that are advertised by the ProDAQ 6100 device. Clear this value to revert to factory default.

The device has a LAN reset mechanism that restores all the IP and IEEE1588 configuration back to factory defaults. In order to reset the settings press the “RST” button that can be found in the rear panel of the ProDAQ 6100.

4.3. Synchronization Configuration

The ProDAQ 6100 uses IEEE 1588-2008 (version 2) for clock synchronization. The Synchronization Configuration page shows the current IEEE 1588 status and allows its configuration. You can also configure here whether the LXI triggers of the device shall act as bias in Wired-Or trigger chain.


LXI - Bustec Production Ltd. - ProDAQ 6100-AA - 00000001 - LXI Function Card Carrier - Windows Internet Explorer
http://prodaq6100-00000001.local/syncconfig.html
File Edit View Favorites Tools Help
Bonjour
About Bonjour
Bustec Production Ltd. - ProDAQ 6100 - 00000001
Bustec Production Ltd. - ProDAQ 6100 - 00000002
bustec
ProDAQ 6100
LXI Function Card Carrier
LXI
Home
IP Configuration
Sync Configuration
Function Cards
Device Status
System Log
Device Configuration
Datasheet
Manual

Synchronization Configuration

PTP version	Version 2 (IEEE 1588-2008)
IEEE 1588	[✓ Enabled] Disabled
IEEE 1588 domain	Default domain (0)
IEEE 1588 mode	Automatic
Delay asymmetry	0 ns
Offset from master threshold	1000 ns
Grandmaster clock	00:30:4E:FF:FE:00:00:00
Parent clock	00:30:4E:FF:FE:00:00:00
State	SLAVE (SYNCHRONIZED)
Synchronized elapsed time	888 s
Current PTP time	1249631965.904777120 s Fri, 07 Aug 2009 07:59:25.904777120 UTC
Current local time	Fri, 07 Aug 2009 07:59:26 UTC
Grandmaster traceability to UTC	INTERNAL_OSCILLATOR
Mean path delay	2740 ns
Variance of parent clock	Unavailable
Offset from Master	26 ns
Observed drift	-48018 ns

LXI Module-to-Module Parametres	
LXI Domain	0

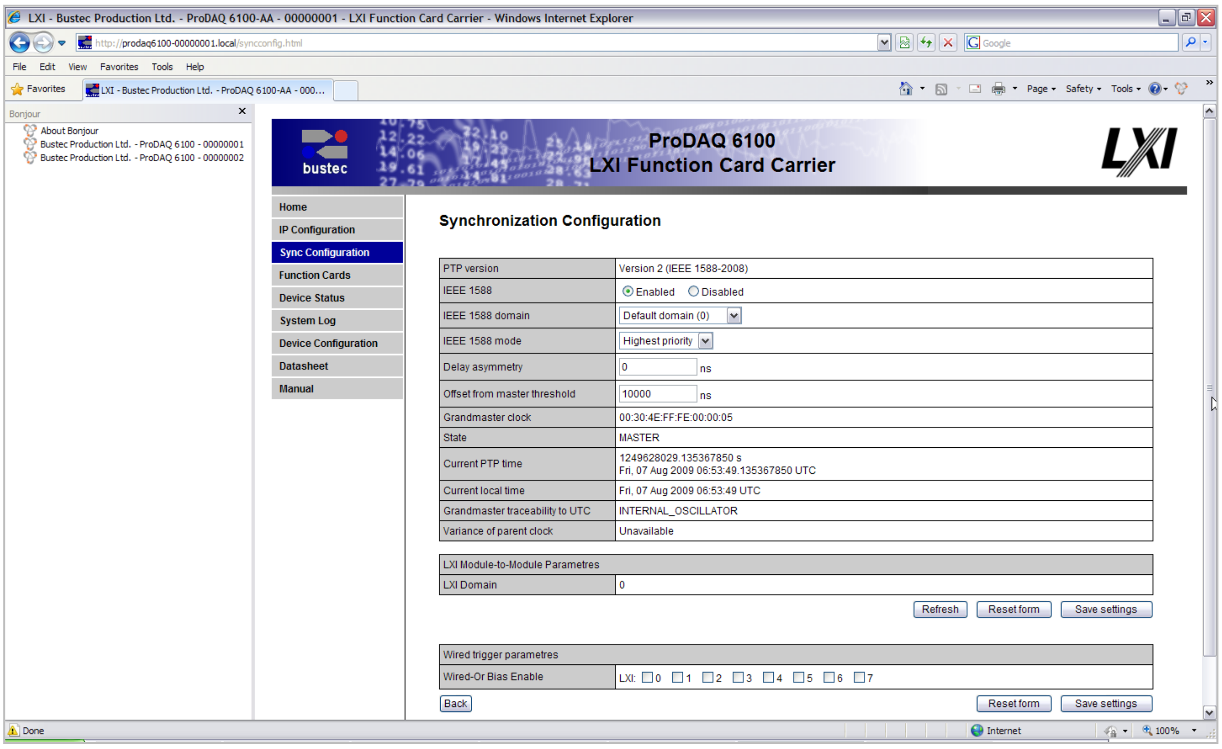
Wired trigger parametres	
Wired-Or Bias Enable	LXI: ☐0 ☐1 ☐2 ☐3 ☐4 ☐5 ☐6 ☐7

Figure 11 - Synchronization Configuration Page (Slave Mode)

IEEE 1588 domain Specifies the IEEE 1588 domain number – logical part of the network in which the device should work. Default domain is 0.

IEEE 1588 mode Specifies the priority of the device for the master clock selection procedure. “Slave only” mode ensures that the device cannot become a master clock in any network – it is always slave. “Highest priority” mode means that the device has highest chances of becoming a master. It may remain slave however if there is another device in the network with highest priority defined that advertises itself as having a better clock accuracy. The “Automatic” mode on the other hand relies only on the best clock selection algorithm.

Delay asymmetry	For better performance, this parameter should reflect the network architecture that affects the difference between the time a packet gets from one device to another and the return time. If the return time is longer – the value should be positive. By default this value is 0 ns.
Offset from Master threshold	If the "Offset from Master" value is below the user defined threshold the device is considered as synchronized.


LXI - Bustec Production Ltd. - ProDAQ 6100-AA - 00000001 - LXI Function Card Carrier - Windows Internet Explorer
http://prodaq6100-00000001.local/syncconfig.html
Bonjour
About Bonjour
Bustec Production Ltd. - ProDAQ 6100 - 00000001
Bustec Production Ltd. - ProDAQ 6100 - 00000002
bustec
ProDAQ 6100
LXI Function Card Carrier
LXI
Home
IP Configuration
Sync Configuration
Function Cards
Device Status
System Log
Device Configuration
Datasheet
Manual
Refresh
Reset form
Save settings
Back
Done
Internet
100%

Synchronization Configuration

PTP version	Version 2 (IEEE 1588-2008)
IEEE 1588	[✓ Enabled] Disabled
IEEE 1588 domain	Default domain (0)
IEEE 1588 mode	Highest priority
Delay asymmetry	0 ns
Offset from master threshold	10000 ns
Grandmaster clock	00:30:4E:FF:FE:00:00:05
State	MASTER
Current PTP time	1249628029.135367850 s Fri, 07 Aug 2009 06:53:49.135367850 UTC
Current local time	Fri, 07 Aug 2009 06:53:49 UTC
Grandmaster traceability to UTC	INTERNAL_OSCILLATOR
Variance of parent clock	Unavailable

LXI Module-to-Module Parametres	
LXI Domain	0

Wired trigger parametres	
Wired-Or Bias Enable	LXI: ☐0 ☐1 ☐2 ☐3 ☐4 ☐5 ☐6 ☐7

Figure 12 - Synchronization Configuration Page (Master Mode)

The IEEE 1588 status parameters are:

Grandmaster clock	Grandmaster clock ID (MAC address extended to IEEE EUI-64).
Parent clock	Parent clock ID (MAC address extended to IEEE EUI-64).
State	Current state of the IEEE 1588 protocol engine working on the ProDAQ 6100. Can be one of the following: INITIALIZING, FAULTY, DISABLED, LISTENING, PRE_MASTER, MASTER, PASSIVE, UNCALIBRATED. In case of SLAVE state also the synchronization state is displayed:

	SYNCHRONIZED or NOT SYNCHRONIZED.
Synchronized elapsed time	Time in seconds since the device remains constantly synchronized (considering the “Offset from Master threshold”).
Current PTP time	Shows the current IEEE 1588 time kept by the device. Seconds since midnight, January 1st, 1970 (represented as “seconds.fractional seconds”). Additionally a formatted date is also displayed.
Current local time	Local system time kept by the system clock on the device. It may differ from the PTP time when the PTP state is not Master.
Grandmaster traceability to UTC	Displays a string defined by IEEE 1588 standard that defines how the grandmaster clock is related to the Universal Coordinated Time (UTC).
Mean path delay	The average value of time in which the packet gets to the Master and the time the response gets back from Master.
Variance of parent clock	This value is unavailable.
Offset from Master	Current observed difference between the Master and the Slave clock.
Observed drift	An implementation specific value which is related to the precision of the clock oscillator.

Please note that only the parameters that are relevant to the current IEEE 1588 state are displayed on the page (see Figure 11 and Figure 12).

The “LXI Domain” parameter displays the LXI domain number used in the module-to-module data communication (LXI Sync messages).

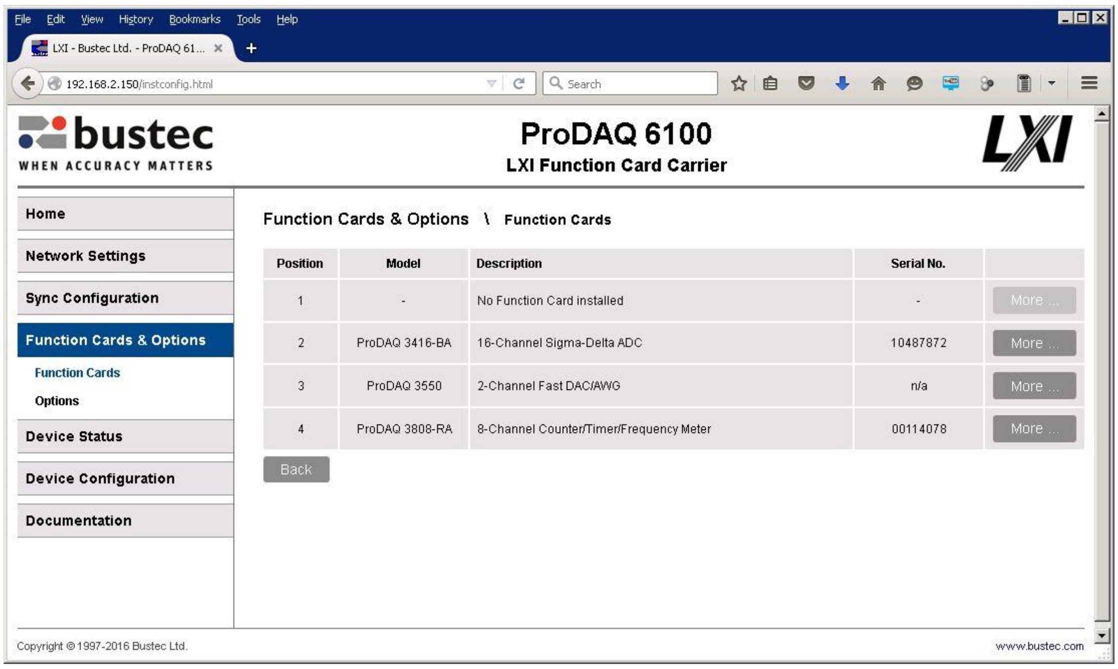
In the “Wired Trigger Parameters” section it is possible to select whether a trigger line on the LXI trigger bus is enabled to act as a bias for a Wired-OR chain. In any Wired-Or chain there shall be exactly one device acting as a bias for the chain.

4.4. Function Cards & Options

The pages under “Function Cards & Options” show the functions cards and options like for example the ProDAQ 3202 Voltage Reference Plug-in installed in the ProDAQ 6100.

4.4.1. Function Cards

The first page, “Function Cards”, gives an overview about the function cards installed. Each function card is listed with its model number, description and serial number (when available).



Function Cards & Options \ Function Cards

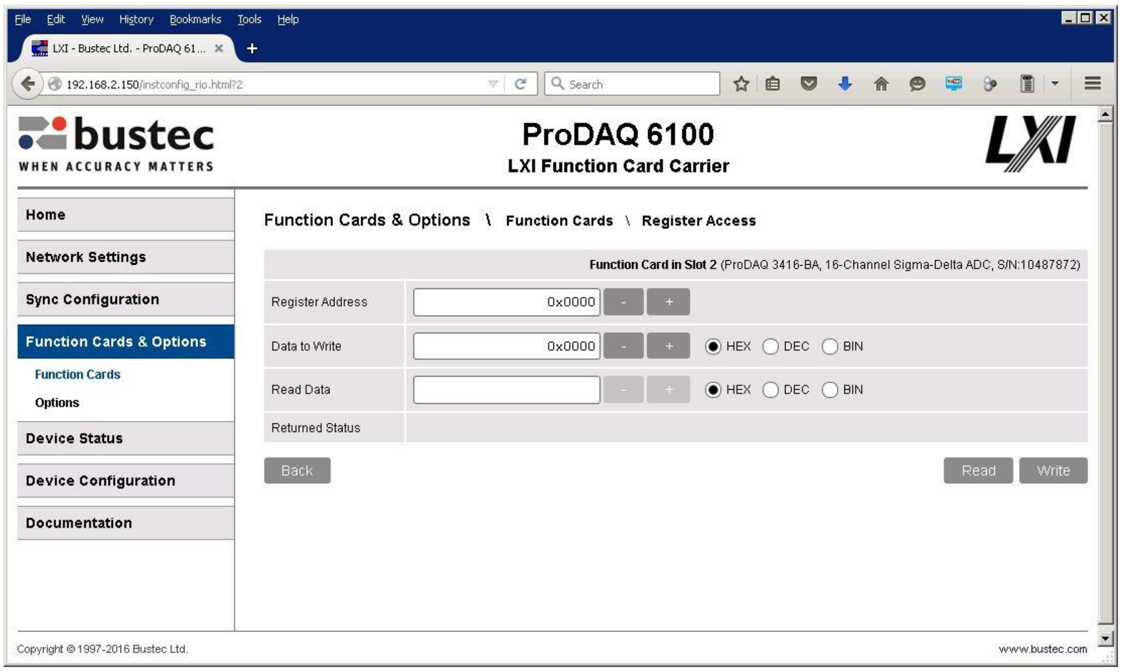
Position	Model	Description	Serial No.	
1	-	No Function Card installed	-	More ...
2	ProDAQ 3416-BA	16-Channel Sigma-Delta ADC	10487872	More ...
3	ProDAQ 3550	2-Channel Fast DAC/AWG	n/a	More ...
4	ProDAQ 3808-RA	8-Channel Counter/Timer/Frequency Meter	00114078	More ...

Figure 13 - Function Card Page

Selecting the “More …” button for one of the function cards opens the next page (see below), which allows a basic read/write access to the function card registers.

Function Card Register Access

The “Register Access” page allows basic read/write access of the function card registers.



Function Cards & Options \ Function Cards \ Register Access

Function Card in Slot 2 (ProDAQ 3416-BA, 16-Channel Sigma-Delta ADC, S/N:10487872)

Register Address	0x0000	
Data to Write	0x0000	[✓ HEX] DEC BIN
Read Data		[✓ HEX] DEC BIN
Returned Status		

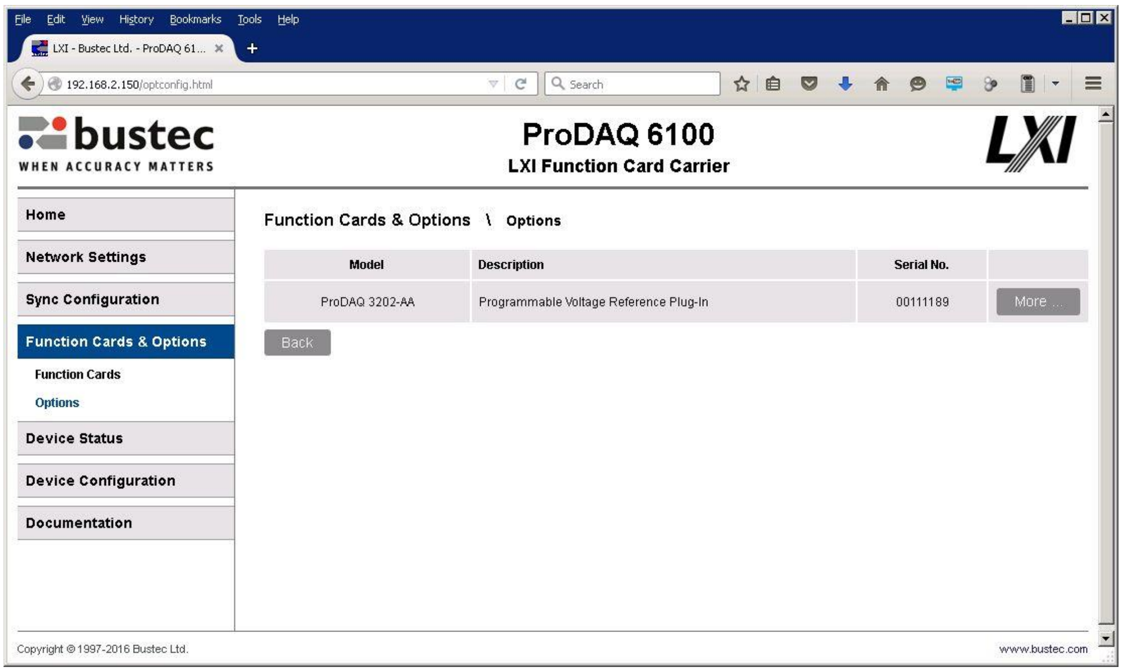
Figure 14 - Function Card Access Page

To read a function card register, enter the register address in the “Register Address” field or use the “+”/”-“ buttons to change/set the desired address and then select the “Read” button below. After completion of the read operation, the “Read Data” field display the data read from the register and the “Returned Status” field shows whether the operation was successful or not.

To write to a function card register, you need to specify the value to write in addition to the register address in the “Data to Write” field. After you have entered or changed both the address and data to write, select the “Write” button below. After completion of the write operation, the “Returned Status” field shows whether the operation was successful or not.

4.4.2. Options

The second page “Options”, shows all options installed in the ProDAQ 6100.



Model	Description	Serial No.	
ProDAQ 3202-AA	Programmable Voltage Reference Plug-In	00111189	More ...

Figure 15 - Options Page

Selecting the “More …” button to the right of the listed option opens a page that shows the details for the option, like for example the calibration data for the ProDAQ 3202 Voltage Reference Plug-in.

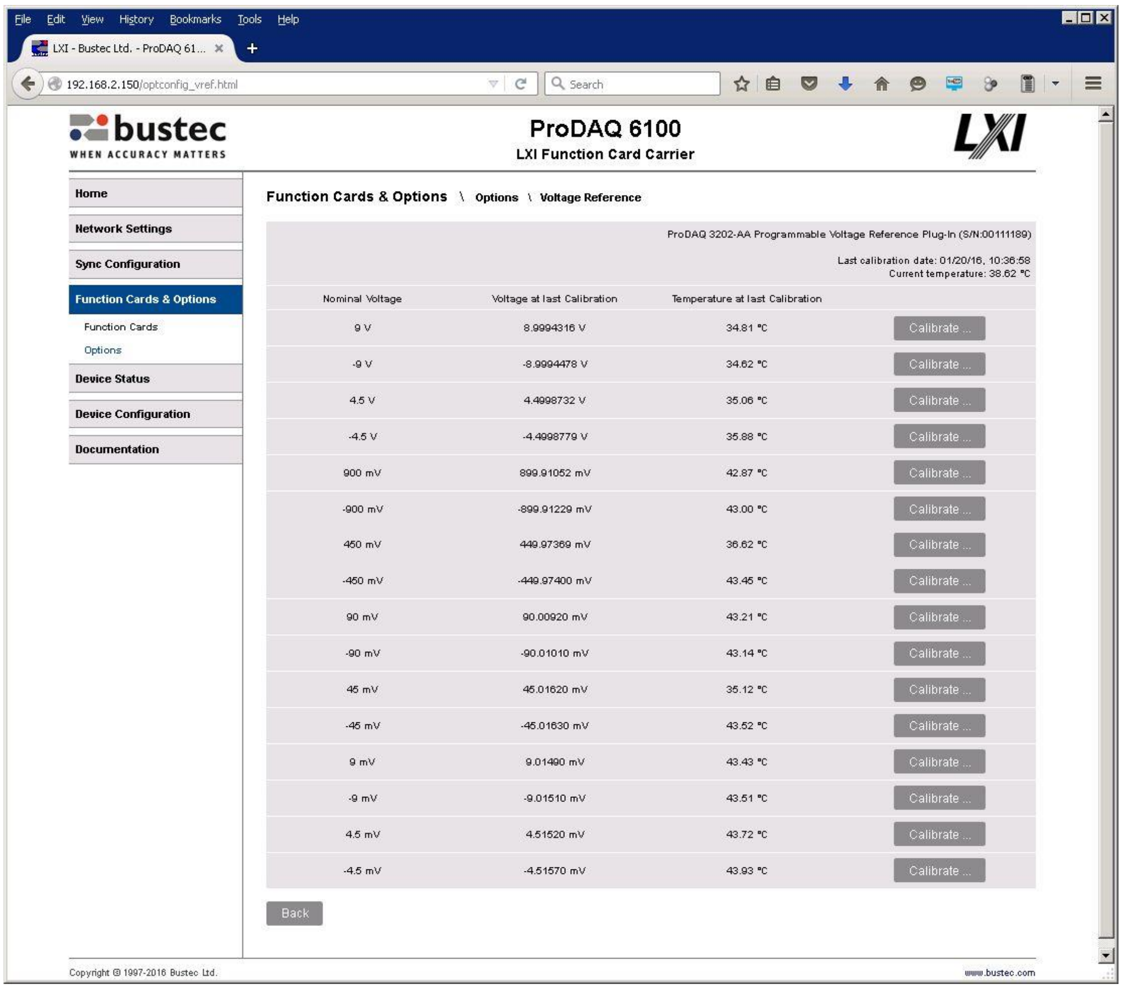
ProDAQ 3202 Voltage Reference Plug-In

The ProDAQ 3202 Voltage Reference Plug-in module provides a set of highly stable voltages that can be used to calibrate the ProDAQ function cards and signal conditioning cards and modules on-the-fly before any measurement without disconnecting any sensor cables. The output voltage of the ProDAQ 3202 is distributed via the internal voltage reference bus to the function cards and via the I/O cables further on to the ProDAQ signal conditioning cards and modules, where it then can be switched into the input of the front-most input circuitry. The software drivers use this feature to provide fully automated calibration routines to the user.

During calibration the output voltages of the ProDAQ 3202 are not adjusted. Instead the precise value measured is stored in an on-board EEPROM and used by the calibration routines. This provides a better precision and what is more, a higher stability than provided by an adjustable circuitry.

The table shown in on the “Voltage Reference” page (Figure 16) lists all voltages by their nominal values and stored calibrated values. It also shows both the current temperature and the temperature at the last calibration. Please note that this is the temperature measured on the ProDAQ 3202 Voltage Reference Plug-in and not the ambient temperature.

The “Calibrate …” bottom to the right of each voltage listing allows the re-calibration of the particular setting by the user. To do so, a precision voltage meter must be connected to the voltage reference monitor output of the ProDAQ 6100 (see 3.3: Rear Panel Switches and Connectors, “VREF”).



Function Cards & Options \ Options \ Voltage Reference

ProDAQ 3202-AA Programmable Voltage Reference Plug-In (S/N:00111189)

Last calibration date: 01/20/16, 10:36:58
Current temperature: 38.62 °C

Nominal Voltage	Voltage at last Calibration	Temperature at last Calibration	
9 V	8.9994316 V	34.81 °C	Calibrate ...
-9 V	-8.9994478 V	34.62 °C	Calibrate ...
4.5 V	4.4998732 V	35.06 °C	Calibrate ...
-4.5 V	-4.4998779 V	35.88 °C	Calibrate ...
900 mV	899.91052 mV	42.87 °C	Calibrate ...
-900 mV	-899.91229 mV	43.00 °C	Calibrate ...
450 mV	449.97369 mV	36.62 °C	Calibrate ...
-450 mV	-449.97400 mV	43.45 °C	Calibrate ...
90 mV	90.00920 mV	43.21 °C	Calibrate ...
-90 mV	-90.01010 mV	43.14 °C	Calibrate ...
45 mV	45.01620 mV	35.12 °C	Calibrate ...
-45 mV	-45.01630 mV	43.52 °C	Calibrate ...
9 mV	9.01490 mV	43.43 °C	Calibrate ...
-9 mV	-9.01510 mV	43.51 °C	Calibrate ...
4.5 mV	4.51520 mV	43.72 °C	Calibrate ...
-4.5 mV	-4.51570 mV	43.93 °C	Calibrate ...

Figure 16 - Voltage Reference Calibration Data

WARNING

Changing the calibration values of the ProDAQ 3202 will directly influence the precision of any measurements taken. Please make sure that you use a calibrated voltage meter providing at least the accuracy stated in the ProDAQ 3202 datasheet.

Note

When calibrating the voltage settings it is of utmost importance that the ProDAQ 6100 with the ProDAQ 3202 was warmed up at least for one hour. This applies also when doing measurements. Only then can be guaranteed that all parts of the system are equally warmed up and that the drift caused by temperature changes is minimized.

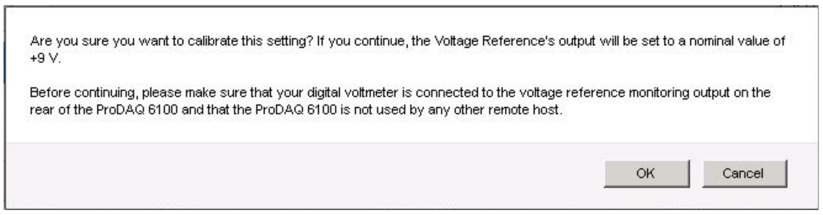
WARNING

While using the following procedure to calibrate the ProDAQ 3202, please make sure that the ProDAQ 6100 it is installed in is not otherwise used or accessed. Interrupting the process may cause invalid data to be stored and may require to

recalibrate the ProDAQ 3202 again.

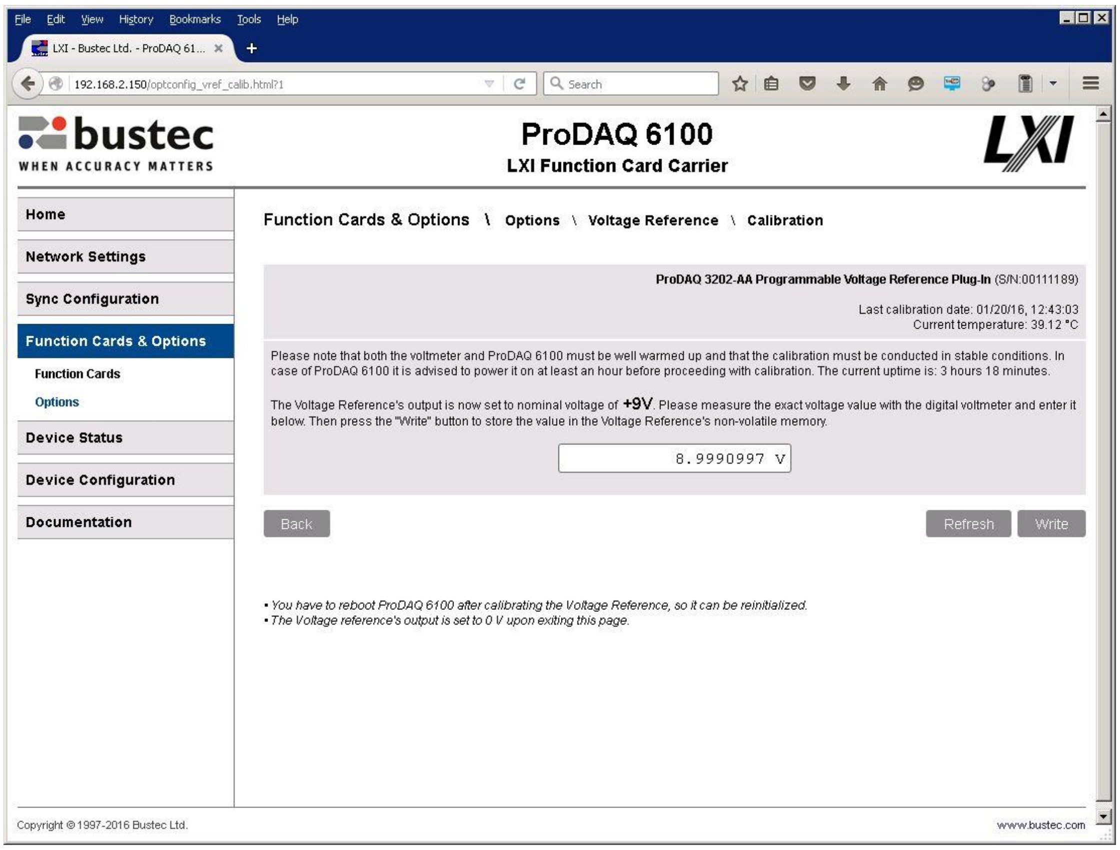
Calibration Procedure

When selecting one of the "Calibrate …" buttons to the right of a voltage listed, the following pop-up will be shown:


Are you sure you want to calibrate this setting? If you continue, the Voltage Reference's output will be set to a nominal value of +9 V.

Before continuing, please make sure that your digital voltmeter is connected to the voltage reference monitoring output on the rear of the ProDAQ 6100 and that the ProDAQ 6100 is not used by any other remote host.

When selecting "OK", the ProDAQ 3202 is set to generate the selected voltage and the output is enabled, allowing you to measure the voltage. On the next page shown the new measured voltage can be entered:



Function Cards & Options \ Options \ Voltage Reference \ Calibration

ProDAQ 3202-AA Programmable Voltage Reference Plug-In (S/N:00111189)

Last calibration date: 01/20/16, 12:43:03
Current temperature: 39.12 °C

Please note that both the voltmeter and ProDAQ 6100 must be well warmed up and that the calibration must be conducted in stable conditions. In case of ProDAQ 6100 it is advised to power it on at least an hour before proceeding with calibration. The current uptime is: 3 hours 18 minutes.

The Voltage Reference's output is now set to nominal voltage of +9V. Please measure the exact voltage value with the digital voltmeter and enter it below. Then press the "Write" button to store the value in the Voltage Reference's non-volatile memory.

8.9990997 V

- *You have to reboot ProDAQ 6100 after calibrating the Voltage Reference, so it can be reinitialized.*
- *The Voltage reference's output is set to 0 V upon exiting this page.*

Copyright © 1997-2016 Bustec Ltd. www.bustec.com

The "Refresh" button allows updating the information regarding temperature and uptime shown on the page. "Back" allows navigating back to the previous page.

If a new calibration voltage shall be stored, the measured value must be entered into the input field provided and the "Write" button must be selected. The data in the input field must be entered as floating point value followed by a unit, either "V" or "mV".

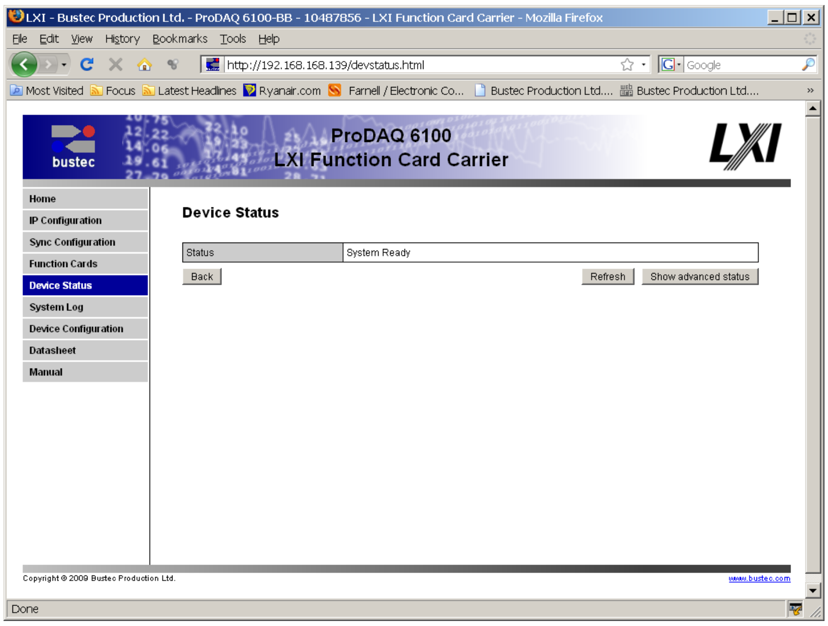
After confirming the value in the next dialog, the value is stored together with the current date and temperature in the onboard EEPROM of the ProDAQ 6100, the ProDAQ 3202 output is switched back to ground and the page providing the listing of calibration data is shown.

Please note that the calibration data in the EEPROM of the ProDAQ 3202 is stored as 32-bit floating point values, so the resolution for the nominal values +9V, -9V, +4.5V and -4.5V is 119 nV and for all other nominal values 119 pV. The setting stored will be rounded to the nearest possible value of the number entered.

To calibrate additional settings, one of the other “Calibrate …” buttons in the table can be selected, invoking the procedure for one of the other values. When finished calibrating, the ProDAQ 6100 needs to be restarted to re-read the values from the EEPROM of the ProDAQ 3202.

4.5. Device Status

The Device Status page display the general status of the device and allows access to more detailed information as well.


LXI - Bustec Production Ltd. - ProDAQ 6100-BB - 10487856 - LXI Function Card Carrier - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://192.168.168.139/devstatus.html
Most Visited Focus Latest Headlines Ryanair.com Farnell / Electronic Co... Bustec Production Ltd.... Bustec Production Ltd....
bustec
ProDAQ 6100
LXI Function Card Carrier
LXI
Home
IP Configuration
Sync Configuration
Function Cards
Device Status
System Log
Device Configuration
Datasheet
Manual
Back
Refresh
Show advanced status
Copyright © 2009 Bustec Production Ltd.
www.bustec.com
Done

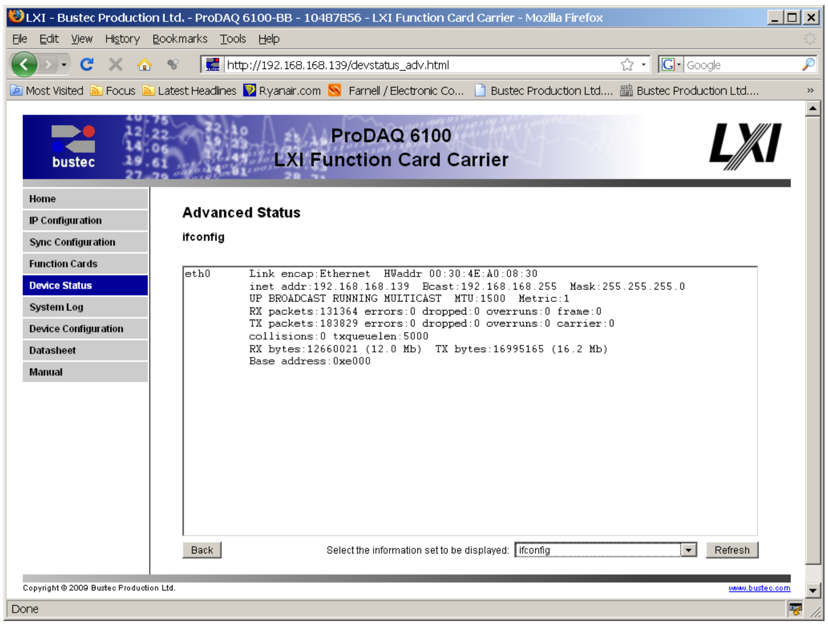
Device Status

Status	System Ready

Figure 17 - Device Status Page

4.5.1. Advanced Status

The Advanced Status page allows you to review the output of a number of embedded tools to analyze the device status in more details.


LXI - Bustec Production Ltd. - ProDAQ 6100-BB - 10487856 - LXI Function Card Carrier - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://192.168.168.139/devstatus_adv.html
ProDAQ 6100
LXI Function Card Carrier
Home
IP Configuration
Sync Configuration
Function Cards
Device Status
System Log
Device Configuration
Datasheet
Manual
Advanced Status
ifconfig
Back
Select the information set to be displayed:
ifconfig
Refresh
Copyright © 2009 Bustec Production Ltd.
www.bustec.com
Done

```
eth0      Link encap:Ethernet  HWaddr 00:30:4E:A0:08:30
          inet addr:192.168.168.139  Bcast:192.168.168.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:131364 errors:0 dropped:0 overruns:0 frame:0
          TX packets:183829 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:5000
          RX bytes:12660021 (12.0 Mb)  TX bytes:16995165 (16.2 Mb)
          Base address:0xe000
```

Figure 18 - Advanced Status Page

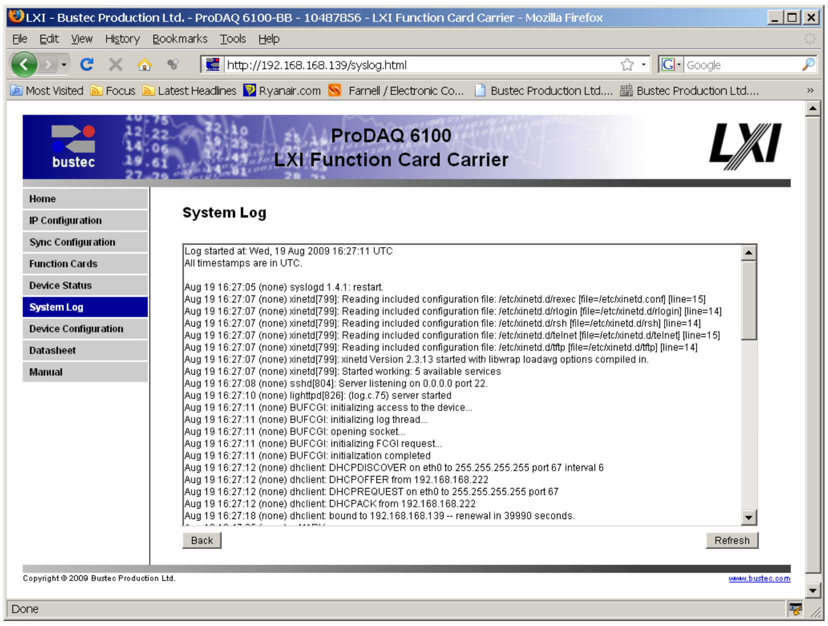
Figure 18 shows an example of the Advanced Status page showing the output of the embedded “ifconfig” command. To select any of the available advanced status information, select one of the items from the list and select the “Refresh” button.

The following advanced status information is currently available:

ifconfig	Displays the output of the embedded “ifconfig” command, showing the settings for the Ethernet interface as seen by the operating system.
route	Displays the output of the embedded “route” command, showing the kernel routing table.
resolv.conf	Shows the contents of the “resolv.conf” file containing the current name server settings.
hosts	Shows the contents of the local “hosts” file.
device.conf	Shows the contents of the device.conf file as created by the embedded web interface. This file stores the information specified by the IP setup page.
device and firmware revision	Shows the current revisions of the different parts of the system.

4.6. System Log

The System Log page shows the contents of the system log. Any housekeeping or debugging information will be entered here by the operating system. To get an up-to-date status, press the "Refresh" button below.


LXI - Bustec Production Ltd. - ProDAQ 6100-BB - 10487856 - LXI Function Card Carrier - Mozilla Firefox
http://192.168.168.139/syslog.html
ProDAQ 6100
LXI Function Card Carrier
Home
IP Configuration
Sync Configuration
Function Cards
Device Status
System Log
Device Configuration
Datasheet
Manual
System Log
Back
Refresh
Copyright © 2009 Bustec Production Ltd.
www.bustec.com
Done

```
Log started at: Wed, 19 Aug 2009 16:27:11 UTC
All timestamps are in UTC.

Aug 19 16:27:05 (none) syslogd 1.4.1: restart.
Aug 19 16:27:07 (none) xinetd[799]: Reading included configuration file: /etc/xinetd.d/rexec [file=/etc/xinetd.conf] [line=15]
Aug 19 16:27:07 (none) xinetd[799]: Reading included configuration file: /etc/xinetd.d/rlogin [file=/etc/xinetd.d/rlogin] [line=14]
Aug 19 16:27:07 (none) xinetd[799]: Reading included configuration file: /etc/xinetd.d/rsh [file=/etc/xinetd.d/rsh] [line=14]
Aug 19 16:27:07 (none) xinetd[799]: Reading included configuration file: /etc/xinetd.d/telnet [file=/etc/xinetd.d/telnet] [line=15]
Aug 19 16:27:07 (none) xinetd[799]: Reading included configuration file: /etc/xinetd.d/tftp [file=/etc/xinetd.d/tftp] [line=14]
Aug 19 16:27:07 (none) xinetd[799]: xinetd Version 2.3.13 started with libwrap loadavg options compiled in.
Aug 19 16:27:07 (none) xinetd[799]: Started working: 5 available services
Aug 19 16:27:08 (none) sshd[804]: Server listening on 0.0.0.0 port 22.
Aug 19 16:27:10 (none) lighttpd[826]: (log.c.75) server started
Aug 19 16:27:11 (none) BUFCGI: initializing access to the device...
Aug 19 16:27:11 (none) BUFCGI: initializing log thread...
Aug 19 16:27:11 (none) BUFCGI: opening socket...
Aug 19 16:27:11 (none) BUFCGI: initializing FCGI request...
Aug 19 16:27:11 (none) BUFCGI: initialization completed
Aug 19 16:27:12 (none) dhclient: DHCPDISCOVER on eth0 to 255.255.255.255 port 67 interval 6
Aug 19 16:27:12 (none) dhclient: DHCPOFFER from 192.168.168.222
Aug 19 16:27:12 (none) dhclient: DHCPREQUEST on eth0 to 255.255.255.255 port 67
Aug 19 16:27:12 (none) dhclient: DHCPACK from 192.168.168.222
Aug 19 16:27:18 (none) dhclient: bound to 192.168.168.139 -- renewal in 39990 seconds.
```

Figure 19 - System Log Page

4.7. Device Configuration

The Device Configuration page allows you to change several parameters of the internal configuration of the device. In addition it allows you to reboot the device or update its firmware.

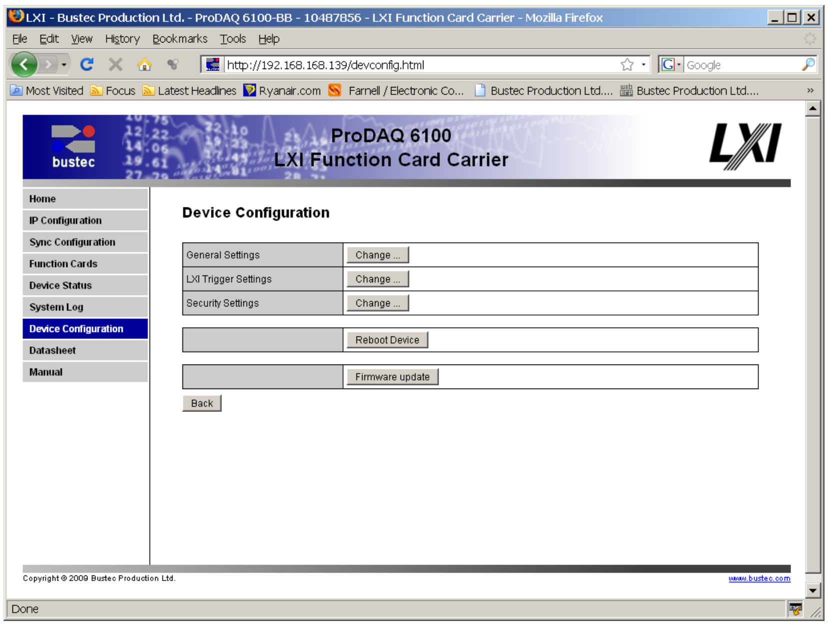
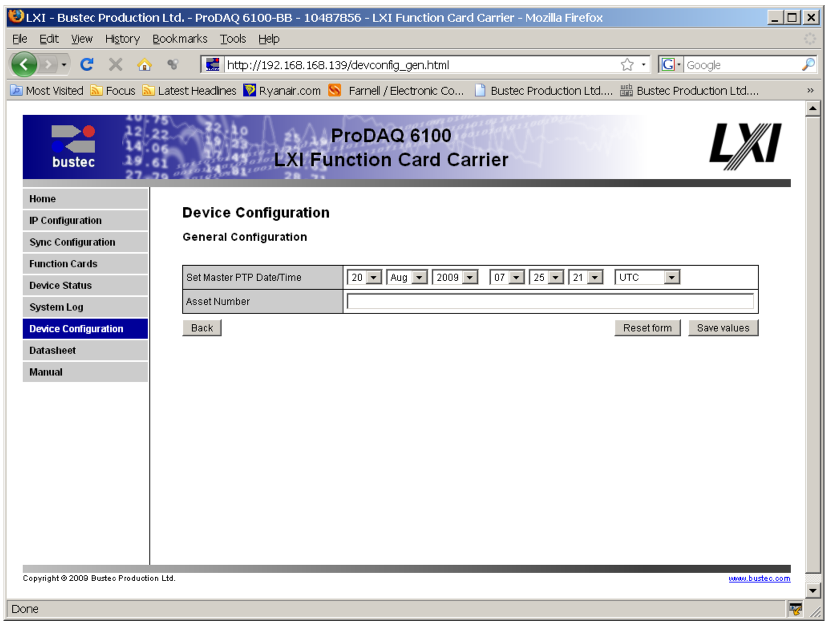


Figure 20 - Device Configuration Page

The device configuration is split up into several sub-items. Click on one of the "Change …" buttons to the right of the different sections to access them. To reboot the device, select the "Reboot Device" button. A reboot my take up to 30 seconds to complete. To update the firmware, select the "Firmware Update" button to access the Firmware Update page.

4.7.1. General Settings

This page allows you to change the system time and assign an asset number to the device, which will be shown on the instrument home page.

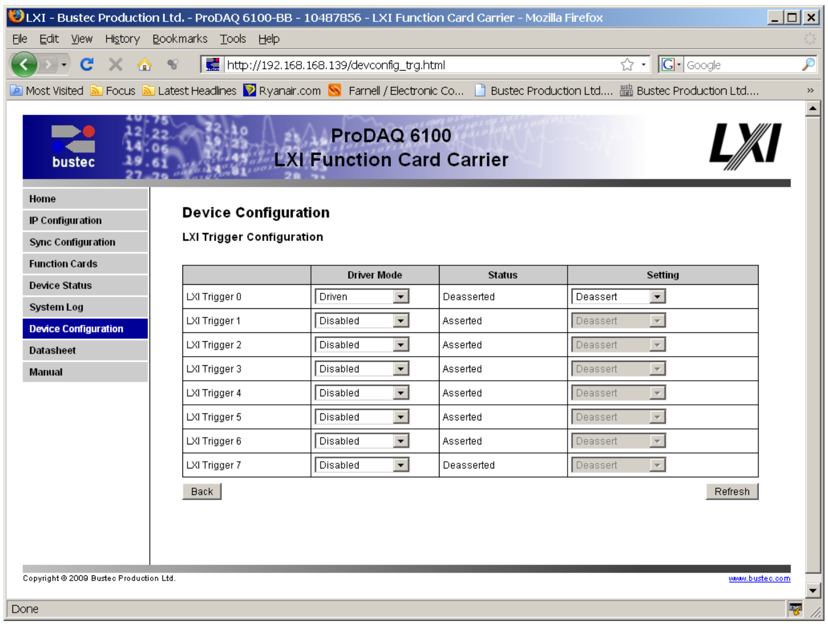

LXI - Bustec Production Ltd. - ProDAQ 6100-BB - 10487856 - LXI Function Card Carrier - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://192.168.168.139/devconfig_gen.html
ProDAQ 6100
LXI Function Card Carrier
Home
IP Configuration
Sync Configuration
Function Cards
Device Status
System Log
Device Configuration
Datasheet
Manual
Device Configuration
General Configuration
Back
Reset form
Save values
Copyright © 2009 Bustec Production Ltd.
www.bustec.com
Done

Set Master PTP Date/Time	20 Aug 2009 07 25 21 UTC
Asset Number	

Figure 21 - General Settings Page

4.7.2. LXI Trigger Settings

This Page allows configuration of the LXI triggers. For each of the 8 triggers it is possible to set a driver mode and, if the driver is enabled, also the state of the trigger line (Asserted or De-asserted). Additionally the current trigger status is displayed.


LXI - Bustec Production Ltd. - ProDAQ 6100-BB - 10487856 - LXI Function Card Carrier - Mozilla Firefox

http://192.168.168.139/devconfig_trg.html

ProDAQ 6100
LXI Function Card Carrier

Device Configuration

LXI Trigger Configuration

	Driver Mode	Status	Setting
LXI Trigger 0	Driven	Deasserted	Deassert
LXI Trigger 1	Disabled	Asserted	Deassert
LXI Trigger 2	Disabled	Asserted	Deassert
LXI Trigger 3	Disabled	Asserted	Deassert
LXI Trigger 4	Disabled	Asserted	Deassert
LXI Trigger 5	Disabled	Asserted	Deassert
LXI Trigger 6	Disabled	Asserted	Deassert
LXI Trigger 7	Disabled	Deasserted	Deassert

Copyright © 2009 Bustec Production Ltd.
File Edit View History Bookmarks Tools Help
Most Visited Focus Latest Headlines Ryanair.com Farnell / Electronic Co... Bustec Production Ltd.... Bustec Production Ltd....
bustec
LXI
Home
IP Configuration
Sync Configuration
Function Cards
Device Status
System Log
Device Configuration
Datasheet
Manual
Back
Refresh
www.bustec.com
Done

Figure 22 - LXI trigger Settings Page

The driver for a trigger line may operate in one of the following modes:

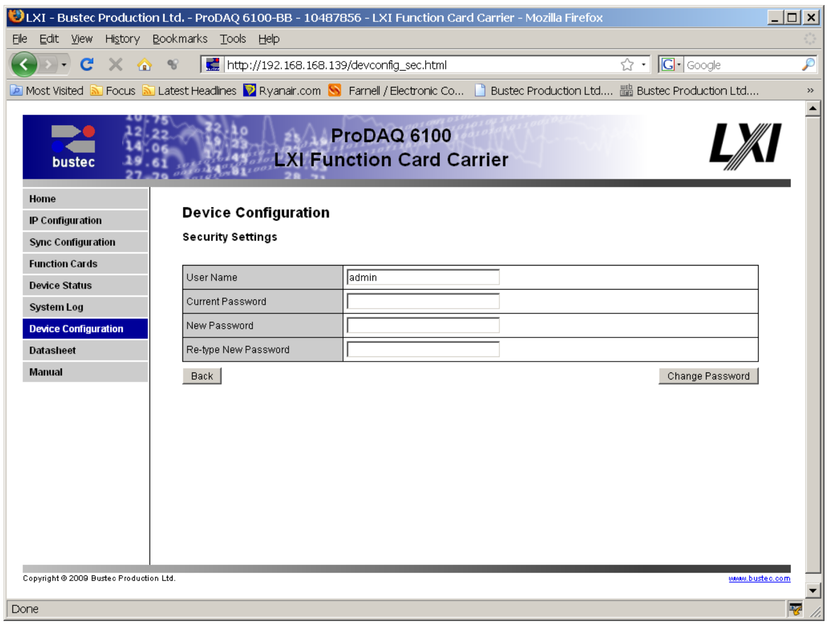
Disabled	LXI devices that are not taking part in trigger operation shall have their drivers disabled.
Driven	This provides point-to-multipoint operation. One device initiates a trigger event to one or more receiving devices. This mode uses one driver per LXI Device for each LXI Trigger Bus channel.
Wired-OR	This is a multipoint-to-multipoint operation. One or more devices initiate a trigger event to one or more receiving devices. In this mode, the event can be initiated by the first device to trigger (first device to recognize an event starts others to perform tasks), or the last device to trigger (last device ready initiates others to perform tasks).
Wired-OR Bias	The Wired-OR Mode requires one device to be configured as the Wired-OR Bias Device to provide a bias for the LXI Trigger Bus channel.

If the driver for a particular trigger line is enabled either as “Driven”, “Wired-OR” or “Wired-OR Bias”, it can be asserted or de-asserted using the control in the “Settings” column of the table.

The “Status” column fields always display the current status of the trigger lines. Select the “Refresh” button below to get an up-to-date status.

4.7.3. Security Settings

On this page you can change the password that is used to protect the pages of the ProDAQ 6100. Please type in your old password, the new one and confirm it by re-typing.


LXI - Bustec Production Ltd. - ProDAQ 6100-BB - 10487856 - LXI Function Card Carrier - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://192.168.168.139/devconfig_sec.html
Most Visited Focus Latest Headlines Ryanair.com Farnell / Electronic Co... Bustec Production Ltd.... Bustec Production Ltd....
bustec
ProDAQ 6100
LXI Function Card Carrier
LXI
Home
IP Configuration
Sync Configuration
Function Cards
Device Status
System Log
Device Configuration
Datasheet
Manual
Back
Change Password
Copyright © 2009 Bustec Production Ltd.
www.bustec.com
Done

Device Configuration

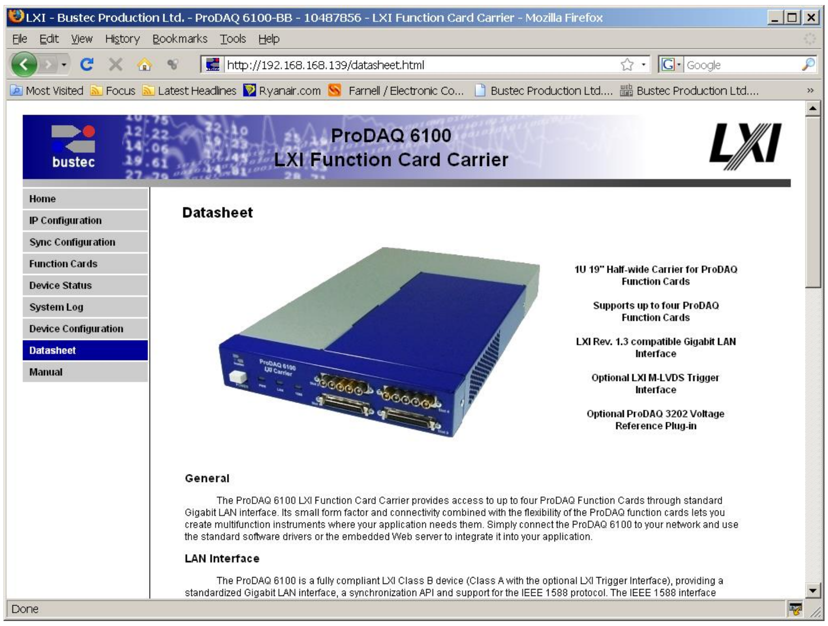
Security Settings

User Name	admin
Current Password	
New Password	
Re-type New Password	

Figure 23 - Security Settings Page

4.8. Datasheet

The datasheet page displays the datasheet for the ProDAQ 6100 LXI Function Card carrier.


LXI - Bustec Production Ltd. - ProDAQ 6100-BB - 10487856 - LXI Function Card Carrier - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://192.168.168.139/datasheet.html
Most Visited Focus Latest Headlines Ryanair.com Farnell / Electronic Co... Bustec Production Ltd.... Bustec Production Ltd....
bustec
ProDAQ 6100
LXI Function Card Carrier
LXI
Home
IP Configuration
Sync Configuration
Function Cards
Device Status
System Log
Device Configuration
Datasheet
Manual
ProDAQ 6100
LXI Carrier
Done

Datasheet

1U 19" Half-wide Carrier for ProDAQ Function Cards

Supports up to four ProDAQ Function Cards

LXI Rev. 1.3 compatible Gigabit LAN Interface

Optional LXI M-LVDS Trigger Interface

Optional ProDAQ 3202 Voltage Reference Plug-in

General

The ProDAQ 6100 LXI Function Card Carrier provides access to up to four ProDAQ Function Cards through standard Gigabit LAN interface. Its small form factor and connectivity combined with the flexibility of the ProDAQ function cards lets you create multifunction instruments where your application needs them. Simply connect the ProDAQ 6100 to your network and use the standard software drivers or the embedded Web server to integrate it into your application.

LAN Interface

The ProDAQ 6100 is a fully compliant LXI Class B device (Class A with the optional LXI Trigger Interface), providing a standardized Gigabit LAN interface, a synchronization API and support for the IEEE 1588 protocol. The IEEE 1588 interface

Figure 24 - Datasheet Page

4.9. Manual

The Manual page displays the manual for the ProDAQ 6100 or allows you to download it onto your computer. You will need to have the Adobe Acrobat plug-in for your browser installed to view the manual in-line.

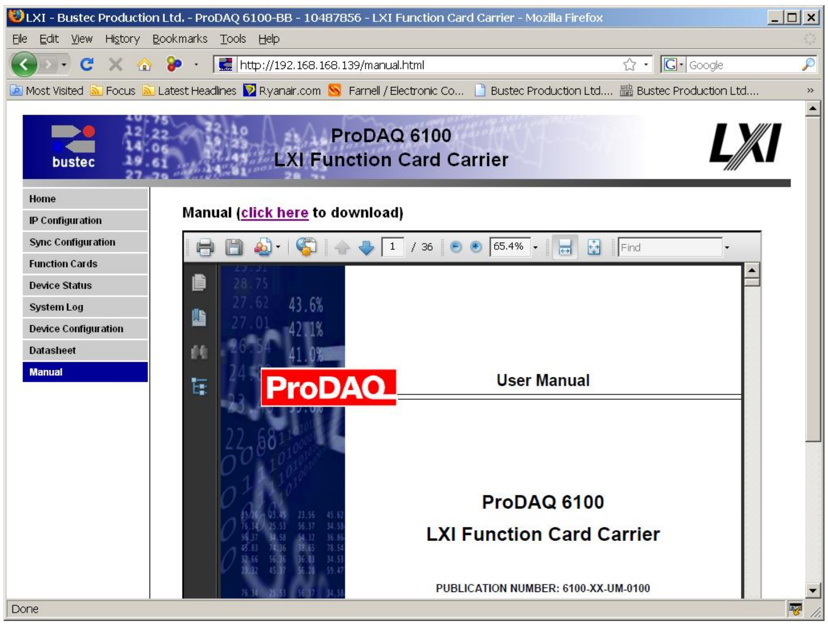

LXI - Bustec Production Ltd. - ProDAQ 6100-BB - 10487856 - LXI Function Card Carrier - Mozilla Firefox
http://192.168.168.139/manual.html
ProDAQ 6100
LXI Function Card Carrier
Home
IP Configuration
Sync Configuration
Function Cards
Device Status
System Log
Device Configuration
Datasheet
Manual
Manual (click here to download)
User Manual
ProDAQ 6100
LXI Function Card Carrier
PUBLICATION NUMBER: 6100-XX-UM-0100
Done

Figure 25 - Manual Page

5. IVI Instrument Driver

The ProDAQ 6100 LXI Function Card Carrier is provided with a IVI-C Specific Driver. The IVI-C driver is written in conformance with IVI-3.1 Driver Architecture Specification (Rev. 2.2) and IVI-3.15 IviLxiSync Specification (Rev 1.0). IVI Class specification version is 2.0. The driver contains functions for opening, configuring, taking measurements from, and closing the instrument.

5.1. Assumptions

To successfully use this device, the following conditions must be met:

- The device must be connected to the LAN over LAN port;
- The device must have proper IP configuration (IP address, Subnet Mask, Default Gateway, etc.);
- The computer, where the IVI driver is used, should have the VISA and IVI Shared components installed and properly configured;

5.2. Error and Status Information

Each function in this instrument driver returns a status code that either indicates success or describes an error or warning condition. Your program should examine the status code returned by each call to an instrument driver function to determine if an error occurred.

The general meaning of the status code is:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

The description of each instrument driver function lists possible error codes and their meanings.

5.3. How to use this manual

Use this document as a programming reference manual. It describes each function for the ProDAQ 6100 LXI Function Card Carrier instrument. The functions appear in alphabetical order, with a description of the function and its C syntax, a description of each parameter, and a list of possible error codes.

5.4. IVI-C Driver Functions

```
Class/Panel Name:                                 Function Name:

   Initialize                                     bu6100_init
   Initialize With Options                        bu6100_InitWithOptions
   Configuration Functions
      Set/Get/Check Attribute
         Set Attribute
            Set Attribute ViInt32                 bu6100_SetAttributeViInt32
            Set Attribute ViReal64                bu6100_SetAttributeViReal64
            Set Attribute ViString                bu6100_SetAttributeViString
            Set Attribute ViBoolean               bu6100_SetAttributeViBoolean
            Set Attribute ViSession               bu6100_SetAttributeViSession
         Get Attribute
```

```
          Get Attribute ViInt32               bu6100_GetAttributeViInt32
          Get Attribute ViReal64              bu6100_GetAttributeViReal64
          Get Attribute ViString              bu6100_GetAttributeViString
          Get Attribute ViBoolean             bu6100_GetAttributeViBoolean
          Get Attribute ViSession             bu6100_GetAttributeViSession
       Check Attribute
          Check Attribute ViInt32             bu6100_CheckAttributeViInt32
          Check Attribute ViReal64            bu6100_CheckAttributeViReal64
          Check Attribute ViString            bu6100_CheckAttributeViString
          Check Attribute ViBoolean           bu6100_CheckAttributeViBoolean
          Check Attribute ViSession           bu6100_CheckAttributeViSession
    Function Card Functions
       Select Function Card CCLK              bu6100_setFcCclk
       Get Function Card CCLK                 bu6100_getFcCclk
    Triggers and Interrupts Functions
       Configure Trigger Lines                bu6100_configTrig
       Install Interrupt Handler              bu6100_installIrqHandler
       Install Interrupt Watcher              bu6100_installIrqWatcher
       Remove Interrupt Service               bu6100_removeIrqService
       Configure 1588 PPP                     bu6100_config1588ppp
    Voltage Refernce Functions
       Get Voltage Reference Info             bu6100_getVoltRefInfo
    1588 Functions
       Get 1588 Configuration                 bu6100_get1588config
       Set 1588 Configuration                 bu6100_set1588config
    DRAM Functions
       Allocate DRAM segment                  bu6100_allocDram
       Free DRAM segment                      bu6100_freeDram
    Circular Buffer Functions
       Set CB Configuration                   bu6100_setConfigCB
       Get CB Configuration                   bu6100_getConfigCB
 LxiSync Functions
    Arm
       Alarm
          Add Arm Alarm                       bu6100_IviLxiSync_AddArmAlarm
          Configure Arm Alarm                 bu6100_IviLxiSync_ConfigureArmAlarm
          Disable All Arm Alarms              bu6100_IviLxiSync_DisableAllArmAlarms
          Get Arm Alarm Name                  bu6100_IviLxiSync_GetArmAlarmName
          Remove All Custom Arm Alarms        bu6100_IviLxiSync_RemoveAllCustomArmAlarms
          Remove Arm Alarm                    bu6100_IviLxiSync_RemoveArmAlarm
       Source
          Add Arm Source                      bu6100_IviLxiSync_AddArmSource
          Configure Arm Source                bu6100_IviLxiSync_ConfigureArmSource
          Disable All Arm Sources             bu6100_IviLxiSync_DisableAllArmSources
          Get Arm Source Name                 bu6100_IviLxiSync_GetArmSourceName
          Remove All Custom Arm Sources       bu6100_IviLxiSync_RemoveAllCustomArmSources
          Remove Arm Source                   bu6100_IviLxiSync_RemoveArmSource
    Event
       Add Event                              bu6100_IviLxiSync_AddEvent
       Configure Event                        bu6100_IviLxiSync_ConfigureEvent
       Disable All Events                     bu6100_IviLxiSync_DisableAllEvents
       Get Event Name                         bu6100_IviLxiSync_GetEventName
       Remove All Custom Events               bu6100_IviLxiSync_RemoveAllCustomEvents
       Remove Event                           bu6100_IviLxiSync_RemoveEvent
    Trigger
       Alarm
          Add Trigger Alarm                   bu6100_IviLxiSync_AddTriggerAlarm
          Configure Trigger Alarm             bu6100_IviLxiSync_ConfigureTriggerAlarm
          Disable All Trigger Alarms          bu6100_IviLxiSync_DisableAllTriggerAlarms
          Get Trigger Alarm Name              bu6100_IviLxiSync_GetTriggerAlarmName
          Remove All Trigger Alarms           bu6100_IviLxiSync_RemoveAllTriggerAlarms
          Remove Trigger Alarm                bu6100_IviLxiSync_RemoveTriggerAlarm
       Source
          Add Trigger Source                  bu6100_IviLxiSync_AddTriggerSource
          Configure Trigger Source            bu6100_IviLxiSync_ConfigureTriggerSource
          Get Trigger Source Name             bu6100_IviLxiSync_GetTriggerSourceName
          Remove All Custom Trigger Sources   bu6100_IviLxiSync_RemoveAllCustomTriggerSources
          Remove Trigger Source               bu6100_IviLxiSync_RemoveTriggerSource
       Arm Trigger                            bu6100_IviLxiSync_ArmTrigger
 Event Log
    Clear Event Log                           bu6100_IviLxiSync_ClearEventLog
    Get Next Event Log Entry                  bu6100_IviLxiSync_GetNextEventLogEntry
    Configure Trigger Log                     bu6100_IviLxiSync_ConfigureTriggerLog
    Clear Trigger Log                         bu6100_IviLxiSync_ClearTriggerLog
    Get Number Of Trigger Log Entries         bu6100_IviLxiSync_GetNumberOfTriggerLogEntries
    Get Next Trigger Log Entry                bu6100_IviLxiSync_GetNextTriggerLogEntry
```

```
Time
   Get System Time                            bu6100_IviLxiSync_GetSystemTime
   Set System Time                            bu6100_IviLxiSync_SetSystemTime
Action/Status Functions
   Function Card Functions
      Reset Function Card                     bu6100_fcReset
   Triggers and Interrupts Functions
      Pulse Trigger                           bu6100_pulseTrig
      Assert 1588 Trigger                     bu6100_assert1588trig
      Pulse Triggers Synchronously            bu6100_pulseTrigSynch
      Wait for Interrupt Watcher              bu6100_waitIrqWatcher
      Read Trigger Status                     bu6100_getTrigStatus
      Read Signal Lines Status                bu6100_readSigLinesStat
   Voltage Reference Functions
      Set Voltage Reference Output            bu6100_setVoltRefOutput
      Get Voltage Reference Output            bu6100_getVoltRefOutput
      Read Volt. Ref. Temperature             bu6100_readVRTemp
   1588 Functions
      Get 1588 Status                         bu6100_get1588status
      Get 1588 Timing                         bu6100_get1588timing
   List Control
      Load List                               bu6100_loadList
      Start List                              bu6100_startList
      Kill List                               bu6100_killList
      Wait List                               bu6100_waitList
      Synchronize List Variables              bu6100_synchronizeListVars
   Circular Buffer Functions
      Get CB Status                           bu6100_getCBstatus
Data Functions
   Function Card Access Functions
      Read from Function Card                 bu6100_fcRead
      Read32 from Function Cards              bu6100_fcRead32
      Write to Function Card                  bu6100_fcWrite
      Write32 to Function Card                bu6100_fcWrite32
      Broadcast Write to Func. Cards          bu6100_fcWriteBcast
      Read Block from Function Card           bu6100_fcReadBlock
      Write Block to Function Card            bu6100_fcWriteBlock
      Read Blck32 from Function Cards         bu6100_fcReadBlock32
      Write Block32 to Function Cards         bu6100_fcWriteBlock32
      Read Block64 from Function Card         bu6100_fcReadBlock64
      Write Block64 to Function Cards         bu6100_fcWriteBlock64
   DRAM Functions
      Read DRAM                               bu6100_readDram
      Write DRAM                              bu6100_writeDram
   Circular Buffer Functions
      Read From CB                            bu6100_readCB
      Write To CB                             bu6100_writeCB
Utility Functions
   Reset                                      bu6100_reset
   Reset With Defaults                        bu6100_ResetWithDefaults
   Disable                                    bu6100_Disable
   Self-Test                                  bu6100_self_test
   Revision Query                             bu6100_revision_query
   Error-Query                                bu6100_error_query
   Error Message                              bu6100_error_message
   Invalidate All Attributes                  bu6100_InvalidateAllAttributes
   Error Info
      Get Error                               bu6100_GetError
      Clear Error                             bu6100_ClearError
   Coercion Info
      Get Next Coercion Record                bu6100_GetNextCoercionRecord
   Interchangeability Info
      Get Next Interchange Warning            bu6100_GetNextInterchangeWarning
      Clear Interchange Warnings              bu6100_ClearInterchangeWarnings
      Reset Interchange Check                 bu6100_ResetInterchangeCheck
   Locking
      Lock Session                            bu6100_LockSession
      Unlock Session                          bu6100_UnlockSession
   Instrument I/O
      Write Instrument Data                   bu6100_WriteInstrData
      Read Instrument Data                    bu6100_ReadInstrData
   Identify Instrument                        bu6100_identify
   Read Board Temperature                     bu6100_readBoardTemp
Close                                         bu6100_close
```

5.5. IVI-C Driver Function Details

5.5.1. bu6100_allocDram

```
    ViStatus bu6100_allocDram (ViSession instrumentHandle, ViInt32 size,
                               ViPInt32 offset);

Purpose

    Allocates the segment of DRAM in the ProDAQ 6100 module.

Parameter List

    instrumentHandle

        Variable Type       ViSession

        The Instrument Handle is used to identify the unique session or
        communication channel between the driver and the instrument.

        If more than one instrument of the same model type is used, this
        Handle will be used to differentiate between them.


    size

        Variable Type       ViInt32

        Specifies the number of32-bit words to be allocated in the 6100
        Module DRAM.

    offset

        Variable Type       ViInt32 (passed by reference)

        Returns the offset of the allocated DRAM memory segment. This address
        can be used directly in functions bu6100_readDram() or
        bu6100_writeDram as an "Offset" parameter.

Return Value

        Returns the status code of this operation.  The status code either
        indicates success or describes an error or warning condition. You
        examine the status code from each call to an instrument driver
        function to determine if an error occurred.

        To obtain a text description of the status code, call the
        bu6100_error_message function.  To obtain additional information
        about the error condition, call the bu6100_GetError function.  To
        clear the error information from the driver, call the
        bu6100_ClearError function.

        The general meaning of the status code is as follows:

        Value                  Meaning
        -------------------------------
        0                      Success
        Positive Values        Warnings
        Negative Values        Errors

        This instrument driver returns errors and warnings defined by other
        sources.  The following table defines the ranges of additional status
        codes that this driver can return.  The table lists the different
        include files that contain the defined constants for the particular
        status codes:

        Numeric Range (in Hex)   Status Code Types
        -------------------------------------------------
        3FFA0000 to 3FFA1FFF     IVI      Warnings
        3FFF0000 to 3FFFFFFF     VISA     Warnings
        3FFC0000 to 3FFCFFFF     VXIPnP   Driver Warnings

        BFFA0000 to BFFA1FFF     IVI      Errors
```

```
BFFF0000 to BFFFFFFF     VISA     Errors
BFFC0000 to BFFCFFFF     VXIPnP   Driver Errors
```

5.5.2. bu6100_assert1588trig

```
    ViStatus bu6100_assert1588trig (ViSession instrumentHandle,
                                    ViReal64 timeSeconds,
                                    ViReal64 timeFractional);

Purpose

    This function configures the 1588 trigger to be asseted at the specified
    1588 time

Parameter List

    instrumentHandle

        Variable Type       ViSession

        The Instrument Handle is used to identify the unique session or
        communication channel between the driver and the instrument.

        If more than one instrument of the same model type is used, this
        Handle will be used to differentiate between them.


    timeSeconds

        Variable Type       ViReal64

        Specifies the seconds portion of the 1588 time when the trigger
        should be asserted.

    timeFractional

        Variable Type       ViReal64

        Specifies the fractional portion of the 1588 time when the trigger
        should be asserted.

Return Value

        Returns the status code of this operation.  The status code  either
        indicates success or describes an error or warning condition.  You
        examine the status code from each call to an instrument driver
        function to determine if an error occurred.

        To obtain a text description of the status code, call the
        bu6100_error_message function.  To obtain additional information
        about the error condition, call the bu6100_GetError function.  To
        clear the error information from the driver, call the
        bu6100_ClearError function.

        The general meaning of the status code is as follows:

        Value                  Meaning
        -------------------------------
        0                      Success
        Positive Values        Warnings
        Negative Values        Errors

        This instrument driver returns errors and warnings defined by other
        sources.  The following table defines the ranges of additional status
        codes that this driver can return.  The table lists the different
        include files that contain the defined constants for the particular
        status codes:

        Numeric Range (in Hex)   Status Code Types
        -------------------------------------------------
        3FFA0000 to 3FFA1FFF     IVI      Warnings
        3FFF0000 to 3FFFFFFF     VISA     Warnings
        3FFC0000 to 3FFCFFFF     VXIPnP   Driver Warnings

        BFFA0000 to BFFA1FFF     IVI      Errors
        BFFF0000 to BFFFFFFF     VISA     Errors
        BFFC0000 to BFFCFFFF     VXIPnP   Driver Errors
```

5.5.3. bu6100_CheckAttributeViBoolean

```
    ViStatus bu6100_CheckAttributeViBoolean (ViSession instrumentHandle,
                                             ViChar _VI_FAR channelName[],
                                             ViAttr attributeID,
                                             ViBoolean attributeValue);

Purpose

    This function checks the validity of a value you specify for a ViBoolean
    attribute.

Parameter List

    instrumentHandle

        Variable Type       ViSession

        The ViSession handle that you obtain from the bu6100_init or
        bu6100_InitWithOptions function.  The handle identifies a particular
        instrument session.

        Default Value:  None


    channelName

        Variable Type       ViChar[]

        If the attribute is channel-based, this parameter specifies the name
        of the channel on which to check the attribute value. If the
        attribute is not channel-based, then pass VI_NULL or an empty string.

        Valid Channel Names:  1

        Default Value:  ""


    attributeID

        Variable Type       ViAttr

        Pass the ID of an attribute.

        From the function panel window, you can use this control as follows.

        - Click on the control or press <ENTER>, <spacebar>, or
          <ctrl-down arrow>, to display a dialog box containing a
          hierarchical list of the available attributes.  Attributes
          whose value cannot be set are dim.  Help text is shown for
          each attribute.  Select an attribute by double-clicking on it
          or by selecting it and then pressing <ENTER>.

          Read-only attributes appear dim in the list box.  If you
          select a read-only attribute, an error message appears.

          A ring control at the top of the dialog box allows you to see
          all IVI attributes or only the attributes of the ViBoolean
          type.  If you choose to see all IVI attributes, the data types
          appear to the right of the attribute names in the list box.
          Attributes with data types other than ViBoolean are dim. If
          you select an attribute data type that is dim, LabWindows/CVI
          transfers you to the function panel for the corresponding
          function that is consistent with the data type.

        - If you want to enter a variable name, press <CTRL-T> to change
          this ring control to a manual input box.

        - If the attribute in this ring control has named constants as
          valid values, you can view the constants by moving to the
          Attribute Value control and pressing <ENTER>.


    attributeValue
```

```
        Variable Type       ViBoolean

        Pass the value which you want to verify as a valid value for the
        attribute.

        From the function panel window, you can use this control as follows.

        - If the attribute currently showing in the Attribute ID ring
          control has constants as valid values, you can view a list of
          the constants by pressing <ENTER> on this control.  Select a
          value by double-clicking on it or by selecting it and then
          pressing <ENTER>.

          Note:  Some of the values might not be valid depending on the
          current settings of the instrument session.

        Default Value: none

Return Value

        Returns the status code of this operation.  The status code  either
        indicates success or describes an error or warning condition.  You
        examine the status code from each call to an instrument driver
        function to determine if an error occurred.

        To obtain a text description of the status code, call the
        bu6100_error_message function.  To obtain additional information
        about the error condition, call the bu6100_GetError function.  To
        clear the error information from the driver, call the
        bu6100_ClearError function.

        The general meaning of the status code is as follows:

        Value                  Meaning
        -------------------------------
        0                      Success
        Positive Values        Warnings
        Negative Values        Errors

        This instrument driver returns errors and warnings defined by other
        sources.  The following table defines the ranges of additional status
        codes that this driver can return.  The table lists the different
        include files that contain the defined constants for the particular
        status codes:

        Numeric Range (in Hex)   Status Code Types
        -------------------------------------------------
        3FFA0000 to 3FFA1FFF     IVI      Warnings
        3FFF0000 to 3FFFFFFF     VISA     Warnings
        3FFC0000 to 3FFCFFFF     VXIPnP   Driver Warnings

        BFFA0000 to BFFA1FFF     IVI      Errors
        BFFF0000 to BFFFFFFF     VISA     Errors
        BFFC0000 to BFFCFFFF     VXIPnP   Driver Errors
```

5.5.4. bu6100_CheckAttributeViInt32

```
    ViStatus bu6100_CheckAttributeViInt32 (ViSession instrumentHandle,
                                           ViChar _VI_FAR channelName[],
                                           ViAttr attributeID,
                                           ViInt32 attributeValue);

Purpose

    This function checks the validity of a value you specify for a ViInt32
    attribute.

Parameter List

    instrumentHandle

        Variable Type       ViSession

        The ViSession handle that you obtain from the bu6100_init or
        bu6100_InitWithOptions function.  The handle identifies a particular
        instrument session.

        Default Value:  None


    channelName

        Variable Type       ViChar[]

        If the attribute is channel-based, this parameter specifies the name
        of the channel on which to check the attribute value. If the
        attribute is not channel-based, then pass VI_NULL or an empty string.

        Valid Channel Names:  1

        Default Value:  ""


    attributeID

        Variable Type       ViAttr

        Pass the ID of an attribute.

        From the function panel window, you can use this control as follows.

        - Click on the control or press <ENTER>, <spacebar>, or
          <ctrl-down arrow>, to display a dialog box containing a
          hierarchical list of the available attributes.  Attributes
          whose value cannot be set are dim.  Help text is shown for
          each attribute.  Select an attribute by double-clicking on it
          or by selecting it and then pressing <ENTER>.

          Read-only attributes appear dim in the list box.  If you
          select a read-only attribute, an error message appears.

          A ring control at the top of the dialog box allows you to see
          all IVI attributes or only the attributes of the ViInt32 type.
          If you choose to see all IVI attributes, the data types appear
          to the right of the attribute names in the list box.
          Attributes with data types other than ViInt32 are dim. If
          you select an attribute data type that is dim, LabWindows/CVI
          transfers you to the function panel for the corresponding
          function that is consistent with the data type.

        - If you want to enter a variable name, press <CTRL-T> to change
          this ring control to a manual input box.

        - If the attribute in this ring control has named constants as
          valid values, you can view the constants by moving to the
          Attribute Value control and pressing <ENTER>.


    attributeValue
```

```
        Variable Type       ViInt32

        Pass the value which you want to verify as a valid value for the
        attribute.

        From the function panel window, you can use this control as follows.

        - If the attribute currently showing in the Attribute ID ring
          control has constants as valid values, you can view a list of
          the constants by pressing <ENTER> on this control.  Select a
          value by double-clicking on it or by selecting it and then
          pressing <ENTER>.

          Note:  Some of the values might not be valid depending on the
          current settings of the instrument session.

        Default Value: none

Return Value

        Returns the status code of this operation.  The status code  either
        indicates success or describes an error or warning condition.  You
        examine the status code from each call to an instrument driver
        function to determine if an error occurred.

        To obtain a text description of the status code, call the
        bu6100_error_message function.  To obtain additional information
        about the error condition, call the bu6100_GetError function.  To
        clear the error information from the driver, call the
        bu6100_ClearError function.

        The general meaning of the status code is as follows:

        Value                  Meaning
        -------------------------------
        0                      Success
        Positive Values        Warnings
        Negative Values        Errors

        This instrument driver returns errors and warnings defined by other
        sources.  The following table defines the ranges of additional status
        codes that this driver can return.  The table lists the different
        include files that contain the defined constants for the particular
        status codes:

        Numeric Range (in Hex)   Status Code Types
        -------------------------------------------------
        3FFA0000 to 3FFA1FFF     IVI      Warnings
        3FFF0000 to 3FFFFFFF     VISA     Warnings
        3FFC0000 to 3FFCFFFF     VXIPnP   Driver Warnings

        BFFA0000 to BFFA1FFF     IVI      Errors
        BFFF0000 to BFFFFFFF     VISA     Errors
        BFFC0000 to BFFCFFFF     VXIPnP   Driver Errors
```

5.5.5.

bu6100_CheckAttributeViReal64

```
    ViStatus bu6100_CheckAttributeViReal64 (ViSession instrumentHandle,
                                            ViChar _VI_FAR channelName[],
                                            ViAttr attributeID,
                                            ViReal64 attributeValue);

Purpose

    This function checks the validity of a value you specify for a ViReal64
    attribute.

Parameter List

    instrumentHandle

        Variable Type       ViSession

        The ViSession handle that you obtain from the bu6100_init or
        bu6100_InitWithOptions function.  The handle identifies a particular
        instrument session.

        Default Value:  None


    channelName

        Variable Type       ViChar[]

        If the attribute is channel-based, this parameter specifies the name
        of the channel on which to check the attribute value. If the
        attribute is not channel-based, then pass VI_NULL or an empty string.

        Valid Channel Names:  1

        Default Value:  ""


    attributeID

        Variable Type       ViAttr

        Pass the ID of an attribute.

        From the function panel window, you can use this control as follows.

        - Click on the control or press <ENTER>, <spacebar>, or
          <ctrl-down arrow>, to display a dialog box containing a
          hierarchical list of the available attributes.  Attributes
          whose value cannot be set are dim.  Help text is shown for
          each attribute.  Select an attribute by double-clicking on it
          or by selecting it and then pressing <ENTER>.

          Read-only attributes appear dim in the list box.  If you
          select a read-only attribute, an error message appears.

          A ring control at the top of the dialog box allows you to see
          all IVI attributes or only the attributes of the ViReal64
          type.  If you choose to see all IVI attributes, the data types
          appear to the right of the attribute names in the list box.
          Attributes with data types other than ViReal64 are dim. If
          you select an attribute data type that is dim, LabWindows/CVI
          transfers you to the function panel for the corresponding
          function that is consistent with the data type.

        - If you want to enter a variable name, press <CTRL-T> to change
          this ring control to a manual input box.

        - If the attribute in this ring control has named constants as
          valid values, you can view the constants by moving to the
          Attribute Value control and pressing <ENTER>.


    attributeValue
```

```
        Variable Type       ViReal64

        Pass the value which you want to verify as a valid value for the
        attribute.

        From the function panel window, you can use this control as follows.

        - If the attribute currently showing in the Attribute ID ring
          control has constants as valid values, you can view a list of
          the constants by pressing <ENTER> on this control.  Select a
          value by double-clicking on it or by selecting it and then
          pressing <ENTER>.

          Note:  Some of the values might not be valid depending on the
          current settings of the instrument session.

        Default Value: none

Return Value

        Returns the status code of this operation.  The status code  either
        indicates success or describes an error or warning condition.  You
        examine the status code from each call to an instrument driver
        function to determine if an error occurred.

        To obtain a text description of the status code, call the
        bu6100_error_message function.  To obtain additional information
        about the error condition, call the bu6100_GetError function.  To
        clear the error information from the driver, call the
        bu6100_ClearError function.

        The general meaning of the status code is as follows:

        Value                  Meaning
        -------------------------------
        0                      Success
        Positive Values        Warnings
        Negative Values        Errors

        This instrument driver returns errors and warnings defined by other
        sources.  The following table defines the ranges of additional status
        codes that this driver can return.  The table lists the different
        include files that contain the defined constants for the particular
        status codes:

        Numeric Range (in Hex)   Status Code Types
        -------------------------------------------------
        3FFA0000 to 3FFA1FFF     IVI      Warnings
        3FFF0000 to 3FFFFFFF     VISA     Warnings
        3FFC0000 to 3FFCFFFF     VXIPnP   Driver Warnings

        BFFA0000 to BFFA1FFF     IVI      Errors
        BFFF0000 to BFFFFFFF     VISA     Errors
        BFFC0000 to BFFCFFFF     VXIPnP   Driver Errors
```

5.5.6. bu6100_CheckAttributeViSession

```
    ViStatus bu6100_CheckAttributeViSession (ViSession instrumentHandle,
                                             ViChar _VI_FAR channelName[],
                                             ViAttr attributeID,
                                             ViSession attributeValue);

Purpose

    This function checks the validity of a value you specify for a ViSession
    attribute.

Parameter List

    instrumentHandle

        Variable Type       ViSession

        The ViSession handle that you obtain from the bu6100_init or
        bu6100_InitWithOptions function.  The handle identifies a particular
        instrument session.

        Default Value:  None


    channelName

        Variable Type       ViChar[]

        If the attribute is channel-based, this parameter specifies the name
        of the channel on which to check the attribute value. If the
        attribute is not channel-based, then pass VI_NULL or an empty string.

        Valid Channel Names:  1

        Default Value:  ""


    attributeID

        Variable Type       ViAttr

        Pass the ID of an attribute.

        From the function panel window, you can use this control as follows.

        - Click on the control or press <ENTER>, <spacebar>, or
          <ctrl-down arrow>, to display a dialog box containing a
          hierarchical list of the available attributes.  Attributes
          whose value cannot be set are dim.  Help text is shown for
          each attribute.  Select an attribute by double-clicking on it
          or by selecting it and then pressing <ENTER>.

          Read-only attributes appear dim in the list box.  If you
          select a read-only attribute, an error message appears.

          A ring control at the top of the dialog box allows you to see
          all IVI attributes or only the attributes of the ViSession
          type.  If you choose to see all IVI attributes, the data types
          appear to the right of the attribute names in the list box.
          Attributes with data types other than ViSession are dim. If
          you select an attribute data type that is dim, LabWindows/CVI
          transfers you to the function panel for the corresponding
          function that is consistent with the data type.

        - If you want to enter a variable name, press <CTRL-T> to change
          this ring control to a manual input box.

        - If the attribute in this ring control has named constants as
          valid values, you can view the constants by moving to the
          Attribute Value control and pressing <ENTER>.


    attributeValue
```

```
        Variable Type       ViSession

        Pass the value which you want to verify as a valid value for the
        attribute.

        From the function panel window, you can use this control as follows.

        - If the attribute currently showing in the Attribute ID ring
          control has constants as valid values, you can view a list of
          the constants by pressing <ENTER> on this control.  Select a
          value by double-clicking on it or by selecting it and then
          pressing <ENTER>.

          Note:  Some of the values might not be valid depending on the
          current settings of the instrument session.

        Default Value: none

  Return Value

        Returns the status code of this operation.  The status code  either
        indicates success or describes an error or warning condition.  You
        examine the status code from each call to an instrument driver
        function to determine if an error occurred.

        To obtain a text description of the status code, call the
        bu6100_error_message function.  To obtain additional information
        about the error condition, call the bu6100_GetError function.  To
        clear the error information from the driver, call the
        bu6100_ClearError function.

        The general meaning of the status code is as follows:

        Value                  Meaning
        -------------------------------
        0                      Success
        Positive Values        Warnings
        Negative Values        Errors

        This instrument driver returns errors and warnings defined by other
        sources.  The following table defines the ranges of additional status
        codes that this driver can return.  The table lists the different
        include files that contain the defined constants for the particular
        status codes:

        Numeric Range (in Hex)   Status Code Types
        -------------------------------------------------
        3FFA0000 to 3FFA1FFF     IVI      Warnings
        3FFF0000 to 3FFFFFFF     VISA     Warnings
        3FFC0000 to 3FFCFFFF     VXIPnP   Driver Warnings

        BFFA0000 to BFFA1FFF     IVI      Errors
        BFFF0000 to BFFFFFFF     VISA     Errors
        BFFC0000 to BFFCFFFF     VXIPnP   Driver Errors
```

5.5.7. bu6100_CheckAttributeViString

```
ViStatus bu6100_CheckAttributeViString (ViSession instrumentHandle,
                                        ViChar _VI_FAR channelName[],
                                        ViAttr attributeID,
                                        ViChar _VI_FAR attributeValue[]);
```

```
Purpose

    This function checks the validity of a value you specify for a ViString
    attribute.

Parameter List

    instrumentHandle

        Variable Type       ViSession

        The ViSession handle that you obtain from the bu6100_init or
        bu6100_InitWithOptions function.  The handle identifies a particular
        instrument session.

        Default Value:  None


    channelName

        Variable Type       ViChar[]

        If the attribute is channel-based, this parameter specifies the name
        of the channel on which to check the attribute value. If the
        attribute is not channel-based, then pass VI_NULL or an empty string.

        Valid Channel Names:  1

        Default Value:  ""


    attributeID

        Variable Type       ViAttr

        Pass the ID of an attribute.

        From the function panel window, you can use this control as follows.

        - Click on the control or press <ENTER>, <spacebar>, or
          <ctrl-down arrow>, to display a dialog box containing a
          hierarchical list of the available attributes.  Attributes
          whose value cannot be set are dim.  Help text is shown for
          each attribute.  Select an attribute by double-clicking on it
          or by selecting it and then pressing <ENTER>.

          Read-only attributes appear dim in the list box.  If you
          select a read-only attribute, an error message appears.

          A ring control at the top of the dialog box allows you to see
          all IVI attributes or only the attributes of the ViString
          type.  If you choose to see all IVI attributes, the data types
          appear to the right of the attribute names in the list box.
          Attributes with data types other than ViString are dim. If
          you select an attribute data type that is dim, LabWindows/CVI
          transfers you to the function panel for the corresponding
          function that is consistent with the data type.

        - If you want to enter a variable name, press <CTRL-T> to change
          this ring control to a manual input box.

        - If the attribute in this ring control has named constants as
          valid values, you can view the constants by moving to the
          Attribute Value control and pressing <ENTER>.


    attributeValue
```

Variable Type ViChar[]

Pass the value which you want to verify as a valid value for the attribute.

From the function panel window, you can use this control as follows.

- If the attribute currently showing in the Attribute ID ring control has constants as valid values, you can view a list of the constants by pressing <ENTER> on this control. Select a value by double-clicking on it or by selecting it and then pressing <ENTER>.

  Note: Some of the values might not be valid depending on the current settings of the instrument session.

Default Value: none

Return Value

Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You examine the status code from each call to an instrument driver function to determine if an error occurred.

To obtain a text description of the status code, call the bu6100_error_message function. To obtain additional information about the error condition, call the bu6100_GetError function. To clear the error information from the driver, call the bu6100_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

This instrument driver returns errors and warnings defined by other sources. The following table defines the ranges of additional status codes that this driver can return. The table lists the different include files that contain the defined constants for the particular status codes:

Numeric Range (in Hex)	Status Code Types	
3FFA0000 to 3FFA1FFF	IVI	Warnings
3FFF0000 to 3FFFFFFF	VISA	Warnings
3FFC0000 to 3FFCFFFF	VXIPnP	Driver Warnings
BFFA0000 to BFFA1FFF	IVI	Errors
BFFF0000 to BFFFFFFF	VISA	Errors
BFFC0000 to BFFCFFFF	VXIPnP	Driver Errors

5.5.8. bu6100_ClearError

```
    ViStatus bu6100_ClearError (ViSession instrumentHandle);

Purpose

    This function clears the error code and error description for the IVI
    session. If the user specifies a valid IVI session for the
    instrument_handle parameter, this function clears the error information
    for the session. If the user passes VI_NULL for the Vi parameter, this
    function clears the error information for the current execution thread.
    If the Vi parameter is an invalid session, the function does nothing and
    returns an error.
    The function clears the error code by setting it to VI_SUCCESS.  If the
    error description string is non-NULL, the function de-allocates the error
    description string and sets the address to VI_NULL.

    Maintaining the error information separately for each thread is useful if
    the user does not have a session handle to pass to the bu6100_GetError
    function, which occurs when a call to bu6100_init or
    bu6100_InitWithOptions fails.

Parameter List

    instrumentHandle

        Variable Type       ViSession

        The ViSession handle that you obtain from the bu6100_init or
        bu6100_InitWithOptions function.  The handle identifies a particular
        instrument session.

        Default Value:  None


Return Value

        Returns the status code of this operation.  The status code  either
        indicates success or describes an error or warning condition.  You
        examine the status code from each call to an instrument driver
        function to determine if an error occurred.

        To obtain a text description of the status code, call the
        bu6100_error_message function.  To obtain additional information
        about the error condition, call the bu6100_GetError function.  To
        clear the error information from the driver, call the
        bu6100_ClearError function.

        The general meaning of the status code is as follows:

        Value                  Meaning
        -------------------------------
        0                      Success
        Positive Values        Warnings
        Negative Values        Errors

        This instrument driver returns errors and warnings defined by other
        sources.  The following table defines the ranges of additional status
        codes that this driver can return.  The table lists the different
        include files that contain the defined constants for the particular
        status codes:

        Numeric Range (in Hex)   Status Code Types
        -------------------------------------------------
        3FFA0000 to 3FFA1FFF     IVI      Warnings
        3FFF0000 to 3FFFFFFF     VISA     Warnings
        3FFC0000 to 3FFCFFFF     VXIPnP   Driver Warnings

        BFFA0000 to BFFA1FFF     IVI      Errors
        BFFF0000 to BFFFFFFF     VISA     Errors
        BFFC0000 to BFFCFFFF     VXIPnP   Driver Errors
```

5.5.9. bu6100_ClearInterchangeWarnings

```
    ViStatus bu6100_ClearInterchangeWarnings (ViSession instrumentHandle);

Purpose

    This function clears the list of current interchange warnings.


Parameter List

    instrumentHandle

        Variable Type       ViSession

        The ViSession handle that you obtain from the bu6100_init or
        bu6100_InitWithOptions function.  The handle identifies a particular
        instrument session.

        Default Value:  None


Return Value

        Returns the status code of this operation.  The status code  either
        indicates success or describes an error or warning condition.  You
        examine the status code from each call to an instrument driver
        function to determine if an error occurred.

        To obtain a text description of the status code, call the
        bu6100_error_message function.  To obtain additional information
        about the error condition, call the bu6100_GetError function.  To
        clear the error information from the driver, call the
        bu6100_ClearError function.

        The general meaning of the status code is as follows:

        Value                  Meaning
        -------------------------------
        0                      Success
        Positive Values        Warnings
        Negative Values        Errors

        This instrument driver returns errors and warnings defined by other
        sources.  The following table defines the ranges of additional status
        codes that this driver can return.  The table lists the different
        include files that contain the defined constants for the particular
        status codes:

        Numeric Range (in Hex)   Status Code Types
        -------------------------------------------------
        3FFA0000 to 3FFA1FFF     IVI      Warnings
        3FFF0000 to 3FFFFFFF     VISA     Warnings
        3FFC0000 to 3FFCFFFF     VXIPnP   Driver Warnings

        BFFA0000 to BFFA1FFF     IVI      Errors
        BFFF0000 to BFFFFFFF     VISA     Errors
        BFFC0000 to BFFCFFFF     VXIPnP   Driver Errors
```

5.5.10. bu6100_close

```
    ViStatus bu6100_close (ViSession instrumentHandle);

Purpose

    This function performs the following operations:

    - Closes the instrument I/O session.

    - Destroys the instrument driver session and all of its attributes.

    - Deallocates any memory resources the driver uses.

    Notes:

    (1) You must unlock the session before calling bu6100_close.

    (2) After calling bu6100_close, you cannot use the instrument driver
    again until you call bu6100_init or bu6100_InitWithOptions.



Parameter List

    instrumentHandle

        Variable Type       ViSession

        The ViSession handle that you obtain from the bu6100_init or
        bu6100_InitWithOptions function.  The handle identifies a particular
        instrument session.

        Default Value:  None


Return Value

        Returns the status code of this operation.  The status code  either
        indicates success or describes an error or warning condition.  You
        examine the status code from each call to an instrument driver
        function to determine if an error occurred.

        To obtain a text description of the status code, call the
        bu6100_error_message function.  To obtain additional information
        about the error condition, call the bu6100_GetError function.  To
        clear the error information from the driver, call the
        bu6100_ClearError function.

        The general meaning of the status code is as follows:

        Value                  Meaning
        -------------------------------
        0                      Success
        Positive Values        Warnings
        Negative Values        Errors

        This instrument driver returns errors and warnings defined by other
        sources.  The following table defines the ranges of additional status
        codes that this driver can return.  The table lists the different
        include files that contain the defined constants for the particular
        status codes:

        Numeric Range (in Hex)   Status Code Types
        -------------------------------------------------
        3FFA0000 to 3FFA1FFF     IVI      Warnings
        3FFF0000 to 3FFFFFFF     VISA     Warnings
        3FFC0000 to 3FFCFFFF     VXIPnP   Driver Warnings

        BFFA0000 to BFFA1FFF     IVI      Errors
        BFFF0000 to BFFFFFFF     VISA     Errors
        BFFC0000 to BFFCFFFF     VXIPnP   Driver Errors
```

5.5.11. bu6100_config1588ppp

```
ViStatus bu6100_config1588ppp (ViSession instrumentHandle,
                               ViReal64 startTimeSeconds,
                               ViReal64 startTimeFractional,
                               ViReal64 period);
```

```
Purpose

    This function configures the 1588 pulse-per-period signal. This signal
    will start at the specified time with the specified period.

Parameter List

    instrumentHandle

        Variable Type       ViSession

        The Instrument Handle is used to identify the unique session or
        communication channel between the driver and the instrument.

        If more than one instrument of the same model type is used, this
        Handle will be used to differentiate between them.


    startTimeSeconds

        Variable Type       ViReal64

        Specifies the seconds portion of the 1588 time when the ppp signal
        should be asserted for the first time.
        Default Value: 0.0

    startTimeFractional

        Variable Type       ViReal64

        Specifies the fractional portion of the 1588 time when the ppp signal
        should be asserted for the first time.
        Default Value: 0.0

    period

        Variable Type       ViReal64

        Specifies the period of the ppp signal in seconds.
        Default Value: 500.0E-9 (500 nanoseconds)

Return Value

        Returns the status code of this operation.  The status code  either
        indicates success or describes an error or warning condition.  You
        examine the status code from each call to an instrument driver
        function to determine if an error occurred.

        To obtain a text description of the status code, call the
        bu6100_error_message function.  To obtain additional information
        about the error condition, call the bu6100_GetError function.  To
        clear the error information from the driver, call the
        bu6100_ClearError function.

        The general meaning of the status code is as follows:

        Value                  Meaning
        -------------------------------
        0                      Success
        Positive Values        Warnings
        Negative Values        Errors

        This instrument driver returns errors and warnings defined by other
        sources.  The following table defines the ranges of additional status
        codes that this driver can return.  The table lists the different
        include files that contain the defined constants for the particular
        status codes:
```

```
Numeric Range (in Hex)   Status Code Types
-------------------------------------------------
3FFA0000 to 3FFA1FFF     IVI      Warnings
3FFF0000 to 3FFFFFFF     VISA     Warnings
3FFC0000 to 3FFCFFFF     VXIPnP   Driver Warnings

BFFA0000 to BFFA1FFF     IVI      Errors
BFFF0000 to BFFFFFFF     VISA     Errors
BFFC0000 to BFFCFFFF     VXIPnP   Driver Errors
```

5.5.12. bu6100_configTrig

```
ViStatus bu6100_configTrig (ViSession instrumentHandle,
                            ViInt16 triggerSource,
                            ViInt16 triggerDestination,
                            ViBoolean function);
```

Purpose

Connects/Disconnects two trigger lines specified by the controls Trigger Source and Trigger Output

Parameter List

instrumentHandle

Variable Type	ViSession

The Instrument Handle is used to identify the unique session or communication channel between the driver and the instrument.

If more than one instrument of the same model type is used, this Handle will be used to differentiate between them.

triggerSource

Variable Type	ViInt16

The source of trigger line.

Any source may be chosen for each of the trigger outputs except that if an attempt is made to select a trigger input for its own output then an error will be returned.

Valid Values:

bu3100_FCTrigOutA1	0	FC 1 Trigger Output A
bu3100_FCTrigOutA2	1	FC 2 Trigger Output A
bu3100_FCTrigOutA3	2	FC 3 Trigger Output A
bu3100_FCTrigOutA4	3	FC 4 Trigger Output A
bu3100_FCTrigOutB1	8	FC 1 Trigger Output B
bu3100_FCTrigOutB2	9	FC 2 Trigger Output B
bu3100_FCTrigOutB3	10	FC 3 Trigger Output B
bu3100_FCTrigOutB4	11	FC 4 Trigger Output B
bu3100_LXITrigIn0	16	LXI Trig In 0
bu3100_LXITrigIn1	17	LXI Trig In 1
bu3100_LXITrigIn2	18	LXI Trig In 2
bu3100_LXITrigIn3	19	LXI Trig In 3
bu3100_LXITrigIn4	20	LXI Trig In 4
bu3100_LXITrigIn5	21	LXI Trig In 5
bu3100_LXITrigIn6	22	LXI Trig In 6
bu3100_LXITrigIn7	23	LXI Trig In 7
bu3100_CLK10	26	CLK10 (10MHz)
bu3100_CLK5	27	CLK10/2 (5MHz)
bu3100_CLK2	28	CLK10/5 (2MHz)
bu3100_IEEE1588_PPP	29	IEEE15588 Pulse Per Period Line
bu3100_IEEE1588_TRG	30	IEEE1588 Trigger Line

triggerDestination

Variable Type	ViInt16

Selects the trigger destination.

Valid Values:

bu3100_FCTrigInA1	0	FC 1 Trigger Input A
bu3100_FCTrigInA2	1	FC 2 Trigger Input A
bu3100_FCTrigInA3	2	FC 3 Trigger Input A
bu3100_FCTrigInA4	3	FC 4 Trigger Input A

```
                    bu3100_FCTrigInB1      8   FC 1 Trigger Input B
                    bu3100_FCTrigInB2      9   FC 2 Trigger Input B
                    bu3100_FCTrigInB3      10  FC 3 Trigger Input B
                    bu3100_FCTrigInB4      11  FC 4 Trigger Input B

                    bu3100_LXITrigOut0     16  LXITrig0
                    bu3100_LXITrigOut1     17  LXITrig1
                    bu3100_LXITrigOut2     18  LXITrig2
                    bu3100_LXITrigOut3     19  LXITrig3
                    bu3100_LXITrigOut4     20  LXITrig4
                    bu3100_LXITrigOut5     21  LXITrig5
                    bu3100_LXITrigOut6     22  LXITrig6
                    bu3100_LXITrigOut7     23  LXITrig7


      function

          Variable Type       ViBoolean

          Specifies the operation on selected trigger lines.
          Valid Values:
                 VI_FALSE (Disconnect)
                 VI_TRUE  (Connect)
          Default Value:
                 VI_TRUE (Connect)


  Return Value

          Returns the status code of this operation.  The status code  either
          indicates success or describes an error or warning condition.  You
          examine the status code from each call to an instrument driver
          function to determine if an error occurred.

          To obtain a text description of the status code, call the
          bu6100_error_message function.  To obtain additional information
          about the error condition, call the bu6100_GetError function.  To
          clear the error information from the driver, call the
          bu6100_ClearError function.

          The general meaning of the status code is as follows:

          Value                  Meaning
          -------------------------------
          0                      Success
          Positive Values        Warnings
          Negative Values        Errors

          This instrument driver returns errors and warnings defined by other
          sources.  The following table defines the ranges of additional status
          codes that this driver can return.  The table lists the different
          include files that contain the defined constants for the particular
          status codes:

          Numeric Range (in Hex)   Status Code Types
          -------------------------------------------------
          3FFA0000 to 3FFA1FFF     IVI      Warnings
          3FFF0000 to 3FFFFFFF     VISA     Warnings
          3FFC0000 to 3FFCFFFF     VXIPnP   Driver Warnings

          BFFA0000 to BFFA1FFF     IVI      Errors
          BFFF0000 to BFFFFFFF     VISA     Errors
          BFFC0000 to BFFCFFFF     VXIPnP   Driver Errors
```

5.5.13. bu6100_Disable

```
    ViStatus bu6100_Disable (ViSession instrumentHandle);

Purpose

    This function places the instrument in a quiescent state where it has
    minimal or no impact on the system to which it is connected.

Parameter List

    instrumentHandle

        Variable Type       ViSession

        The ViSession handle that you obtain from the bu6100_init or
        bu6100_InitWithOptions function.  The handle identifies a particular
        instrument session.

        Default Value:  None


Return Value

        Returns the status code of this operation.  The status code  either
        indicates success or describes an error or warning condition.  You
        examine the status code from each call to an instrument driver
        function to determine if an error occurred.

        To obtain a text description of the status code, call the
        bu6100_error_message function.  To obtain additional information
        about the error condition, call the bu6100_GetError function.  To
        clear the error information from the driver, call the
        bu6100_ClearError function.

        The general meaning of the status code is as follows:

        Value                  Meaning
        -------------------------------
        0                      Success
        Positive Values        Warnings
        Negative Values        Errors

        This instrument driver returns errors and warnings defined by other
        sources.  The following table defines the ranges of additional status
        codes that this driver can return.  The table lists the different
        include files that contain the defined constants for the particular
        status codes:

        Numeric Range (in Hex)   Status Code Types
        -------------------------------------------------
        3FFA0000 to 3FFA1FFF     IVI      Warnings
        3FFF0000 to 3FFFFFFF     VISA     Warnings
        3FFC0000 to 3FFCFFFF     VXIPnP   Driver Warnings

        BFFA0000 to BFFA1FFF     IVI      Errors
        BFFF0000 to BFFFFFFF     VISA     Errors
        BFFC0000 to BFFCFFFF     VXIPnP   Driver Errors
```

5.5.14. bu6100_error_message

```
    ViStatus bu6100_error_message (ViSession instrumentHandle,
                                   ViStatus errorCode,
                                   ViChar _VI_FAR errorMessage[]);

Purpose

    This function converts a status code returned by an instrument driver
    function into a user-readable string.

Parameter List

    instrumentHandle

        Variable Type       ViSession

        The ViSession handle that you obtain from the bu6100_init or
        bu6100_InitWithOptions function.  The handle identifies a particular
        instrument session.

        You can pass VI_NULL for this parameter.  This is useful when one of
        the initialize functions fail.

        Default Value:  VI_NULL


    errorCode

        Variable Type       ViStatus

        Pass the Status parameter that is returned from any of the instrument
        driver functions.

        Default Value:  0  (VI_SUCCESS)

    errorMessage

        Variable Type       ViChar[]

        Returns the user-readable message string that corresponds to the
        status code you specify.

        You must pass a ViChar array with at least 256 bytes.


Return Value

        Reports the status of this operation.

        This function can return only three possible status codes:

        Status    Description
        -------------------------------------------------
               0  No error (the call was successful).

        3FFF0085  Unknown status code (warning).

        BFFF000A  Invalid parameter (Error Message buffer is VI_NULL).
```

5.5.15. bu6100_error_query

```
     ViStatus bu6100_error_query (ViSession instrumentHandle,
                                  ViPInt32 errorCode,
                                  ViChar _VI_FAR errorMessage[]);

 Purpose

     This function reads an error code and a message from the instrument's
     error queue.



 Parameter List

     instrumentHandle

         Variable Type       ViSession

         The ViSession handle that you obtain from the bu6100_init or
         bu6100_InitWithOptions function.  The handle identifies a particular
         instrument session.

         Default Value:  None


     errorCode

         Variable Type       ViInt32 (passed by reference)

         Returns the error code read from the instrument's error queue.



     errorMessage

         Variable Type       ViChar[]

         Returns the error message string read from the instrument's error
         message queue.

         You must pass a ViChar array with at least 256 bytes.


 Return Value

         Returns the status code of this operation.  The status code  either
         indicates success or describes an error or warning condition.  You
         examine the status code from each call to an instrument driver
         function to determine if an error occurred.

         To obtain a text description of the status code, call the
         bu6100_error_message function.  To obtain additional information
         about the error condition, call the bu6100_GetError function.  To
         clear the error information from the driver, call the
         bu6100_ClearError function.

         The general meaning of the status code is as follows:

         Value                  Meaning
         -------------------------------
         0                      Success
         Positive Values        Warnings
         Negative Values        Errors

         This instrument driver returns errors and warnings defined by other
         sources.  The following table defines the ranges of additional status
         codes that this driver can return.  The table lists the different
         include files that contain the defined constants for the particular
         status codes:

         Numeric Range (in Hex)   Status Code Types
         -------------------------------------------------
         3FFA0000 to 3FFA1FFF     IVI      Warnings
         3FFF0000 to 3FFFFFFF     VISA     Warnings
```

```
3FFC0000 to 3FFCFFFF     VXIPnP   Driver Warnings

BFFA0000 to BFFA1FFF     IVI      Errors
BFFF0000 to BFFFFFFF     VISA     Errors
BFFC0000 to BFFCFFFF     VXIPnP   Driver Errors
```

5.5.16. bu6100_fcRead

```
ViStatus bu6100_fcRead (ViSession instrumentHandle,ViInt16 functionCard,
                                       ViInt32 offset, ViPInt16 readData);
```

```
Purpose

    Performs a read of single 16-bit word from a Function Card

Parameter List

    instrumentHandle

        Variable Type       ViSession

        The Instrument Handle is used to identify the unique session or
        communication channel between the driver and the instrument.

        If more than one instrument of the same model type is used, this
        Handle will be used to differentiate between them.


    functionCard

        Variable Type       ViInt16

        The function card to access.

        Valid Values: 1, 2, 3, 4

        Default Value: 1

    offset

        Variable Type       ViInt32

        Offset within the address space of the Function Card

        Default Value: 0x0000

    readData

        Variable Type       ViInt16 (passed by reference)

        Value received from the Function Card.

Return Value

        Returns the status code of this operation.  The status code  either
        indicates success or describes an error or warning condition.  You
        examine the status code from each call to an instrument driver
        function to determine if an error occurred.
        To obtain a text description of the status code, call the
        bu6100_error_message function.  To obtain additional information
        about the error condition, call the bu6100_GetError function.  To
        clear the error information from the driver, call the
        bu6100_ClearError function.
        The general meaning of the status code is as follows:

        Value                  Meaning
        -------------------------------
        0                      Success
        Positive Values        Warnings
        Negative Values        Errors

        This instrument driver returns errors and warnings defined by other
        sources.  The following table defines the ranges of additional status
        codes that this driver can return.  The table lists the different
        include files that contain the defined constants for the particular
        status codes:

        Numeric Range (in Hex)   Status Code Types
        -------------------------------------------------
        3FFA0000 to 3FFA1FFF     IVI      Warnings
        3FFF0000 to 3FFFFFFF     VISA     Warnings
```

```
3FFC0000 to 3FFCFFFF     VXIPnP   Driver Warnings

BFFA0000 to BFFA1FFF     IVI      Errors
BFFF0000 to BFFFFFFF     VISA     Errors
BFFC0000 to BFFCFFFF     VXIPnP   Driver Errors
```

5.5.17. bu6100_fcRead32

```
    ViStatus bu6100_fcRead32 (ViSession instrumentHandle,
                              ViInt16 functionCards, ViInt32 offset,
                              ViPInt32 readData);

Purpose

    Performs a read of single 32-bit word from a Function Card

Parameter List

    instrumentHandle

        Variable Type       ViSession

        The Instrument Handle is used to identify the unique session or
        communication channel between the driver and the instrument.

        If more than one instrument of the same model type is used, this
        Handle will be used to differentiate between them.


    functionCards

        Variable Type       ViInt16

        The function cards to access.

        Valid Values:
        FC 1-3    BU6100_FC_1_3   0
        FC 2-4    BU6100_FC_2_4   1
        FC 1-2    BU6100_FC_1_2   4
        FC 3-4    BU6100_FC_3_4   5

        In the case of reading from Function Cards 1-3 or 2-4:
        The data from the Function Cards 1 and 2 will be placed to lower 16
        bit of each 32-bit word of data buffer.
        The data from the Function Cards 3 and 4 will be placed to upper 16
        bit of each 32-bit word of data buffer.

        In the case of reading from Function Cards 1-2 or 3-4:
        The data from the Function Cards 1 and 3 will be placed to lower 16
        bit of each 32-bit word of data buffer.
        The data from the Function Cards 2 and 4 will be placed to upper 16
        bit of each 32-bit word of data buffer.

        Default Value: 0 (BU6100_FC_1_3)


    offset

        Variable Type       ViInt32

        Offset within the address space of the Function Card.

        Default Value: 0x0000

    readData

        Variable Type       ViInt32 (passed by reference)

        Value received from the Function Card.

Return Value

        Returns the status code of this operation.  The status code  either
        indicates success or describes an error or warning condition.  You
        examine the status code from each call to an instrument driver
        function to determine if an error occurred.

        To obtain a text description of the status code, call the
        bu6100_error_message function.  To obtain additional information
        about the error condition, call the bu6100_GetError function.  To
        clear the error information from the driver, call the
```

bu6100_ClearError function.

The general meaning of the status code is as follows:

```
Value                  Meaning
-------------------------------
0                      Success
Positive Values        Warnings
Negative Values        Errors

This instrument driver returns errors and warnings defined by other
sources.  The following table defines the ranges of additional status
codes that this driver can return.  The table lists the different
include files that contain the defined constants for the particular
status codes:

Numeric Range (in Hex)   Status Code Types
-------------------------------------------------
3FFA0000 to 3FFA1FFF     IVI      Warnings
3FFF0000 to 3FFFFFFF     VISA     Warnings
3FFC0000 to 3FFCFFFF     VXIPnP   Driver Warnings

BFFA0000 to BFFA1FFF     IVI      Errors
BFFF0000 to BFFFFFFF     VISA     Errors
BFFC0000 to BFFCFFFF     VXIPnP   Driver Errors
```

5.5.18. bu6100_fcReadBlock

```
ViStatus bu6100_fcReadBlock (ViSession instrumentHandle,
                             ViInt16 functionCard, ViInt32 offset,
                             ViInt32 count, ViInt16 _VI_FAR readData[]);
```

Purpose

Reads a block of 16-bit words from the Function Card.

Parameter List

instrumentHandle

Variable Type	ViSession

The Instrument Handle is used to identify the unique session or communication channel between the driver and the instrument.

If more than one instrument of the same model type is used, this Handle will be used to differentiate between them.

functionCard

Variable Type	ViInt16

The function card to access.

Valid Values: 1, 2, 3, 4

Default Value: 1

offset

Variable Type	ViInt32

Starting address of Function Card Memory / Address of FIFO register within the address space of the Function Card.

Default Value: 0x0000

count

Variable Type	ViInt32

Number of elements of data (16-bit words) to read from the specified Function Card. If count is greater than 1 the function performs block read.

Default Value: 1

readData

Variable Type	ViInt16[]

An array to receive the block of data read from the Function Card.

This array must be declared at least as large as the number of elements to be read from the instrument - failure to do so may result in a system failure.

Return Value

Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You examine the status code from each call to an instrument driver function to determine if an error occurred.

To obtain a text description of the status code, call the bu6100_error_message function. To obtain additional information about the error condition, call the bu6100_GetError function. To clear the error information from the driver, call the bu6100_ClearError function.

The general meaning of the status code is as follows:

```
Value                  Meaning
-------------------------------
0                      Success
Positive Values        Warnings
Negative Values        Errors

This instrument driver returns errors and warnings defined by other
sources.  The following table defines the ranges of additional status
codes that this driver can return.  The table lists the different
include files that contain the defined constants for the particular
status codes:

Numeric Range (in Hex)   Status Code Types
-------------------------------------------------
3FFA0000 to 3FFA1FFF     IVI      Warnings
3FFF0000 to 3FFFFFFF     VISA     Warnings
3FFC0000 to 3FFCFFFF     VXIPnP   Driver Warnings

BFFA0000 to BFFA1FFF     IVI      Errors
BFFF0000 to BFFFFFFF     VISA     Errors
BFFC0000 to BFFCFFFF     VXIPnP   Driver Errors
```

5.5.19. bu6100_fcReadBlock32

```
ViStatus bu6100_fcReadBlock32 (ViSession instrumentHandle,
                               ViInt16 functionCards, ViInt32 offset,
                               ViInt32 count,
                               ViInt32 _VI_FAR readData32[]);
```

```
Purpose

    Reads a block of 32-bit words from two Function Cards.

Parameter List

    instrumentHandle

        Variable Type       ViSession

        The Instrument Handle is used to identify the unique session or
        communication channel between the driver and the instrument.

        If more than one instrument of the same model type is used, this
        Handle will be used to differentiate between them.


    functionCards

        Variable Type       ViInt16

        The function cards to access.

        Valid Values:
        FC 1-3    BU6100_FC_1_3   0
        FC 2-4    BU6100_FC_2_4   1
        FC 1-2    BU6100_FC_1_2   4
        FC 3-4    BU6100_FC_3_4   5

        In the case of reading from Function Cards 1-3 or 2-4:
        The data from the Function Cards 1 and 2 will be placed to lower 16
        bit of each 32-bit word of data buffer.
        The data from the Function Cards 3 and 4 will be placed to upper 16
        bit of each 32-bit word of data buffer.

        In the case of reading from Function Cards 1-2 or 3-4:
        The data from the Function Cards 1 and 3 will be placed to lower 16
        bit of each 32-bit word of data buffer.
        The data from the Function Cards 2 and 4 will be placed to upper 16
        bit of each 32-bit word of data buffer.

        Default Value: 0 (BU6100_FC_1_3)

    offset

        Variable Type       ViInt32

        Starting address of Function Card Memory / Address of FIFO register
        within the address space of the Function Card. The same offset will
        be used for both Function Cards.

        Default Value: 0x0000

    count

        Variable Type       ViInt32

        Number of elements of data to read from the specified Function Cards.

        This number of 16-bit elements will be read from each function card.
        And this number of 32-bit elements will be placed in the read buffer
        upon successive completion of the function call.

        Default Value: 1

    readData32

        Variable Type       ViInt32[]
```

```
An array to read the block of data from the Function Cards.

Lower 16 bit in each 32-bit word in this buffer will contain the data
from the first function card (Function Card 1).
```

Upper 16 bit in each 32-bit word in this buffer will contain the data from the second function card (Function Card 2).

This array must be declared at least as large as the number of elements to be read from the instrument - failure to do so may result in a system failure.

Return Value

Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You examine the status code from each call to an instrument driver function to determine if an error occurred.

To obtain a text description of the status code, call the bu6100_error_message function. To obtain additional information about the error condition, call the bu6100_GetError function. To clear the error information from the driver, call the bu6100_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

This instrument driver returns errors and warnings defined by other sources. The following table defines the ranges of additional status codes that this driver can return. The table lists the different include files that contain the defined constants for the particular status codes:

Numeric Range (in Hex)	Status Code Types	
3FFA0000 to 3FFA1FFF	IVI	Warnings
3FFF0000 to 3FFFFFFF	VISA	Warnings
3FFC0000 to 3FFCFFFF	VXIPnP	Driver Warnings
BFFA0000 to BFFA1FFF	IVI	Errors
BFFF0000 to BFFFFFFF	VISA	Errors
BFFC0000 to BFFCFFFF	VXIPnP	Driver Errors

5.5.20. bu6100_fcReadBlock64

```
ViStatus bu6100_fcReadBlock64 (ViSession instrumentHandle,
                               ViInt32 offset, ViInt32 count,
                               ViInt32 _VI_FAR readData64[]);
```

```
Purpose

    Reads a block of 64-bit words from four Function Cards.
    The data from Function Card 1 will be placed to bits 0-15;
    The data from Function Card 2 will be placed to bits 16-31;
    The data from Function Card 3 will be placed to bits 32-47;
    The data from Function Card 4 will be placed to bits 48-63;

Parameter List

    instrumentHandle

        Variable Type       ViSession

        The Instrument Handle is used to identify the unique session or
        communication channel between the driver and the instrument.

        If more than one instrument of the same model type is used, this
        Handle will be used to differentiate between them.


    offset

        Variable Type       ViInt32

        Starting address of Function Card Memory / Address of FIFO register
        within the address space of the Function Card. The same offset will
        be used for both Function Cards.

        Default Value: 0x0000

    count

        Variable Type       ViInt32

        Number of 64-bit elements of data to read from the specified Function
        Cards.

        Default Value: 1

    readData64

        Variable Type       ViInt32[]

        An array to read the block of data from the Function Cards.

        In the case of reading from the Function Cards 1, 3, 5, 7 or 2, 4, 6,
        8 data will be arranged in the following way:
        FC1, FC2 - First word, lower 16 bit;
        FC3, FC4 - First word, upper 16 bit;
        FC5, FC6 - Second word, lower 16 bit;
        FC7, FC8 - Second word, upper 16 bit;

        In the case of reading from the Function Cards 1, 2, 3, 4 or 5, 6, 7,
        8 data will be placed in the following way:
        FC1, FC5 - First word, lower 16 bit;
        FC2, FC6 - First word, upper 16 bit;
        FC3, FC7 - Second word, lower 16 bit;
        FC4, FC8 - Second word, upper 16 bit;

        This array must be declared at least as large as the double number of
        64-bit elements to be read from the instrument (Count*2) - failure to
        do so may result in a system failure.

Return Value

        Returns the status code of this operation.  The status code  either
        indicates success or describes an error or warning condition.  You
        examine the status code from each call to an instrument driver
        function to determine if an error occurred.
```

To obtain a text description of the status code, call the bu6100_error_message function. To obtain additional information about the error condition, call the bu6100_GetError function. To clear the error information from the driver, call the bu6100_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

This instrument driver returns errors and warnings defined by other sources. The following table defines the ranges of additional status codes that this driver can return. The table lists the different include files that contain the defined constants for the particular status codes:

Numeric Range (in Hex)	Status Code Types	
3FFA0000 to 3FFA1FFF	IVI	Warnings
3FFF0000 to 3FFFFFFF	VISA	Warnings
3FFC0000 to 3FFCFFFF	VXIPnP	Driver Warnings
BFFA0000 to BFFA1FFF	IVI	Errors
BFFF0000 to BFFFFFFF	VISA	Errors
BFFC0000 to BFFCFFFF	VXIPnP	Driver Errors

5.5.21. bu6100_fcReset

```
    ViStatus bu6100_fcReset (ViSession instrumentHandle,
                             ViInt16 functionCardMask, ViInt16 function);

Purpose

    Reset the function cards at the selected locations.


Parameter List

    instrumentHandle

        Variable Type       ViSession

        The Instrument Handle is used to identify the unique session or
        communication channel between the driver and the instrument.

        If more than one instrument of the same model type is used, this
        Handle will be used to differentiate between them.


    functionCardMask

        Variable Type       ViInt16

        Specifies the Function Cards to reset.
        Bit 0  corresponds to FC 1
        ........
        Bit 3  corresponds to FC 4

        "1" written to appropriate bit means that
        function card should be reset.

        Default Value: 0

    function

        Variable Type       ViInt16

        Specifies which action will be taken on Function Card Reset lines.

        Valid Values:

        BU6100_RESET_DEASSERT 0 De-asserts reset lines previously
                                asertted by
                                fcReset(vi, fcMask,bu3100_RESET_ASSERT)
                                function call. After de-asserting reset
                                lines the program should wait at least
                                bu3100_FC_RESET_WAIT seconds before any
                                access to the function cards.

        BU6100_RESET_ASSERT   1 Asserts the reset lines on the selected
                                function cards. The reset lines should
                                be de-asserted later by function call
                                fcReset(vi,fcMask,bu3100_RESET_DEASSERT)
                                The program should wait at least
                                bu3100_FC_RESET_DOWN seconds before
                                de-assert the reset lines.

        BU6100_RESET_PULSE    2 Performs complete reset cycle on the
                                selected function cards: Asserts reset
                                lines, waits bu3100_FC_RESET_DOWN
                                seconds, de-asserts reset lines, and
                                then waits bu3100_FC_RESET_WAIT seconds.

        Default Value: 0 (BU6100_RESET_DEASSERT)

Return Value

        Returns the status code of this operation.  The status code  either
        indicates success or describes an error or warning condition.  You
        examine the status code from each call to an instrument driver
        function to determine if an error occurred.
```

```
To obtain a text description of the status code, call the
bu6100_error_message function.  To obtain additional information
about the error condition, call the bu6100_GetError function.  To
clear the error information from the driver, call the
bu6100_ClearError function.

The general meaning of the status code is as follows:

Value                  Meaning
-------------------------------
0                      Success
Positive Values        Warnings
Negative Values        Errors

This instrument driver returns errors and warnings defined by other
sources.  The following table defines the ranges of additional status
codes that this driver can return.  The table lists the different
include files that contain the defined constants for the particular
status codes:

Numeric Range (in Hex)   Status Code Types
-------------------------------------------------
3FFA0000 to 3FFA1FFF     IVI      Warnings
3FFF0000 to 3FFFFFFF     VISA     Warnings
3FFC0000 to 3FFCFFFF     VXIPnP   Driver Warnings

BFFA0000 to BFFA1FFF     IVI      Errors
BFFF0000 to BFFFFFFF     VISA     Errors
BFFC0000 to BFFCFFFF     VXIPnP   Driver Errors
```

5.5.22. bu6100_fcWrite

```
    ViStatus bu6100_fcWrite (ViSession instrumentHandle,
                             ViInt16 functionCard, ViInt32 offset,
                             ViInt16 writeData);
```

```
Purpose

    Writes a single 16-bit word to the Function Card.

Parameter List

    instrumentHandle

        Variable Type       ViSession

        The Instrument Handle is used to identify the unique session or
        communication channel between the driver and the instrument.

        If more than one instrument of the same model type is used, this
        Handle will be used to differentiate between them.


    functionCard

        Variable Type       ViInt16

        The function card to access.

        Valid Values: 1, 2, 3, 4

        Default Value: 1

    offset

        Variable Type       ViInt32

        Offset within the address space of the Function Card.

        Default Value: 0x0000

    writeData

        Variable Type       ViInt16

        16-bit word of data to be written to the specified function card

Return Value

        Returns the status code of this operation.  The status code  either
        indicates success or describes an error or warning condition.  You
        examine the status code from each call to an instrument driver
        function to determine if an error occurred.

        To obtain a text description of the status code, call the
        bu6100_error_message function.  To obtain additional information
        about the error condition, call the bu6100_GetError function.  To
        clear the error information from the driver, call the
        bu6100_ClearError function.

        The general meaning of the status code is as follows:

        Value                  Meaning
        -------------------------------
        0                      Success
        Positive Values        Warnings
        Negative Values        Errors

        This instrument driver returns errors and warnings defined by other
        sources.  The following table defines the ranges of additional status
        codes that this driver can return.  The table lists the different
        include files that contain the defined constants for the particular
        status codes:
```

```
Numeric Range (in Hex)   Status Code Types
-------------------------------------------------
3FFA0000 to 3FFA1FFF     IVI      Warnings
3FFF0000 to 3FFFFFFF     VISA     Warnings
3FFC0000 to 3FFCFFFF     VXIPnP   Driver Warnings

BFFA0000 to BFFA1FFF     IVI      Errors
BFFF0000 to BFFFFFFF     VISA     Errors
BFFC0000 to BFFCFFFF     VXIPnP   Driver Errors
```

5.5.23. bu6100_fcWrite32

```
ViStatus bu6100_fcWrite32 (ViSession instrumentHandle,
                           ViInt16 functionCards, ViInt32 offset,
                           ViInt32 writeData);
```

Purpose

Writes a single 32-bit word to the Function Card.

Parameter List

instrumentHandle

Variable Type	ViSession

The Instrument Handle is used to identify the unique session or communication channel between the driver and the instrument.

If more than one instrument of the same model type is used, this Handle will be used to differentiate between them.

functionCards

Variable Type	ViInt16

The function cards to access.

Valid Values:

FC 1-3	BU6100_FC_1_3	0
FC 2-4	BU6100_FC_2_4	1
FC 1-2	BU6100_FC_1_2	4
FC 3-4	BU6100_FC_3_4	5

In the case of writing to Function Cards 1-3 or 2-4:
The lower 16 bit of data will be placed to the Function Cards 1 and 2.
The upper 16 bit of data will be placed to the Function Cards 3 and 4.
In the case of writing to Function Cards 1-2 or 3-4:
The lower 16 bit of data will be placed to the Function Cards 1 and 3.
The upper 16 bit of data will be placed to the Function Cards 2 and 4.

Default Value: 0 (BU6100_FC_1_3)

offset

Variable Type	ViInt32

Offset within the address space of the Function Card.

Default Value: 0x0000

writeData

Variable Type	ViInt32

32-bit word of data to be written to the specified function card

Return Value

Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You examine the status code from each call to an instrument driver function to determine if an error occurred.

To obtain a text description of the status code, call the bu6100_error_message function. To obtain additional information about the error condition, call the bu6100_GetError function. To clear the error information from the driver, call the bu6100_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

This instrument driver returns errors and warnings defined by other sources. The following table defines the ranges of additional status codes that this driver can return. The table lists the different include files that contain the defined constants for the particular status codes:

Numeric Range (in Hex)	Status Code Types	
3FFA0000 to 3FFA1FFF	IVI	Warnings
3FFF0000 to 3FFFFFFF	VISA	Warnings
3FFC0000 to 3FFCFFFF	VXIPnP	Driver Warnings
BFFA0000 to BFFA1FFF	IVI	Errors
BFFF0000 to BFFFFFFF	VISA	Errors
BFFC0000 to BFFCFFFF	VXIPnP	Driver Errors

5.5.24. bu6100_fcWriteBcast

```
ViStatus bu6100_fcWriteBcast (ViSession instrumentHandle,
                              ViInt16 functionCardMask, ViInt32 offset,
                              ViInt16 writeData);
```

```
Purpose

    Simultaneously writes a single 16-bit word to multiple Function Cards.

Parameter List

    instrumentHandle

        Variable Type       ViSession

        The Instrument Handle is used to identify the unique session or
        communication channel between the driver and the instrument.

        If more than one instrument of the same model type is used, this
        Handle will be used to differentiate between them.


    functionCardMask

        Variable Type       ViInt16

        Specifies to which Function Cards Data will be written.

        Bit 0 corresponds to Function Card 1
        ...
        Bit 3 corresponds to Function Card 4

        Default Value: 0

    offset

        Variable Type       ViInt32

        Offset within the address space of each Function Card.

        Default Value: 0x0000

    writeData

        Variable Type       ViInt16

        16-bit word of data to be written to all specified Function Cards.

Return Value

        Returns the status code of this operation.  The status code  either
        indicates success or describes an error or warning condition.  You
        examine the status code from each call to an instrument driver
        function to determine if an error occurred.

        To obtain a text description of the status code, call the
        bu6100_error_message function.  To obtain additional information
        about the error condition, call the bu6100_GetError function.  To
        clear the error information from the driver, call the
        bu6100_ClearError function.

        The general meaning of the status code is as follows:

        Value                  Meaning
        -------------------------------
        0                      Success
        Positive Values        Warnings
        Negative Values        Errors

        This instrument driver returns errors and warnings defined by other
        sources.  The following table defines the ranges of additional status
        codes that this driver can return.  The table lists the different
        include files that contain the defined constants for the particular
        status codes:
```

```
Numeric Range (in Hex)   Status Code Types
-------------------------------------------------
3FFA0000 to 3FFA1FFF     IVI      Warnings
3FFF0000 to 3FFFFFFF     VISA     Warnings
3FFC0000 to 3FFCFFFF     VXIPnP   Driver Warnings

BFFA0000 to BFFA1FFF     IVI      Errors
BFFF0000 to BFFFFFFF     VISA     Errors
BFFC0000 to BFFCFFFF     VXIPnP   Driver Errors
```

5.5.25. bu6100_fcWriteBlock

```
ViStatus bu6100_fcWriteBlock (ViSession instrumentHandle,
                              ViInt16 functionCard, ViInt32 offset,
                              ViInt32 count,
                              ViInt16 _VI_FAR writeData[]);
```

Purpose

Writes a block of 16-bit words to the Function Card.

Parameter List

instrumentHandle

Variable Type	ViSession

The Instrument Handle is used to identify the unique session or communication channel between the driver and the instrument.

If more than one instrument of the same model type is used, this Handle will be used to differentiate between them.

functionCard

Variable Type	ViInt16

The function card to access.

Valid Values: 1, 2, 3, 4

Default Value: 1

offset

Variable Type	ViInt32

Starting address of Function Card Memory / Address of FIFO register within the address space of the Function Card.

Default Value: 0x0000

count

Variable Type	ViInt32

Number of elements of data to write to the specified Function Card. If count is greater than 1 the function performs block write.

Default Value: 1

writeData

Variable Type	ViInt16[]

An array to write the block of data read to the Function Card.

This array must be declared at least as large as the number of elements to be written to the instrument - failure to do so may result in a system failure.

Return Value

Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You examine the status code from each call to an instrument driver function to determine if an error occurred.

To obtain a text description of the status code, call the bu6100_error_message function. To obtain additional information about the error condition, call the bu6100_GetError function. To clear the error information from the driver, call the bu6100_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

This instrument driver returns errors and warnings defined by other sources. The following table defines the ranges of additional status codes that this driver can return. The table lists the different include files that contain the defined constants for the particular status codes:

Numeric Range (in Hex)	Status Code Types	
3FFA0000 to 3FFA1FFF	IVI	Warnings
3FFF0000 to 3FFFFFFF	VISA	Warnings
3FFC0000 to 3FFCFFFF	VXIPnP	Driver Warnings
BFFA0000 to BFFA1FFF	IVI	Errors
BFFF0000 to BFFFFFFF	VISA	Errors
BFFC0000 to BFFCFFFF	VXIPnP	Driver Errors

5.5.26. bu6100_fcWriteBlock32

```
ViStatus bu6100_fcWriteBlock32 (ViSession instrumentHandle,
                                ViInt16 functionCards, ViInt32 offset,
                                ViInt32 count,
                                ViInt32 _VI_FAR writeData32[]);
```

```
Purpose

    Writes a block of 32-bit words to two Function Cards.

Parameter List

    instrumentHandle

        Variable Type       ViSession

        The Instrument Handle is used to identify the unique session or
        communication channel between the driver and the instrument.

        If more than one instrument of the same model type is used, this
        Handle will be used to differentiate between them.


    functionCards

        Variable Type       ViInt16

        The function cards to access.

        Valid Values:
        FC 1-3    BU6100_FC_1_3   0
        FC 2-4    BU6100_FC_2_4   1
        FC 1-2    BU6100_FC_1_2   4
        FC 3-4    BU6100_FC_3_4   5

        In the case of writing to Function Cards 1-3 or 2-4:
        The lower 16 bit of data will be placed to the Function Cards 1 and 2.
        The upper 16 bit of data will be placed to the Function Cards 3 and 4.

        In the case of writing to Function Cards 1-2 or 3-4:
        The lower 16 bit of data will be placed to the Function Cards 1 and 3.
        The upper 16 bit of data will be placed to the Function Cards 2 and 4.

        Default Value: 0 (BU6100_FC_1_3)

    offset

        Variable Type       ViInt32

        Starting address of Function Card Memory / Address of FIFO register
        within the address space of the Function Card. The same offset will
        be used for both Function Cards.

        Default Value: 0x0000

    count

        Variable Type       ViInt32

        Number of elements of data to write to the specified Function Cards.
        This control specifies the number of 32-bit elements in the write
        buffer. Each function card will get this number of 16-bit elements.

        Default Value: 1

    writeData32

        Variable Type       ViInt32[]

        An array to write the block of data read to the Function Card.

        Lower 16 bit in each 32-bit word in this buffer should contain the
        data for the first function card (Function Card 1).
        Upper 16 bit in each 32-bit word in this buffer should contain the
        data for the second function card (Function Card 2).
        This array must be declared at least as large as the number of
        elements to be written to the instrument - failure to do so may
        result in a system failure.

  Return Value

        Returns the status code of this operation.  The status code  either
        indicates success or describes an error or warning condition.  You
        examine the status code from each call to an instrument driver
        function to determine if an error occurred.

        To obtain a text description of the status code, call the
        bu6100_error_message function.  To obtain additional information
        about the error condition, call the bu6100_GetError function.  To
        clear the error information from the driver, call the
        bu6100_ClearError function.

        The general meaning of the status code is as follows:

        Value                  Meaning
        -------------------------------
        0                      Success
        Positive Values        Warnings
        Negative Values        Errors

        This instrument driver returns errors and warnings defined by other
        sources.  The following table defines the ranges of additional status
        codes that this driver can return.  The table lists the different
        include files that contain the defined constants for the particular
        status codes:

        Numeric Range (in Hex)   Status Code Types
        -------------------------------------------------
        3FFA0000 to 3FFA1FFF     IVI      Warnings
        3FFF0000 to 3FFFFFFF     VISA     Warnings
        3FFC0000 to 3FFCFFFF     VXIPnP   Driver Warnings

        BFFA0000 to BFFA1FFF     IVI      Errors
        BFFF0000 to BFFFFFFF     VISA     Errors
        BFFC0000 to BFFCFFFF     VXIPnP   Driver Errors
```

5.5.27. bu6100_fcWriteBlock64

```
    ViStatus bu6100_fcWriteBlock64 (ViSession instrumentHandle,
                                    ViInt32 offset, ViInt32 count,
                                    ViInt32 _VI_FAR writeData64[]);

Purpose

    Writes a block of 32-bit words to two Function Cards.
    The data from Function Card 1 will be placed to bits 0-15;
    The data from Function Card 2 will be placed to bits 16-31;
    The data from Function Card 3 will be placed to bits 32-47;
    The data from Function Card 4 will be placed to bits 48-63;

Parameter List

    instrumentHandle

        Variable Type       ViSession

        The Instrument Handle is used to identify the unique session or
        communication channel between the driver and the instrument.

        If more than one instrument of the same model type is used, this
        Handle will be used to differentiate between them.


    offset

        Variable Type       ViInt32

        Starting address of Function Card Memory / Address of FIFO register
        within the address space of the Function Card. The same offset will
        be used for both Function Cards.

        Default Value: 0x0000

    count

        Variable Type       ViInt32

        Number of elements of data to write to the specified Function Cards.
        This control specifies the number of 64-bit elements in the write
        buffer. Each function card will get this number of 16-bit elements.

        Default Value: 1

    writeData64

        Variable Type       ViInt32[]

        An array to write the block of data read to the Function Card.

        In the case of writing to the Function Cards 1, 3, 5, 7 or 2, 4, 6, 8
        data should be arranged in the following way:
        FC1, FC2 - First word, lower 16 bit;
        FC3, FC4 - First word, upper 16 bit;
        FC5, FC6 - Second word, lower 16 bit;
        FC7, FC8 - Second word, upper 16 bit;

        In the case of writing to the Function Cards 1, 2, 3, 4 or 5, 6, 7, 8
        data should be placed in the following way:
        FC1, FC5 - First word, lower 16 bit;
        FC2, FC6 - First word, upper 16 bit;
        FC3, FC7 - Second word, lower 16 bit;
        FC4, FC8 - Second word, upper 16 bit;

        This array must be declared at least as large as the double number of
        64-bit elements to be read from the instrument (Count*2) - failure to
        do so may result in a system failure.

Return Value

        Returns the status code of this operation.  The status code  either
        indicates success or describes an error or warning condition.  You
        examine the status code from each call to an instrument driver
```

```
function to determine if an error occurred.
```

```
To obtain a text description of the status code, call the
bu6100_error_message function.  To obtain additional information
about the error condition, call the bu6100_GetError function.  To
clear the error information from the driver, call the
bu6100_ClearError function.

The general meaning of the status code is as follows:

Value                  Meaning
-------------------------------
0                      Success
Positive Values        Warnings
Negative Values        Errors

This instrument driver returns errors and warnings defined by other
sources.  The following table defines the ranges of additional status
codes that this driver can return.  The table lists the different
include files that contain the defined constants for the particular
status codes:

Numeric Range (in Hex)   Status Code Types
-------------------------------------------------
3FFA0000 to 3FFA1FFF     IVI      Warnings
3FFF0000 to 3FFFFFFF     VISA     Warnings
3FFC0000 to 3FFCFFFF     VXIPnP   Driver Warnings

BFFA0000 to BFFA1FFF     IVI      Errors
BFFF0000 to BFFFFFFF     VISA     Errors
BFFC0000 to BFFCFFFF     VXIPnP   Driver Errors
```

5.5.28. bu6100_freeDram

```
    ViStatus bu6100_freeDram (ViSession instrumentHandle, ViInt32 offset);

Purpose

    Deallocates the DRAM segment previously allocated by allocDRAM function

Parameter List

    instrumentHandle

        Variable Type       ViSession

        The Instrument Handle is used to identify the unique session or
        communication channel between the driver and the instrument.

        If more than one instrument of the same model type is used, this
        Handle will be used to differentiate between them.


    offset

        Variable Type       ViInt32

        Selects the offset of the memory segment within
        the DRAM to be deallocated

Return Value

        Returns the status code of this operation.  The status code  either
        indicates success or describes an error or warning condition.  You
        examine the status code from each call to an instrument driver
        function to determine if an error occurred.

        To obtain a text description of the status code, call the
        bu6100_error_message function.  To obtain additional information
        about the error condition, call the bu6100_GetError function.  To
        clear the error information from the driver, call the
        bu6100_ClearError function.

        The general meaning of the status code is as follows:

        Value                  Meaning
        -------------------------------
        0                      Success
        Positive Values        Warnings
        Negative Values        Errors

        This instrument driver returns errors and warnings defined by other
        sources.  The following table defines the ranges of additional status
        codes that this driver can return.  The table lists the different
        include files that contain the defined constants for the particular
        status codes:

        Numeric Range (in Hex)   Status Code Types
        -------------------------------------------------
        3FFA0000 to 3FFA1FFF     IVI      Warnings
        3FFF0000 to 3FFFFFFF     VISA     Warnings
        3FFC0000 to 3FFCFFFF     VXIPnP   Driver Warnings

        BFFA0000 to BFFA1FFF     IVI      Errors
        BFFF0000 to BFFFFFFF     VISA     Errors
        BFFC0000 to BFFCFFFF     VXIPnP   Driver Errors
```

5.5.29. bu6100_get1588config

```
ViStatus bu6100_get1588config (ViSession instrumentHandle,
                               ViPInt32 enabled, ViPInt32 threshold,
                               ViPInt32 clock_divider, ViPInt32 pid_p,
                               ViPInt32 pid_i, ViPInt32 pid_d);
```

```
Purpose

    This function returns the configuration of the 1588 interface of the
    ProDAQ 6100 module.

Parameter List

    instrumentHandle

        Variable Type       ViSession

        The Instrument Handle is used to identify the unique session or
        communication channel between the driver and the instrument.

        If more than one instrument of the same model type is used, this
        Handle will be used to differentiate between them.


    enabled

        Variable Type       ViInt32 (passed by reference)

        This parameter returns whether the 1588 interface of the ProDAQ 6100
        module is disabled, enabled or enabled in slave-only mode.
        In the slave-only mode the 6100 will never take a role of the 1588
        clock master.

        Possible values are:
        BU6100_1588_DISABLED            0  1588 interface is disabled;
        BU6100_1588_ENABLED             1  1588 interface is enabled;
        BU6100_1588_ENABLED_SLAVE_ONLY  2  1588 interface is enabled as
                                           a slave-only device;

    threshold

        Variable Type       ViInt32 (passed by reference)

        This parameter returns the synchronization thershold (in
        nanoseconds). It is used when the 6100 module operates as a 1588
        slave device. If the offset from the 1588 Master is less then the
        specified threshold, the module is considered as synchronized. If the
        offset is bigger than the threshold, the module status will be "not
        synchronized".

    clock_divider

        Variable Type       ViInt32 (passed by reference)

        This parameter returns the content of the TCLK divider of the 1588
        interface.

    pid_p

        Variable Type       ViInt32 (passed by reference)

        This parameter returns the 'p' coefficient of the PTP PID control
        loop.

    pid_i

        Variable Type       ViInt32 (passed by reference)

        This parameter returns the 'i' coefficient of the PTP PID control
        loop.

    pid_d

        Variable Type       ViInt32 (passed by reference)
```

```
This parameter returns the 'd' coefficient of the PTP PID control
loop.
```

```
Return Value

        Returns the status code of this operation.  The status code  either
        indicates success or describes an error or warning condition.  You
        examine the status code from each call to an instrument driver
        function to determine if an error occurred.

        To obtain a text description of the status code, call the
        bu6100_error_message function.  To obtain additional information
        about the error condition, call the bu6100_GetError function.  To
        clear the error information from the driver, call the
        bu6100_ClearError function.

        The general meaning of the status code is as follows:

        Value                  Meaning
        -------------------------------
        0                      Success
        Positive Values        Warnings
        Negative Values        Errors

        This instrument driver returns errors and warnings defined by other
        sources.  The following table defines the ranges of additional status
        codes that this driver can return.  The table lists the different
        include files that contain the defined constants for the particular
        status codes:

        Numeric Range (in Hex)   Status Code Types
        -------------------------------------------------
        3FFA0000 to 3FFA1FFF     IVI      Warnings
        3FFF0000 to 3FFFFFFF     VISA     Warnings
        3FFC0000 to 3FFCFFFF     VXIPnP   Driver Warnings

        BFFA0000 to BFFA1FFF     IVI      Errors
        BFFF0000 to BFFFFFFF     VISA     Errors
        BFFC0000 to BFFCFFFF     VXIPnP   Driver Errors
```

5.5.30. bu6100_get1588status

```
ViStatus bu6100_get1588status (ViSession instrumentHandle,
                               ViPInt32 port_state,
                               ViChar _VI_FAR parent_uuid[],
                               ViChar _VI_FAR grandmaster_uuid[],
                               ViChar _VI_FAR grandmasterTraceability[],
                               ViChar _VI_FAR subdomain[],
                               ViPInt32 variance);
```

```
  Purpose

      This function returns the status of the 1588 interface of the ProDAQ 6100
      module.

  Parameter List

      instrumentHandle

          Variable Type       ViSession

          The Instrument Handle is used to identify the unique session or
          communication channel between the driver and the instrument.

          If more than one instrument of the same model type is used, this
          Handle will be used to differentiate between them.


      port_state

          Variable Type       ViInt32 (passed by reference)

          This parameter returns the state of the 1588 port. Possible values
          are:

          BU6100_1588_PORT_INITIALIZING    0
          BU6100_1588_PORT_FAULTY          1
          BU6100_1588_PORT_DISABLED        2
          BU6100_1588_PORT_LISTENING       3
          BU6100_1588_PORT_PRE_MASTER      4
          BU6100_1588_PORT_MASTER          5
          BU6100_1588_PORT_PASSIVE         6
          BU6100_1588_PORT_UNCALIBRATED    7
          BU6100_1588_PORT_SLAVE_SYNC      8
          BU6100_1588_PORT_SLAVE_NOT_SYNCH 9


      parent_uuid

          Variable Type       ViChar[]

          This parameter returns the parent 6-bytes mac address string in the
          format: xx:xx:xx:xx:xx:xx. The buffer should be allocated prior to
          the function call with the appropriate size. This parameter can be
          NULL pointer.

      grandmaster_uuid

          Variable Type       ViChar[]

          This parameter returns the grandmaster 6-bytes mac address string in
          the format: xx:xx:xx:xx:xx:xx. The buffer should be allocated prior
          to the function call with the appropriate size. This parameter can be
          NULL pointer.

      grandmasterTraceability

          Variable Type       ViChar[]

          This parameter returns the current grandmaster traceability to UTC:
          GPS, NTP, HAND or ATOM (Vendors can define others).

      subdomain

          Variable Type       ViChar[]
```

```
        This parameter returns the current subdomain name: _DFLT, _ALT1,
        _ALT2... and so on.

    variance

        Variable Type       ViInt32 (passed by reference)

        This parameter returns the current observed variance of the parent
        clock (in ns).

Return Value

        Returns the status code of this operation.  The status code  either
        indicates success or describes an error or warning condition.  You
        examine the status code from each call to an instrument driver
        function to determine if an error occurred.

        To obtain a text description of the status code, call the
        bu6100_error_message function.  To obtain additional information
        about the error condition, call the bu6100_GetError function.  To
        clear the error information from the driver, call the
        bu6100_ClearError function.

        The general meaning of the status code is as follows:

        Value                  Meaning
        -------------------------------
        0                      Success
        Positive Values        Warnings
        Negative Values        Errors

        This instrument driver returns errors and warnings defined by other
        sources.  The following table defines the ranges of additional status
        codes that this driver can return.  The table lists the different
        include files that contain the defined constants for the particular
        status codes:

        Numeric Range (in Hex)   Status Code Types
        -------------------------------------------------
        3FFA0000 to 3FFA1FFF     IVI      Warnings
        3FFF0000 to 3FFFFFFF     VISA     Warnings
        3FFC0000 to 3FFCFFFF     VXIPnP   Driver Warnings

        BFFA0000 to BFFA1FFF     IVI      Errors
        BFFF0000 to BFFFFFFF     VISA     Errors
        BFFC0000 to BFFCFFFF     VXIPnP   Driver Errors
```

5.5.31. bu6100_get1588timing

```
ViStatus bu6100_get1588timing (ViSession instrumentHandle,
                               ViPReal64 PTP_time,
                               ViChar _VI_FAR localTime[],
                               ViPReal64 mastertoSlave_delay,
                               ViPReal64 slavetoMaster_delay,
                               ViPReal64 oneWay_delay,
                               ViPReal64 offset_from_master,
                               ViPReal64 observed_drift);
```

Purpose

This function returns the values of various timing parameters of 1588 synchronization.

Parameter List

instrumentHandle

Variable Type	ViSession

The Instrument Handle is used to identify the unique session or communication channel between the driver and the instrument.

If more than one instrument of the same model type is used, this Handle will be used to differentiate between them.

PTP_time

Variable Type	ViReal64 (passed by reference)

This parameter returns the current PTP time in seconds since January 1, 1970.

localTime

Variable Type	ViChar[]

Returns the current local date/time in 24-character UNIX-format.

mastertoSlave_delay

Variable Type	ViReal64 (passed by reference)

Returns the master-to-slave delay in seconds. This value is valid only when the board operates as a PTP slave.

slavetoMaster_delay

Variable Type	ViReal64 (passed by reference)

Returns the slave-to-master delay in seconds. This value is valid only when the board operates as a PTP slave.

oneWay_delay

Variable Type	ViReal64 (passed by reference)

Returns the one-way delay in seconds. This value is valid only when the board operates as a PTP slave.

offset_from_master

Variable Type	ViReal64 (passed by reference)

Returns the offset from master in seconds. This value is valid only when the board operates as a PTP slave.

observed_drift

Variable Type	ViReal64 (passed by reference)

Returns the observed clock drift in seconds. This value is valid only

```
         when the board operates as a PTP slave.
```

```
 Return Value

        Returns the status code of this operation.  The status code  either
        indicates success or describes an error or warning condition.  You
        examine the status code from each call to an instrument driver
        function to determine if an error occurred.

        To obtain a text description of the status code, call the
        bu6100_error_message function.  To obtain additional information
        about the error condition, call the bu6100_GetError function.  To
        clear the error information from the driver, call the
        bu6100_ClearError function.

        The general meaning of the status code is as follows:

        Value                  Meaning
        -------------------------------
        0                      Success
        Positive Values        Warnings
        Negative Values        Errors

        This instrument driver returns errors and warnings defined by other
        sources.  The following table defines the ranges of additional status
        codes that this driver can return.  The table lists the different
        include files that contain the defined constants for the particular
        status codes:

        Numeric Range (in Hex)   Status Code Types
        -------------------------------------------------
        3FFA0000 to 3FFA1FFF     IVI      Warnings
        3FFF0000 to 3FFFFFFF     VISA     Warnings
        3FFC0000 to 3FFCFFFF     VXIPnP   Driver Warnings

        BFFA0000 to BFFA1FFF     IVI      Errors
        BFFF0000 to BFFFFFFF     VISA     Errors
        BFFC0000 to BFFCFFFF     VXIPnP   Driver Errors
```

5.5.32. bu6100_GetAttributeViBoolean

```
ViStatus bu6100_GetAttributeViBoolean (ViSession instrumentHandle,
                                       ViChar _VI_FAR channelName[],
                                       ViAttr attributeID,
                                       ViPBoolean attributeValue);
```

Purpose

This function queries the value of a ViBoolean attribute.

You can use this function to get the values of instrument- specific attributes and inherent IVI attributes. If the attribute represents an instrument state, this function performs instrument I/O in the following cases:

- State caching is disabled for the entire session or for the particular attribute.

- State caching is enabled and the currently cached value is invalid.

Parameter List

instrumentHandle

Variable Type	ViSession

The ViSession handle that you obtain from the bu6100_init or bu6100_InitWithOptions function. The handle identifies a particular instrument session.

Default Value: None

channelName

Variable Type	ViChar[]

If the attribute is channel-based, this parameter specifies the name of the channel on which to obtain the value of the attribute. If the attribute is not channel-based, then pass VI_NULL or an empty string.

Valid Channel Names: 1

Default Value: ""

attributeID

Variable Type	ViAttr

Pass the ID of an attribute.

From the function panel window, you can use this control as follows.

- Click on the control or press <ENTER>, <spacebar>, or <ctrl-down arrow>, to display a dialog box containing a hierarchical list of the available attributes. Help text is shown for each attribute. Select an attribute by double-clicking on it or by selecting it and then pressing <ENTER>.

  A ring control at the top of the dialog box allows you to see all IVI attributes or only the attributes of the ViBoolean type. If you choose to see all IVI attributes, the data types appear to the right of the attribute names in the list box. Attributes with data types other than ViBoolean are dim. If you select an attribute data type that is dim, LabWindows/CVI transfers you to the function panel for the corresponding function that is consistent with the data type.

- If you want to enter a variable name, press <CTRL-T> to change this ring control to a manual input box.

- If the attribute in this ring control has named constants as valid values, you can view the constants by moving to the Attribute Value control and pressing <ENTER>.

attributeValue

Variable Type ViBoolean (passed by reference)

Returns the current value of the attribute. Pass the address of a ViBoolean variable.

From the function panel window, you can use this control as follows.

- If the attribute currently showing in the Attribute ID ring control has named constants as valid values, you can view a list of the constants by pressing <ENTER> on this control. Select a value by double-clicking on it or by selecting it and then pressing <ENTER>.

Return Value

Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You examine the status code from each call to an instrument driver function to determine if an error occurred.

To obtain a text description of the status code, call the bu6100_error_message function. To obtain additional information about the error condition, call the bu6100_GetError function. To clear the error information from the driver, call the bu6100_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

This instrument driver returns errors and warnings defined by other sources. The following table defines the ranges of additional status codes that this driver can return. The table lists the different include files that contain the defined constants for the particular status codes:

Numeric Range (in Hex)	Status Code Types	
3FFA0000 to 3FFA1FFF	IVI	Warnings
3FFF0000 to 3FFFFFFF	VISA	Warnings
3FFC0000 to 3FFCFFFF	VXIPnP	Driver Warnings
BFFA0000 to BFFA1FFF	IVI	Errors
BFFF0000 to BFFFFFFF	VISA	Errors
BFFC0000 to BFFCFFFF	VXIPnP	Driver Errors

5.5.33. bu6100_GetAttributeViInt32

```
    ViStatus bu6100_GetAttributeViInt32 (ViSession instrumentHandle,
                                         ViChar _VI_FAR channelName[],
                                         ViAttr attributeID,
                                         ViPInt32 attributeValue);

Purpose

    This function queries the value of a ViInt32 attribute.

    You can use this function to get the values of instrument- specific
    attributes and inherent IVI attributes.  If the attribute represents an
    instrument state, this function performs instrument I/O in the following
    cases:

    - State caching is disabled for the entire session or for the particular
    attribute.

    - State caching is enabled and the currently cached value is invalid.



Parameter List

    instrumentHandle

        Variable Type       ViSession

        The ViSession handle that you obtain from the bu6100_init or
        bu6100_InitWithOptions function.  The handle identifies a particular
        instrument session.

        Default Value:  None


    channelName

        Variable Type       ViChar[]

        If the attribute is channel-based, this parameter specifies the name
        of the channel on which to obtain the value of the attribute. If the
        attribute is not channel-based, then pass VI_NULL or an empty string.

        Valid Channel Names:  1

        Default Value:  ""


    attributeID

        Variable Type       ViAttr

        Pass the ID of an attribute.

        From the function panel window, you can use this control as follows.

        - Click on the control or press <ENTER>, <spacebar>, or
          <ctrl-down arrow>, to display a dialog box containing a
          hierarchical list of the available attributes.  Help text is
          shown for each attribute.  Select an attribute by
          double-clicking on it or by selecting it and then pressing
          <ENTER>.

          A ring control at the top of the dialog box allows you to see
          all IVI attributes or only the attributes of the ViInt32 type.
          If you choose to see all IVI attributes, the data types appear
          to the right of the attribute names in the list box.
          Attributes with data types other than ViInt32 are dim. If
          you select an attribute data type that is dim, LabWindows/CVI
          transfers you to the function panel for the corresponding
          function that is consistent with the data type.

        - If you want to enter a variable name, press <CTRL-T> to change
          this ring control to a manual input box.
        - If the attribute in this ring control has named constants as
          valid values, you can view the constants by moving to the
          Attribute Value control and pressing <ENTER>.


    attributeValue

        Variable Type       ViInt32 (passed by reference)

        Returns the current value of the attribute.  Pass the address of a
        ViInt32 variable.

        From the function panel window, you can use this control as follows.

        - If the attribute currently showing in the Attribute ID ring
          control has named constants as valid values, you can view a
          list of the constants by pressing <ENTER> on this control.
          Select a value by double-clicking on it or by selecting it and
          then pressing <ENTER>.


Return Value

        Returns the status code of this operation.  The status code  either
        indicates success or describes an error or warning condition.  You
        examine the status code from each call to an instrument driver
        function to determine if an error occurred.

        To obtain a text description of the status code, call the
        bu6100_error_message function.  To obtain additional information
        about the error condition, call the bu6100_GetError function.  To
        clear the error information from the driver, call the
        bu6100_ClearError function.

        The general meaning of the status code is as follows:

        Value                  Meaning
        -------------------------------
        0                      Success
        Positive Values        Warnings
        Negative Values        Errors

        This instrument driver returns errors and warnings defined by other
        sources.  The following table defines the ranges of additional status
        codes that this driver can return.  The table lists the different
        include files that contain the defined constants for the particular
        status codes:

        Numeric Range (in Hex)   Status Code Types
        -------------------------------------------------
        3FFA0000 to 3FFA1FFF     IVI      Warnings
        3FFF0000 to 3FFFFFFF     VISA     Warnings
        3FFC0000 to 3FFCFFFF     VXIPnP   Driver Warnings

        BFFA0000 to BFFA1FFF     IVI      Errors
        BFFF0000 to BFFFFFFF     VISA     Errors
        BFFC0000 to BFFCFFFF     VXIPnP   Driver Errors
```

5.5.34. bu6100_GetAttributeViReal64

```
ViStatus bu6100_GetAttributeViReal64 (ViSession instrumentHandle,
                                      ViChar _VI_FAR channelName[],
                                      ViAttr attributeID,
                                      ViPReal64 attributeValue);
```

Purpose

This function queries the value of a ViReal64 attribute.

You can use this function to get the values of instrument- specific attributes and inherent IVI attributes. If the attribute represents an instrument state, this function performs instrument I/O in the following cases:

- State caching is disabled for the entire session or for the particular attribute.

- State caching is enabled and the currently cached value is invalid.

Parameter List

instrumentHandle

Variable Type   ViSession

The ViSession handle that you obtain from the bu6100_init or bu6100_InitWithOptions function. The handle identifies a particular instrument session.

Default Value: None

channelName

Variable Type   ViChar[]

If the attribute is channel-based, this parameter specifies the name of the channel on which to obtain the value of the attribute. If the attribute is not channel-based, then pass VI_NULL or an empty string.

Valid Channel Names: 1

Default Value: ""

attributeID

Variable Type   ViAttr

Pass the ID of an attribute.

From the function panel window, you can use this control as follows.

- Click on the control or press <ENTER>, <spacebar>, or <ctrl-down arrow>, to display a dialog box containing a hierarchical list of the available attributes. Help text is shown for each attribute. Select an attribute by double-clicking on it or by selecting it and then pressing <ENTER>.

  A ring control at the top of the dialog box allows you to see all IVI attributes or only the attributes of the ViReal64 type. If you choose to see all IVI attributes, the data types appear to the right of the attribute names in the list box. Attributes with data types other than ViReal64 are dim. If you select an attribute data type that is dim, LabWindows/CVI transfers you to the function panel for the corresponding function that is consistent with the data type.

- If you want to enter a variable name, press <CTRL-T> to change this ring control to a manual input box.

```
        - If the attribute in this ring control has named constants as
          valid values, you can view the constants by moving to the
          Attribute Value control and pressing <ENTER>.


    attributeValue

        Variable Type       ViReal64 (passed by reference)

        Returns the current value of the attribute.  Pass the address of a
        ViReal64 variable.

        From the function panel window, you can use this control as follows.

        - If the attribute currently showing in the Attribute ID ring
          control has named constants as valid values, you can view a
          list of the constants by pressing <ENTER> on this control.
          Select a value by double-clicking on it or by selecting it and
          then pressing <ENTER>.


  Return Value

        Returns the status code of this operation.  The status code  either
        indicates success or describes an error or warning condition.  You
        examine the status code from each call to an instrument driver
        function to determine if an error occurred.

        To obtain a text description of the status code, call the
        bu6100_error_message function.  To obtain additional information
        about the error condition, call the bu6100_GetError function.  To
        clear the error information from the driver, call the
        bu6100_ClearError function.

        The general meaning of the status code is as follows:

        Value                  Meaning
        -------------------------------
        0                      Success
        Positive Values        Warnings
        Negative Values        Errors

        This instrument driver returns errors and warnings defined by other
        sources.  The following table defines the ranges of additional status
        codes that this driver can return.  The table lists the different
        include files that contain the defined constants for the particular
        status codes:

        Numeric Range (in Hex)   Status Code Types
        -------------------------------------------------
        3FFA0000 to 3FFA1FFF     IVI      Warnings
        3FFF0000 to 3FFFFFFF     VISA     Warnings
        3FFC0000 to 3FFCFFFF     VXIPnP   Driver Warnings

        BFFA0000 to BFFA1FFF     IVI      Errors
        BFFF0000 to BFFFFFFF     VISA     Errors
        BFFC0000 to BFFCFFFF     VXIPnP   Driver Errors
```

5.5.35. bu6100_GetAttributeViSession

```
ViStatus bu6100_GetAttributeViSession (ViSession instrumentHandle,
                                       ViChar _VI_FAR channelName[],
                                       ViAttr attributeID,
                                       ViPSession attributeValue);
```

```
Purpose

    This function queries the value of a ViSession attribute.

    You can use this function to get the values of instrument- specific
    attributes and inherent IVI attributes.  If the attribute represents an
    instrument state, this function performs instrument I/O in the following
    cases:

    - State caching is disabled for the entire session or for the particular
    attribute.

    - State caching is enabled and the currently cached value is invalid.



Parameter List

    instrumentHandle

        Variable Type       ViSession

        The ViSession handle that you obtain from the bu6100_init or
        bu6100_InitWithOptions function.  The handle identifies a particular
        instrument session.

        Default Value:  None


    channelName

        Variable Type       ViChar[]

        If the attribute is channel-based, this parameter specifies the name
        of the channel on which to obtain the value of the attribute. If the
        attribute is not channel-based, then pass VI_NULL or an empty string.

        Valid Channel Names:  1

        Default Value:  ""


    attributeID

        Variable Type       ViAttr

        Pass the ID of an attribute.

        From the function panel window, you can use this control as follows.

        - Click on the control or press <ENTER>, <spacebar>, or
          <ctrl-down arrow>, to display a dialog box containing a
          hierarchical list of the available attributes.  Help text is
          shown for each attribute.  Select an attribute by
          double-clicking on it or by selecting it and then pressing
          <ENTER>.

          A ring control at the top of the dialog box allows you to see
          all IVI attributes or only the attributes of the ViSession
          type.  If you choose to see all IVI attributes, the data types
          appear to the right of the attribute names in the list box.
          Attributes with data types other than ViSession are dim. If
          you select an attribute data type that is dim, LabWindows/CVI
          transfers you to the function panel for the corresponding
          function that is consistent with the data type.

        - If you want to enter a variable name, press <CTRL-T> to change
          this ring control to a manual input box.
        - If the attribute in this ring control has named constants as
          valid values, you can view the constants by moving to the
          Attribute Value control and pressing <ENTER>.


    attributeValue

        Variable Type       ViSession (passed by reference)

        Returns the current value of the attribute.  Pass the address of a
        ViSession variable.

        From the function panel window, you can use this control as follows.

        - If the attribute currently showing in the Attribute ID ring
          control has named constants as valid values, you can view a
          list of the constants by pressing <ENTER> on this control.
          Select a value by double-clicking on it or by selecting it and
          then pressing <ENTER>.


Return Value

        Returns the status code of this operation.  The status code  either
        indicates success or describes an error or warning condition.  You
        examine the status code from each call to an instrument driver
        function to determine if an error occurred.

        To obtain a text description of the status code, call the
        bu6100_error_message function.  To obtain additional information
        about the error condition, call the bu6100_GetError function.  To
        clear the error information from the driver, call the
        bu6100_ClearError function.

        The general meaning of the status code is as follows:

        Value                  Meaning
        -------------------------------
        0                      Success
        Positive Values        Warnings
        Negative Values        Errors

        This instrument driver returns errors and warnings defined by other
        sources.  The following table defines the ranges of additional status
        codes that this driver can return.  The table lists the different
        include files that contain the defined constants for the particular
        status codes:

        Numeric Range (in Hex)   Status Code Types
        -------------------------------------------------
        3FFA0000 to 3FFA1FFF     IVI      Warnings
        3FFF0000 to 3FFFFFFF     VISA     Warnings
        3FFC0000 to 3FFCFFFF     VXIPnP   Driver Warnings

        BFFA0000 to BFFA1FFF     IVI      Errors
        BFFF0000 to BFFFFFFF     VISA     Errors
        BFFC0000 to BFFCFFFF     VXIPnP   Driver Errors
```

5.5.36. bu6100_GetAttributeViString

```
ViStatus bu6100_GetAttributeViString (ViSession instrumentHandle,
                                      ViChar _VI_FAR channelName[],
                                      ViAttr attributeID,
                                      ViInt32 bufferSize,
                                      ViChar _VI_FAR attributeValue[]);
```

Purpose

This function queries the value of a ViString attribute.

You can use this function to get the values of instrument- specific attributes and inherent IVI attributes. If the attribute represents an instrument state, this function performs instrument I/O in the following cases:

- State caching is disabled for the entire session or for the particular attribute.

- State caching is enabled and the currently cached value is invalid.

You must provide a ViChar array to serve as a buffer for the value. You pass the number of bytes in the buffer as the Buffer Size parameter. If the current value of the attribute, including the terminating NUL byte, is larger than the size you indicate in the Buffer Size parameter, the function copies Buffer Size - 1 bytes into the buffer, places an ASCII NUL byte at the end of the buffer, and returns the buffer size you must pass to get the entire value. For example, if the value is "123456" and the Buffer Size is 4, the function places "123" into the buffer and returns 7.

If you want to call this function just to get the required buffer size, you can pass 0 for the Buffer Size and VI_NULL for the Attribute Value buffer.

If you want the function to fill in the buffer regardless of the number of bytes in the value, pass a negative number for the Buffer Size parameter.

Parameter List

instrumentHandle

Variable Type	ViSession

The ViSession handle that you obtain from the bu6100_init or bu6100_InitWithOptions function. The handle identifies a particular instrument session.

Default Value: None

channelName

Variable Type	ViChar[]

If the attribute is channel-based, this parameter specifies the name of the channel on which to obtain the value of the attribute. If the attribute is not channel-based, then pass VI_NULL or an empty string.

Valid Channel Names: 1

Default Value: ""

attributeID

Variable Type	ViAttr

Pass the ID of an attribute.

From the function panel window, you can use this control as follows.

```
    - Click on the control or press <ENTER>, <spacebar>, or
      <ctrl-down arrow>, to display a dialog box containing a
      hierarchical list of the available attributes.  Help text is
      shown for each attribute.  Select an attribute by
      double-clicking on it or by selecting it and then pressing
      <ENTER>.

      A ring control at the top of the dialog box allows you to see
      all IVI attributes or only the attributes of the ViString
      type.  If you choose to see all IVI attributes, the data types
      appear to the right of the attribute names in the list box.
      Attributes with data types other than ViString are dim. If
      you select an attribute data type that is dim, LabWindows/CVI
      transfers you to the function panel for the corresponding
      function that is consistent with the data type.

    - If you want to enter a variable name, press <CTRL-T> to change
      this ring control to a manual input box.

    - If the attribute in this ring control has named constants as
      valid values, you can view the constants by moving to the
      Attribute Value control and pressing <ENTER>.


bufferSize

    Variable Type       ViInt32

    Pass the number of bytes in the ViChar array you specify for the
    Attribute Value parameter.

    If the current value of the attribute, including the terminating NUL
    byte, contains more bytes that you indicate in this parameter, the
    function copies Buffer Size - 1 bytes into the buffer, places an
    ASCII NUL byte at the end of the buffer, and returns the buffer size
    you must pass to get the entire value.  For example, if the value is
    "123456" and the Buffer Size is 4, the function places "123" into the
    buffer and returns 7.

    If you pass a negative number, the function copies the value to the
    buffer regardless of the number of bytes in the value.

    If you pass 0, you can pass VI_NULL for the Attribute Value buffer
    parameter.

    Default Value: 512


attributeValue

    Variable Type       ViChar[]

    The buffer in which the function returns the current value of the
    attribute.  The buffer must be of type ViChar and have at least as
    many bytes as indicated in the Buffer Size parameter.

    If the current value of the attribute, including the terminating NUL
    byte, contains more bytes that you indicate in this parameter, the
    function copies Buffer Size - 1 bytes into the buffer, places an
    ASCII NUL byte at the end of the buffer, and returns the buffer size
    you must pass to get the entire value.  For example, if the value is
    "123456" and the Buffer Size is 4, the function places "123" into the
    buffer and returns 7.

    If you specify 0 for the Buffer Size parameter, you can pass VI_NULL
    for this parameter.

    From the function panel window, you can use this control as follows.

    - If the attribute currently showing in the Attribute ID ring
      control has named constants as valid values, you can view a
      list of the constants by pressing <ENTER> on this control.
      Select a value by double-clicking on it or by selecting it and
      then pressing <ENTER>.
```

```
Return Value

        Returns the status code of this operation.  The status code  either
        indicates success or describes an error or warning condition.  You
        examine the status code from each call to an instrument driver
        function to determine if an error occurred.

        To obtain a text description of the status code, call the
        bu6100_error_message function.  To obtain additional information
        about the error condition, call the bu6100_GetError function.  To
        clear the error information from the driver, call the
        bu6100_ClearError function.

        The general meaning of the status code is as follows:

        Value                  Meaning
        -------------------------------
        0                      Success
        Positive Values        Warnings
        Negative Values        Errors

        This instrument driver returns errors and warnings defined by other
        sources.  The following table defines the ranges of additional status
        codes that this driver can return.  The table lists the different
        include files that contain the defined constants for the particular
        status codes:

        Numeric Range (in Hex)   Status Code Types
        -------------------------------------------------
        3FFA0000 to 3FFA1FFF     IVI      Warnings
        3FFF0000 to 3FFFFFFF     VISA     Warnings
        3FFC0000 to 3FFCFFFF     VXIPnP   Driver Warnings

        BFFA0000 to BFFA1FFF     IVI      Errors
        BFFF0000 to BFFFFFFF     VISA     Errors
        BFFC0000 to BFFCFFFF     VXIPnP   Driver Errors
```

5.5.37. bu6100_getCBstatus

```
     ViStatus bu6100_getCBstatus (ViSession instrumentHandle,
                                  ViInt16 functionCard, ViPInt32 n_ofSamples,
                                  ViPBoolean overflow);

  Purpose

     This function returns the current status of the Circular Buffer for
     particular Function Card.

  Parameter List

     instrumentHandle

         Variable Type       ViSession

         The Instrument Handle is used to identify the unique session or
         communication channel between the driver and the instrument.

         If more than one instrument of the same model type is used, this
         Handle will be used to differentiate between them.


     functionCard

         Variable Type       ViInt16

         Specifies the Function Card for which the Circular Buffer will be
         configured.

         Valid Values: 1, 2, 3, 4

         Default Value: 1

     n_ofSamples

         Variable Type       ViInt32 (passed by reference)

         This parameter returns the number of 32-bit words currently stored in
         the Circular Buffer.

     overflow

         Variable Type       ViBoolean (passed by reference)

         This parameter tells whether overflow of the Circular Buffer
         happened.

  Return Value

         Returns the status code of this operation.  The status code  either
         indicates success or describes an error or warning condition.  You
         examine the status code from each call to an instrument driver
         function to determine if an error occurred.

         To obtain a text description of the status code, call the
         bu6100_error_message function.  To obtain additional information
         about the error condition, call the bu6100_GetError function.  To
         clear the error information from the driver, call the
         bu6100_ClearError function.

         The general meaning of the status code is as follows:

         Value                  Meaning
         -------------------------------
         0                      Success
         Positive Values        Warnings
         Negative Values        Errors

         This instrument driver returns errors and warnings defined by other
         sources.  The following table defines the ranges of additional status
         codes that this driver can return.  The table lists the different
         include files that contain the defined constants for the particular
         status codes:
```

```
Numeric Range (in Hex)   Status Code Types
-------------------------------------------------
3FFA0000 to 3FFA1FFF     IVI      Warnings
3FFF0000 to 3FFFFFFF     VISA     Warnings
3FFC0000 to 3FFCFFFF     VXIPnP   Driver Warnings

BFFA0000 to BFFA1FFF     IVI      Errors
BFFF0000 to BFFFFFFF     VISA     Errors
BFFC0000 to BFFCFFFF     VXIPnP   Driver Errors
```

5.5.38. bu6100_getConfigCB

```
ViStatus bu6100_getConfigCB (ViSession instrumentHandle,
                             ViInt16 functionCard, ViPInt32 function,
                             ViPInt32 FIFOAddress, ViPInt32 address,
                             ViPInt32 length, ViPInt32 threshold);
```

Purpose

This function returns the configuration of the Circular Buffer for particular Function Card.

Parameter List

instrumentHandle

Variable Type	ViSession

The Instrument Handle is used to identify the unique session or communication channel between the driver and the instrument.

If more than one instrument of the same model type is used, this Handle will be used to differentiate between them.

functionCard

Variable Type	ViInt16

Specifies the Function Card for which the Circular Buffer will be configured.
Valid Values: 1, 2, 3, 4.

function

Variable Type	ViInt32 (passed by reference)

This parameter returns the type of access to Circular Buffer which will be performed by DSP.
Possible values are:

bu3100_LIST_READ16	0	16-bit Packed block read
bu3100_LIST_READ32	1	32-bit block read
bu3100_LIST_READ64	2	64-bit block read
bu3100_LIST_WRITE16	3	16-bit packed block write
bu3100_LIST_WRITE32	4	32-bit block write
bu3100_LIST_WRITE64	5	64-bit block write

FIFOAddress

Variable Type	ViInt32 (passed by reference)

This parameter returns the Address of Function Card FIFO in the Function Card address space.

address

Variable Type	ViInt32 (passed by reference)

This parameter returns the Starting Address of the Circular Buffer in bu6100 DRAM.

length

Variable Type	ViInt32 (passed by reference)

This parameter specifies the length of the Circular Buffer in 32-bit words.

threshold

Variable Type	ViInt32 (passed by reference)

This parameter returns the threshold value - the number of 32-bit words, after collection of which the DSP will send a signal

```
          (interrupt) to the VXI host.
```

```
Return Value

        Returns the status code of this operation.  The status code  either
        indicates success or describes an error or warning condition.  You
        examine the status code from each call to an instrument driver
        function to determine if an error occurred.

        To obtain a text description of the status code, call the
        bu6100_error_message function.  To obtain additional information
        about the error condition, call the bu6100_GetError function.  To
        clear the error information from the driver, call the
        bu6100_ClearError function.

        The general meaning of the status code is as follows:

        Value                  Meaning
        -------------------------------
        0                      Success
        Positive Values        Warnings
        Negative Values        Errors

        This instrument driver returns errors and warnings defined by other
        sources.  The following table defines the ranges of additional status
        codes that this driver can return.  The table lists the different
        include files that contain the defined constants for the particular
        status codes:

        Numeric Range (in Hex)   Status Code Types
        -------------------------------------------------
        3FFA0000 to 3FFA1FFF     IVI      Warnings
        3FFF0000 to 3FFFFFFF     VISA     Warnings
        3FFC0000 to 3FFCFFFF     VXIPnP   Driver Warnings

        BFFA0000 to BFFA1FFF     IVI      Errors
        BFFF0000 to BFFFFFFF     VISA     Errors
        BFFC0000 to BFFCFFFF     VXIPnP   Driver Errors
```

5.5.39. bu6100_GetError

```
ViStatus bu6100_GetError (ViSession instrumentHandle, ViPStatus code,
                          ViInt32 bufferSize,
                          ViChar _VI_FAR description[]);
```

```
Purpose

    This function retrieves and then clears the IVI error information for the
    session or the current execution thread. One exception exists: If the
    BufferSize parameter is 0, the function does not clear the error
    information. By passing 0 for the buffer size, the caller can ascertain
    the buffer size required to get the entire error description string and
    then call the function again with a sufficiently large buffer.

    If the user specifies a valid IVI session for the InstrumentHandle
    parameter, Get Error retrieves and then clears the error information for
    the session.  If the user passes VI_NULL for the InstrumentHandle
    parameter, this function retrieves and then clears the error information
    for the current execution thread.  If the InstrumentHandle parameter is
    an invalid session, the function does nothing and returns an error.
    Normally, the error information describes the first error that occurred
    since the user last called bu6100_GetError or bu6100_ClearError.


Parameter List

    instrumentHandle

        Variable Type       ViSession

        The ViSession handle that you obtain from the bu6100_init or
        bu6100_InitWithOptions function.  The handle identifies a particular
        instrument session.

        Default Value:  None


    code

        Variable Type       ViStatus (passed by reference)

        Returns the error code for the session or execution thread.

        If you pass 0 for the Buffer Size, you can pass VI_NULL for this
        parameter.


    bufferSize

        Variable Type       ViInt32

        Pass the number of bytes in the ViChar array you specify for the
        Description parameter.

        If the error description, including the terminating NUL byte,
        contains more bytes than you indicate in this parameter, the function
        copies BufferSize - 1 bytes into the buffer, places an ASCII NUL byte
        at the end of the buffer, and returns the buffer size you must pass
        to get the entire value.  For example, if the value is "123456" and
        the Buffer Size is 4, the function places "123" into the buffer and
        returns 7.

        If you pass a negative number, the function copies the value to the
        buffer regardless of the number of bytes in the value.

        If you pass 0, you can pass VI_NULL for the Description buffer
        parameter.

        Default Value:  None

    description

        Variable Type       ViChar[]
```

```
Returns the error description for the IVI session or execution
thread.  If there is no description, the function returns an empty
string.
```

The buffer must contain at least as many elements as the value you specify with the Buffer Size parameter. If the error description, including the terminating NUL byte, contains more bytes than you indicate with the Buffer Size parameter, the function copies Buffer Size - 1 bytes into the buffer, places an ASCII NUL byte at the end of the buffer, and returns the buffer size you must pass to get the entire value. For example, if the value is "123456" and the Buffer Size is 4, the function places "123" into the buffer and returns 7.

If you pass 0 for the Buffer Size, you can pass VI_NULL for this parameter.

Return Value

Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You examine the status code from each call to an instrument driver function to determine if an error occurred.

To obtain a text description of the status code, call the bu6100_error_message function. To obtain additional information about the error condition, call the bu6100_GetError function. To clear the error information from the driver, call the bu6100_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

This instrument driver returns errors and warnings defined by other sources. The following table defines the ranges of additional status codes that this driver can return. The table lists the different include files that contain the defined constants for the particular status codes:

Numeric Range (in Hex)	Status Code Types	
3FFA0000 to 3FFA1FFF	IVI	Warnings
3FFF0000 to 3FFFFFFF	VISA	Warnings
3FFC0000 to 3FFCFFFF	VXIPnP	Driver Warnings
BFFA0000 to BFFA1FFF	IVI	Errors
BFFF0000 to BFFFFFFF	VISA	Errors
BFFC0000 to BFFCFFFF	VXIPnP	Driver Errors

5.5.40. bu6100_getFcCclk

```
ViStatus bu6100_getFcCclk (ViSession instrumentHandle,
                           ViInt16 functionCard, ViPInt32 CCLKSource);
```

Purpose

This function returns the source of the Common Clock (CCLK) for the given Function Card.

Parameter List

instrumentHandle

Variable Type ViSession

The Instrument Handle is used to identify the unique session or communication channel between the driver and the instrument.

If more than one instrument of the same model type is used, this Handle will be used to differentiate between them.

functionCard

Variable Type ViInt16

The function card to access.
Valid Values: 1, 2, 3, 4.

CCLKSource

Variable Type ViInt32 (passed by reference)

Returnss the source for the Function Card Common Clock. Possible values are:

```
BU6100_CCLK_0           0  CCLK Disabled and forced to '0'
BU6100_CCLK_1           1  CCLK Disabled and forced to '1'
BU6100_CCLK_10          2  CCLK Connected to CLK10
BU6100_CCLK_5           3  CCLK Connected to CLK10/2
BU6100_CCLK_2           4  CCLK Connected to CLK10/5
BU6100_CCLK_1588_PPP    5  CCLK Connected to IEEE1588_PPP

BU6100_CCLK_LXI_TRG_0   8  CCLK Connected to LXI Trig 0
BU6100_CCLK_LXI_TRG_1   9  CCLK Connected to LXI Trig 1
BU6100_CCLK_LXI_TRG_2   10 CCLK Connected to LXI Trig 2
BU6100_CCLK_LXI_TRG_3   11 CCLK Connected to LXI Trig 3
BU6100_CCLK_LXI_TRG_4   12 CCLK Connected to LXI Trig 4
BU6100_CCLK_LXI_TRG_5   13 CCLK Connected to LXI Trig 5
BU6100_CCLK_LXI_TRG_6   14 CCLK Connected to LXI Trig 6
BU6100_CCLK_LXI_TRG_7   15 CCLK Connected to LXI Trig 7
```

Return Value

Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You examine the status code from each call to an instrument driver function to determine if an error occurred.

To obtain a text description of the status code, call the bu6100_error_message function. To obtain additional information about the error condition, call the bu6100_GetError function. To clear the error information from the driver, call the bu6100_ClearError function.

The general meaning of the status code is as follows:

```
Value                  Meaning
-------------------------------
0                      Success
Positive Values        Warnings
Negative Values        Errors
```

```
This instrument driver returns errors and warnings defined by other
sources.  The following table defines the ranges of additional status
codes that this driver can return.  The table lists the different
include files that contain the defined constants for the particular
status codes:

Numeric Range (in Hex)   Status Code Types
-------------------------------------------------
3FFA0000 to 3FFA1FFF     IVI      Warnings
3FFF0000 to 3FFFFFFF     VISA     Warnings
3FFC0000 to 3FFCFFFF     VXIPnP   Driver Warnings

BFFA0000 to BFFA1FFF     IVI      Errors
BFFF0000 to BFFFFFFF     VISA     Errors
BFFC0000 to BFFCFFFF     VXIPnP   Driver Errors
```

5.5.41. bu6100_GetNextCoercionRecord

```
ViStatus bu6100_GetNextCoercionRecord (ViSession instrumentHandle,
                                       ViInt32 bufferSize,
                                       ViChar _VI_FAR coercionRecord[]);
```

Purpose

This function returns the coercion information associated with the IVI session. This function retrieves and clears the oldest instance in which the instrument driver coerced a value you specified to another value.

If you set the BU6100_ATTR_RECORD_COERCIONS attribute to VI_TRUE, the instrument driver keeps a list of all coercions it makes on ViInt32 or ViReal64 values you pass to instrument driver functions. You use this function to retrieve information from that list.

If the next coercion record string, including the terminating NUL byte, contains more bytes than you indicate in this parameter, the function copies Buffer Size - 1 bytes into the buffer, places an ASCII NUL byte at the end of the buffer, and returns the buffer size you must pass to get the entire value. For example, if the value is "123456" and the Buffer Size is 4, the function places "123" into the buffer and returns 7.

If you pass a negative number, the function copies the value to the buffer regardless of the number of bytes in the value.

If you pass 0, you can pass VI_NULL for the Coercion Record buffer parameter.

The function returns an empty string in the Coercion Record parameter if no coercion records remain for the session.

Parameter List

instrumentHandle

Variable Type	ViSession

The ViSession handle that you obtain from the bu6100_init function. The handle identifies a particular instrument session.

Default Value: None

bufferSize

Variable Type	ViInt32

Pass the number of bytes in the ViChar array you specify for the Coercion Record parameter.

If the next coercion record string, including the terminating NUL byte, contains more bytes than you indicate in this parameter, the function copies Buffer Size - 1 bytes into the buffer, places an ASCII NUL byte at the end of the buffer, and returns the buffer size you must pass to get the entire value. For example, if the value is "123456" and the Buffer Size is 4, the function places "123" into the buffer and returns 7.

If you pass a negative number, the function copies the value to the buffer regardless of the number of bytes in the value.

If you pass 0, you can pass VI_NULL for the Coercion Record buffer parameter.

Default Value: None

coercionRecord

Variable Type	ViChar[]

Returns the next coercion record for the IVI session. If there are

```
        no coercion records, the function returns an empty string.

        The buffer must contain at least as many elements as the value you
        specify with the Buffer Size parameter.  If the next coercion record
        string, including the terminating NUL byte, contains more bytes than
        you indicate with the Buffer Size parameter, the function copies
        Buffer Size - 1 bytes into the buffer, places an ASCII NUL byte at
        the end of the buffer, and returns the buffer size you must pass to
        get the entire value.  For example, if the value is "123456" and the
        Buffer Size is 4, the function places "123" into the buffer and
        returns 7.

        This parameter returns an empty string if no coercion records remain
        for the session.


Return Value

        Returns the status code of this operation.  The status code  either
        indicates success or describes an error or warning condition.  You
        examine the status code from each call to an instrument driver
        function to determine if an error occurred.

        To obtain a text description of the status code, call the
        bu6100_error_message function.  To obtain additional information
        about the error condition, call the bu6100_GetError function.  To
        clear the error information from the driver, call the
        bu6100_ClearError function.

        The general meaning of the status code is as follows:

        Value                  Meaning
        -------------------------------
        0                      Success
        Positive Values        Warnings
        Negative Values        Errors

        This instrument driver returns errors and warnings defined by other
        sources.  The following table defines the ranges of additional status
        codes that this driver can return.  The table lists the different
        include files that contain the defined constants for the particular
        status codes:

        Numeric Range (in Hex)   Status Code Types
        -------------------------------------------------
        3FFA0000 to 3FFA1FFF     IVI      Warnings
        3FFF0000 to 3FFFFFFF     VISA     Warnings
        3FFC0000 to 3FFCFFFF     VXIPnP   Driver Warnings

        BFFA0000 to BFFA1FFF     IVI      Errors
        BFFF0000 to BFFFFFFF     VISA     Errors
        BFFC0000 to BFFCFFFF     VXIPnP   Driver Errors
```

5.5.42. bu6100_GetNextInterchangeWarning

```
    ViStatus bu6100_GetNextInterchangeWarning (ViSession instrumentHandle,
                                               ViInt32 bufferSize,
                                               ViChar _VI_FAR interchangeWarning[]);
```

Purpose

This function returns the interchangeability warnings associated with the IVI session. It retrieves and clears the oldest instance in which the class driver recorded an interchangeability warning. Interchangeability warnings indicate that using your application with a different instrument might cause different behavior. You use this function to retrieve interchangeability warnings.

The driver performs interchangeability checking when the BU6100_ATTR_INTERCHANGE_CHECK attribute is set to VI_TRUE.

The function returns an empty string in the Interchange Warning parameter if no interchangeability warnings remain for the session.

In general, the instrument driver generates interchangeability warnings when an attribute that affects the behavior of the instrument is in a state that you did not specify.

Parameter List

instrumentHandle

Variable Type	ViSession

The ViSession handle that you obtain from the bu6100_init or bu6100_InitWithOptions function. The handle identifies a particular instrument session.

Default Value: None

bufferSize

Variable Type	ViInt32

Pass the number of bytes in the ViChar array you specify for the Interchange Warning parameter.

If the next interchangeability warning string, including the terminating NUL byte, contains more bytes than you indicate in this parameter, the function copies Buffer Size - 1 bytes into the buffer, places an ASCII NUL byte at the end of the buffer, and returns the buffer size you must pass to get the entire value. For example, if the value is "123456" and the Buffer Size is 4, the function places "123" into the buffer and returns 7.

If you pass a negative number, the function copies the value to the buffer regardless of the number of bytes in the value.

If you pass 0, you can pass VI_NULL for the Interchange Warning buffer parameter.

Default Value: None

interchangeWarning

Variable Type	ViChar[]

Returns the next interchange warning for the IVI session. If there are no interchange warnings, the function returns an empty string.

The buffer must contain at least as many elements as the value you specify with the Buffer Size parameter. If the next interchangeability warning string, including the terminating NUL byte, contains more bytes than you indicate with the Buffer Size parameter, the function copies Buffer Size - 1 bytes into the buffer, places an ASCII NUL byte at the end of the buffer, and returns the

```
buffer size you must pass to get the entire value.  For example, if
the value is "123456" and the Buffer Size is 4, the function places
"123" into the buffer and returns 7.
```

This parameter returns an empty string if no interchangeability warnings remain for the session.

Return Value

Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You examine the status code from each call to an instrument driver function to determine if an error occurred.

To obtain a text description of the status code, call the bu6100_error_message function. To obtain additional information about the error condition, call the bu6100_GetError function. To clear the error information from the driver, call the bu6100_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

This instrument driver returns errors and warnings defined by other sources. The following table defines the ranges of additional status codes that this driver can return. The table lists the different include files that contain the defined constants for the particular status codes:

Numeric Range (in Hex)	Status Code Types	
3FFA0000 to 3FFA1FFF	IVI	Warnings
3FFF0000 to 3FFFFFFF	VISA	Warnings
3FFC0000 to 3FFCFFFF	VXIPnP	Driver Warnings
BFFA0000 to BFFA1FFF	IVI	Errors
BFFF0000 to BFFFFFFF	VISA	Errors
BFFC0000 to BFFCFFFF	VXIPnP	Driver Errors

5.5.43. bu6100_getTrigStatus

```
ViStatus bu6100_getTrigStatus (ViSession instrumentHandle,
                               ViPInt32 sourceTriggerStatus,
                               ViPInt32 triggerNodeStatus);
```

```
Purpose

    Reads the current status of all Trigger lines on ProDAQ 6100 module.

Parameter List

    instrumentHandle

        Variable Type       ViSession

        The Instrument Handle is used to identify the unique session or
        communication channel between the driver and the instrument.

        If more than one instrument of the same model type is used, this
        Handle will be used to differentiate between them.


    sourceTriggerStatus

        Variable Type       ViInt32 (passed by reference)

        Contains the current status of all source trigger lines of ProDAQ
        6100 module. Return value is a bitmask:

        Bit No. Name
          0     FC 1 Trigger Out A
          1     FC 2 Trigger Out A
          2     FC 3 Trigger Out A
          3     FC 4 Trigger Out A

          8     FC 1 Trigger Out B
          9     FC 2 Trigger Out B
         10     FC 3 Trigger Out B
         11     FC 4 Trigger Out B

         16     LXI Trigger 0
         17     LXI Trigger 1
         18     LXI Trigger 2
         19     LXI Trigger 3
         20     LXI Trigger 4
         21     LXI Trigger 5
         22     LXI Trigger 6
         23     LXI Trigger 7

         29     IEEE1588 Trigger
         30     IEEE1588 Pulse Per Period

        "1" means that trigger line is active (Logical level low);
        "0" means that trigger line is inactive (Logical level high);


    triggerNodeStatus

        Variable Type       ViInt32 (passed by reference)

        Contains the current status of all trigger nodes of ProDAQ 6100
        module. Return value is a bitmask:

        Bit No. Name
          0     FC 1 Trigger In A
          1     FC 2 Trigger In A
          2     FC 3 Trigger In A
          3     FC 4 Trigger In A

          8     FC 1 Trigger In B
          9     FC 2 Trigger In B
         10     FC 3 Trigger In B
         11     FC 4 Trigger In B
         16     LXI Trigger 0
         17     LXI Trigger 1
         18     LXI Trigger 2
         19     LXI Trigger 3
         20     LXI Trigger 4
         21     LXI Trigger 5
         22     LXI Trigger 6
         23     LXI Trigger 7

        "1" means that trigger node is active (Logical level low);
        "0" means that trigger node is inactive (Logical level high);


Return Value

        Returns the status code of this operation.  The status code  either
        indicates success or describes an error or warning condition.  You
        examine the status code from each call to an instrument driver
        function to determine if an error occurred.

        To obtain a text description of the status code, call the
        bu6100_error_message function.  To obtain additional information
        about the error condition, call the bu6100_GetError function.  To
        clear the error information from the driver, call the
        bu6100_ClearError function.

        The general meaning of the status code is as follows:

        Value                  Meaning
        -------------------------------
        0                      Success
        Positive Values        Warnings
        Negative Values        Errors

        This instrument driver returns errors and warnings defined by other
        sources.  The following table defines the ranges of additional status
        codes that this driver can return.  The table lists the different
        include files that contain the defined constants for the particular
        status codes:

        Numeric Range (in Hex)   Status Code Types
        -------------------------------------------------
        3FFA0000 to 3FFA1FFF     IVI      Warnings
        3FFF0000 to 3FFFFFFF     VISA     Warnings
        3FFC0000 to 3FFCFFFF     VXIPnP   Driver Warnings

        BFFA0000 to BFFA1FFF     IVI      Errors
        BFFF0000 to BFFFFFFF     VISA     Errors
        BFFC0000 to BFFCFFFF     VXIPnP   Driver Errors
```

5.5.44. bu6100_getVoltRefInfo

```
    ViStatus bu6100_getVoltRefInfo (ViSession instrumentHandle,
                                    ViPInt32 nVoltages,
                                    ViReal64 _VI_FAR voltages[]);

  Purpose

    Returns the list of all possible voltages generated by the voltage
    reference module.

  Parameter List

    instrumentHandle

        Variable Type       ViSession

        The Instrument Handle is used to identify the unique session or
        communication channel between the driver and the instrument.

        If more than one instrument of the same model type is used, this
        Handle will be used to differentiate between them.


    nVoltages

        Variable Type       ViInt32 (passed by reference)

        This parameter return the number of elements returned in "Voltages"
        output.

    voltages

        Variable Type       ViReal64[]

        This array contains the list of all possible voltages generated by
        the voltage reference module. It should be allocated with size 20
        prior to the function call.

  Return Value

        Returns the status code of this operation.  The status code  either
        indicates success or describes an error or warning condition.  You
        examine the status code from each call to an instrument driver
        function to determine if an error occurred.

        To obtain a text description of the status code, call the
        bu6100_error_message function.  To obtain additional information
        about the error condition, call the bu6100_GetError function.  To
        clear the error information from the driver, call the
        bu6100_ClearError function.

        The general meaning of the status code is as follows:

        Value                  Meaning
        -------------------------------
        0                      Success
        Positive Values        Warnings
        Negative Values        Errors

        This instrument driver returns errors and warnings defined by other
        sources.  The following table defines the ranges of additional status
        codes that this driver can return.  The table lists the different
        include files that contain the defined constants for the particular
        status codes:

        Numeric Range (in Hex)   Status Code Types
        -------------------------------------------------
        3FFA0000 to 3FFA1FFF     IVI      Warnings
        3FFF0000 to 3FFFFFFF     VISA     Warnings
        3FFC0000 to 3FFCFFFF     VXIPnP   Driver Warnings

        BFFA0000 to BFFA1FFF     IVI      Errors
        BFFF0000 to BFFFFFFF     VISA     Errors
        BFFC0000 to BFFCFFFF     VXIPnP   Driver Errors
```

5.5.45. bu6100_getVoltRefOutput

```
ViStatus bu6100_getVoltRefOutput (ViSession instrumentHandle,
                                  ViPReal64 voltage);
```

```
Purpose

    Returns the current output voltage of Voltage Reference module.

Parameter List

    instrumentHandle

        Variable Type       ViSession

        The Instrument Handle is used to identify the unique session or
        communication channel between the driver and the instrument.

        If more than one instrument of the same model type is used, this
        Handle will be used to differentiate between them.


    voltage

        Variable Type       ViReal64 (passed by reference)

        Returns the current output voltage of Voltage Reference module.

Return Value

        Returns the status code of this operation.  The status code  either
        indicates success or describes an error or warning condition.  You
        examine the status code from each call to an instrument driver
        function to determine if an error occurred.

        To obtain a text description of the status code, call the
        bu6100_error_message function.  To obtain additional information
        about the error condition, call the bu6100_GetError function.  To
        clear the error information from the driver, call the
        bu6100_ClearError function.

        The general meaning of the status code is as follows:

        Value                  Meaning
        -------------------------------
        0                      Success
        Positive Values        Warnings
        Negative Values        Errors

        This instrument driver returns errors and warnings defined by other
        sources.  The following table defines the ranges of additional status
        codes that this driver can return.  The table lists the different
        include files that contain the defined constants for the particular
        status codes:

        Numeric Range (in Hex)   Status Code Types
        -------------------------------------------------
        3FFA0000 to 3FFA1FFF     IVI      Warnings
        3FFF0000 to 3FFFFFFF     VISA     Warnings
        3FFC0000 to 3FFCFFFF     VXIPnP   Driver Warnings

        BFFA0000 to BFFA1FFF     IVI      Errors
        BFFF0000 to BFFFFFFF     VISA     Errors
        BFFC0000 to BFFCFFFF     VXIPnP   Driver Errors
```

5.5.46. bu6100_identify

```
    ViStatus bu6100_identify (ViSession instrumentHandle, ViBoolean enable);

Purpose

    This function enables/disables blinking of LAN LED on the front panel of
    the instrument. Using this LED can help in identifying the LXI
    instrument.

Parameter List

    instrumentHandle

        Variable Type       ViSession

        The Instrument Handle is used to identify the unique session or
        communication channel between the driver and the instrument.

        If more than one instrument of the same model type is used, this
        Handle will be used to differentiate between them.


    enable

        Variable Type       ViBoolean

        This parameter specifies whether the LAN LED on the front panel of
        the instrument is blinking or not.

        Valid Values:

        VI_FALSE  0  The LAN LED is not blinking;
        VI_TRUE   1  The LAN LED is blinking;

        Default Value: VI_FALSE

Return Value

        Returns the status code of this operation.  The status code  either
        indicates success or describes an error or warning condition.  You
        examine the status code from each call to an instrument driver
        function to determine if an error occurred.

        To obtain a text description of the status code, call the
        bu6100_error_message function.  To obtain additional information
        about the error condition, call the bu6100_GetError function.  To
        clear the error information from the driver, call the
        bu6100_ClearError function.

        The general meaning of the status code is as follows:

        Value                  Meaning
        -------------------------------
        0                      Success
        Positive Values        Warnings
        Negative Values        Errors

        This instrument driver returns errors and warnings defined by other
        sources.  The following table defines the ranges of additional status
        codes that this driver can return.  The table lists the different
        include files that contain the defined constants for the particular
        status codes:

        Numeric Range (in Hex)   Status Code Types
        -------------------------------------------------
        3FFA0000 to 3FFA1FFF     IVI      Warnings
        3FFF0000 to 3FFFFFFF     VISA     Warnings
        3FFC0000 to 3FFCFFFF     VXIPnP   Driver Warnings

        BFFA0000 to BFFA1FFF     IVI      Errors
        BFFF0000 to BFFFFFFF     VISA     Errors
        BFFC0000 to BFFCFFFF     VXIPnP   Driver Errors
```

5.5.47. bu6100_init

```
     ViStatus bu6100_init (ViRsrc resourceName, ViBoolean IDQuery,
                           ViBoolean resetDevice,
                           ViPSession instrumentHandle);

 Purpose

     This function performs the following initialization actions:

     - Creates a new IVI instrument driver session.

     - Opens a session to the specified device using the interface and address
     you specify for the Resource Name parameter.

     - If the ID Query parameter is set to VI_TRUE, this function queries the
     instrument ID and checks that it is valid for this instrument driver.

     - If the Reset parameter is set to VI_TRUE, this function resets the
     instrument to a known state.

     - Sends initialization commands to set the instrument to the state
     necessary for the operation of the instrument driver.

     - Returns a ViSession handle that you use to identify the instrument in
     all subsequent instrument driver function calls.

     Note:  This function creates a new session each time you invoke it.
     Although you can open more than one IVI session for the same resource, it
     is best not to do so.  You can use the same session in multiple program
     threads.  You can use the bu6100_LockSession and bu6100_UnlockSession
     functions to protect sections of code that require exclusive access to
     the resource.



 Parameter List

     resourceName

         Variable Type       ViRsrc

         Pass the resource name of the device to initialize.

         You can also pass the name of a driver session or logical name that
         you configure with the IVI Configuration utility.  The driver session
         identifies a specific device and specifies the initial settings for
         the session.  A logical Name identifies a particular driver session.

         Refer to the following table below for the exact grammar to use for
         this parameter.  Optional fields are shown in square brackets ([]).

         Syntax
         ------------------------------------------------------
         GPIB[board]::<primary address>[::secondary address]::INSTR
         VXI[board]::<logical address>::INSTR
         GPIB-VXI[board]::<logical address>::INSTR
         ASRL<port>::INSTR
         <LogicalName>
         <Driver Session>

         If you do not specify a value for an optional field, the following
         values are used:

         Optional Field - Value
         ------------------------------------------------------
         board - 0
         secondary address - none (31)

         The following table contains example valid values for this parameter.

         "Valid Value" - Description
         ------------------------------------------------------
         "GPIB::22::INSTR" - GPIB board 0, primary address 22 no
                             secondary address
```

```
    "GPIB::22::5::INSTR" - GPIB board 0, primary address 22
                           secondary address 5
    "GPIB1::22::5::INSTR" - GPIB board 1, primary address 22
                            secondary address 5
    "VXI::64::INSTR" - VXI board 0, logical address 64
    "VXI1::64::INSTR" - VXI board 1, logical address 64
    "GPIB-VXI::64::INSTR" - GPIB-VXI board 0, logical address 64
    "GPIB-VXI1::64::INSTR" - GPIB-VXI board 1, logical address 64
    "ASRL2::INSTR" - COM port 2
    "SampleInstr" - Logical name "SampleInstr"
    "xyz432" - Logical Name or Driver Session "xyz432"

    Default Value:  "TCPIP0::192.168.168.63::inst0::INSTR"



IDQuery

    Variable Type       ViBoolean

    Specify whether you want the instrument driver to perform an ID
    Query.

    Valid Range:
    VI_TRUE  (1) - Perform ID Query (Default Value)
    VI_FALSE (0) - Skip ID Query

    When you set this parameter to VI_TRUE, the driver verifies that the
    instrument you initialize is a type that this driver supports.

    Circumstances can arise where it is undesirable to send an ID Query
    command string to the instrument.  When you set this parameter to
    VI_FALSE, the function initializes the instrument without performing
    an ID Query.

resetDevice

    Variable Type       ViBoolean

    Specify whether you want the to reset the instrument during the
    initialization procedure.

    Valid Range:
    VI_TRUE  (1) - Reset Device (Default Value)
    VI_FALSE (0) - Don't Reset



instrumentHandle

    Variable Type       ViSession (passed by reference)

    Returns a ViSession handle that you use to identify the instrument in
    all subsequent instrument driver function calls.

    Notes:

    (1) This function creates a new session each time you invoke it.
    This is useful if you have multiple physical instances of the same
    type of instrument.

    (2) Avoid creating multiple concurrent sessions to the same physical
    instrument.  Although you can create more than one IVI session for
    the same resource, it is best not to do so.  A better approach is to
    use the same IVI session in multiple execution threads.  You can use
    functions bu6100_LockSession and bu6100_UnlockSession to protect
    sections of code that require exclusive access to the resource.



Return Value

    Returns the status code of this operation.  The status code  either
    indicates success or describes an error or warning condition.  You
    examine the status code from each call to an instrument driver
    function to determine if an error occurred.
```

```
To obtain a text description of the status code, call the
bu6100_error_message function.  To obtain additional information
about the error condition, call the bu6100_GetError function.  To
clear the error information from the driver, call the
bu6100_ClearError function.
```

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

This instrument driver returns errors and warnings defined by other sources. The following table defines the ranges of additional status codes that this driver can return. The table lists the different include files that contain the defined constants for the particular status codes:

Numeric Range (in Hex)	Status Code Types	
3FFA0000 to 3FFA1FFF	IVI	Warnings
3FFF0000 to 3FFFFFFF	VISA	Warnings
3FFC0000 to 3FFCFFFF	VXIPnP	Driver Warnings
BFFA0000 to BFFA1FFF	IVI	Errors
BFFF0000 to BFFFFFFF	VISA	Errors
BFFC0000 to BFFCFFFF	VXIPnP	Driver Errors

5.5.48. bu6100_InitWithOptions

```
    ViStatus bu6100_InitWithOptions (ViRsrc resourceName, ViBoolean IDQuery,
                                     ViBoolean resetDevice,
                                     ViChar _VI_FAR optionString[],
                                     ViPSession instrumentHandle);

Purpose

    This function performs the following initialization actions:

    - Creates a new IVI instrument driver and optionally sets the initial
    state of the following session attributes:

        BU6100_ATTR_RANGE_CHECK
        BU6100_ATTR_QUERY_INSTRUMENT_STATUS
        BU6100_ATTR_CACHE
        BU6100_ATTR_SIMULATE
        BU6100_ATTR_RECORD_COERCIONS

    - Opens a session to the specified device using the interface and address
    you specify for the Resource Name parameter.

    - If the ID Query parameter is set to VI_TRUE, this function queries the
    instrument ID and checks that it is valid for this instrument driver.

    - If the Reset parameter is set to VI_TRUE, this function resets the
    instrument to a known state.

    - Sends initialization commands to set the instrument to the state
    necessary for the operation of the instrument driver.

    - Returns a ViSession handle that you use to identify the instrument in
    all subsequent instrument driver function calls.

    Note:  This function creates a new session each time you invoke it.
    Although you can open more than one IVI session for the same resource, it
    is best not to do so.  You can use the same session in multiple program
    threads.  You can use the bu6100_LockSession and bu6100_UnlockSession
    functions to protect sections of code that require exclusive access to
    the resource.



Parameter List

    resourceName

        Variable Type       ViRsrc

        Pass the resource name of the device to initialize.

        You can also pass the name of a driver session or logical name that
        you configure with the IVI Configuration utility.  The driver session
        identifies a specific device and specifies the initial settings for
        the session.  A logical Name identifies a particular driver session.

        Refer to the following table below for the exact grammar to use for
        this parameter.  Optional fields are shown in square brackets ([]).

        Syntax
        ------------------------------------------------------
        GPIB[board]::<primary address>[::secondary address]::INSTR
        VXI[board]::<logical address>::INSTR
        GPIB-VXI[board]::<logical address>::INSTR
        ASRL<port>::INSTR
        <LogicalName>
        <Driver Session>

        If you do not specify a value for an optional field, the following
        values are used:

        Optional Field - Value
        ------------------------------------------------------
        board - 0
        secondary address - none (31)

        The following table contains example valid values for this parameter.

        "Valid Value" - Description
        ------------------------------------------------------
        "GPIB::22::INSTR" - GPIB board 0, primary address 22 no
                            secondary address
        "GPIB::22::5::INSTR" - GPIB board 0, primary address 22
                               secondary address 5
        "GPIB1::22::5::INSTR" - GPIB board 1, primary address 22
                                secondary address 5
        "VXI::64::INSTR" - VXI board 0, logical address 64
        "VXI1::64::INSTR" - VXI board 1, logical address 64
        "GPIB-VXI::64::INSTR" - GPIB-VXI board 0, logical address 64
        "GPIB-VXI1::64::INSTR" - GPIB-VXI board 1, logical address 64
        "ASRL2::INSTR" - COM port 2
        "SampleInstr" - Logical name "SampleInstr"
        "xyz432" - Logical Name or Driver Session "xyz432"

        Default Value:  "TCPIP0::192.168.168.63::inst0::INSTR"




    IDQuery

        Variable Type       ViBoolean

        Specify whether you want the instrument driver to perform an ID
        Query.

        Valid Range:
        VI_TRUE  (1) - Perform ID Query (Default Value)
        VI_FALSE (0) - Skip ID Query

        When you set this parameter to VI_TRUE, the driver verifies that the
        instrument you initialize is a type that this driver supports.

        Circumstances can arise where it is undesirable to send an ID Query
        command string to the instrument.  When you set this parameter to
        VI_FALSE, the function initializes the instrument without performing
        an ID Query.

    resetDevice

        Variable Type       ViBoolean

        Specify whether you want the to reset the instrument during the
        initialization procedure.

        Valid Range:
        VI_TRUE  (1) - Reset Device (Default Value)
        VI_FALSE (0) - Don't Reset



    optionString

        Variable Type       ViChar[]

        You can use this control to set the initial value of certain
        attributes for the session.  The following table lists the attributes
        and the name you use in this parameter to identify the attribute.

        Name              Attribute Defined Constant
        --------------------------------------------
        RangeCheck        BU6100_ATTR_RANGE_CHECK
        QueryInstrStatus  BU6100_ATTR_QUERY_INSTRUMENT_STATUS
        Cache             BU6100_ATTR_CACHE
        Simulate          BU6100_ATTR_SIMULATE
        RecordCoercions   BU6100_ATTR_RECORD_COERCIONS

        The format of this string is, "AttributeName=Value" where
        AttributeName is the name of the attribute and Value is the value to
        which the attribute will be set.  To set multiple attributes,
        separate their assignments with a comma.
```

```
If you pass NULL or an empty string for this parameter, the session
uses the default values for the attributes.   You can override the
default values by assigning a value explicitly in a string you pass
for this parameter.  You do not have to specify all of the attributes
and may leave any of them out.  If you do not specify one of the
attributes, its default value will be used.
```

```
        The default values for the attributes are shown below:

            Attribute Name     Default Value
            ----------------   -------------
            RangeCheck         VI_TRUE
            QueryInstrStatus   VI_FALSE
            Cache              VI_TRUE
            Simulate           VI_FALSE
            RecordCoercions    VI_FALSE


        The following are the valid values for ViBoolean attributes:

            True:     1, True, or VI_TRUE
            False:    0, False, or VI_FALSE


        Default Value:
               "Simulate=0,RangeCheck=1,QueryInstrStatus=0,Cache=1"


    instrumentHandle

        Variable Type       ViSession (passed by reference)

        Returns a ViSession handle that you use to identify the instrument in
        all subsequent instrument driver function calls.

        Notes:

        (1) This function creates a new session each time you invoke it.
        This is useful if you have multiple physical instances of the same
        type of instrument.

        (2) Avoid creating multiple concurrent sessions to the same physical
        instrument.  Although you can create more than one IVI session for
        the same resource, it is best not to do so.  A better approach is to
        use the same IVI session in multiple execution threads.  You can use
        functions bu6100_LockSession and bu6100_UnlockSession to protect
        sections of code that require exclusive access to the resource.



Return Value

        Returns the status code of this operation.  The status code  either
        indicates success or describes an error or warning condition.  You
        examine the status code from each call to an instrument driver
        function to determine if an error occurred.

        To obtain a text description of the status code, call the
        bu6100_error_message function.  To obtain additional information
        about the error condition, call the bu6100_GetError function.  To
        clear the error information from the driver, call the
        bu6100_ClearError function.

        The general meaning of the status code is as follows:

        Value                  Meaning
        -------------------------------
        0                      Success
        Positive Values        Warnings
        Negative Values        Errors

        This instrument driver returns errors and warnings defined by other
        sources.  The following table defines the ranges of additional status
        codes that this driver can return.  The table lists the different
        include files that contain the defined constants for the particular
        status codes:

        Numeric Range (in Hex)   Status Code Types
        -------------------------------------------------
        3FFA0000 to 3FFA1FFF     IVI      Warnings
        3FFF0000 to 3FFFFFFF     VISA     Warnings
        3FFC0000 to 3FFCFFFF     VXIPnP   Driver Warnings

        BFFA0000 to BFFA1FFF     IVI      Errors
```

```
BFFF0000 to BFFFFFFF     VISA     Errors
BFFC0000 to BFFCFFFF     VXIPnP   Driver Errors
```

5.5.49. bu6100_installIrqHandler

```
ViStatus bu6100_installIrqHandler (ViSession instrumentHandle,
                                   ViInt16 source,
                                   ViAddr interruptHandler,
                                   ViAddr interruptParameter);
```

```
Purpose

    Installs Interrupt Service Routine for the Function Card.
    This Interrupt Service Routine will be called every time when
    the Output Trigger line of the Function Card goes to active (low) state.
    The Output Trigger line of the Function Card should be configured for
    low-active polarity (where selectable) and "level" type of output (where
    selectable).
    Upon completion, the Interrupt Service Routine should clear the
    Output Trigger line of the Function Card by method depending on the type
    of Function Card (for instance, read out FIFO or clear some bits in
    Function Card control register)

Parameter List

    instrumentHandle

        Variable Type       ViSession

        The Instrument Handle is used to identify the unique session or
        communication channel between the driver and the instrument.

        If more than one instrument of the same model type is used, this
        Handle will be used to differentiate between them.


    source

        Variable Type       ViInt16

        The trigger line for which Interrupt Handler will be installed.


        Valid Values:
                      bu3100_FCTrigOutA1    0   FC 1 Trigger Output A
                      bu3100_FCTrigOutA2    1   FC 2 Trigger Output A
                      bu3100_FCTrigOutA3    2   FC 3 Trigger Output A
                      bu3100_FCTrigOutA4    3   FC 4 Trigger Output A

                      bu3100_FCTrigOutB1    8   FC 1 Trigger Output B
                      bu3100_FCTrigOutB2    9   FC 2 Trigger Output B
                      bu3100_FCTrigOutB3    10  FC 3 Trigger Output B
                      bu3100_FCTrigOutB4    11  FC 4 Trigger Output B

                      bu3100_FCCircBuf1     16  FC 1 Circular Buffer
                      bu3100_FCCircBuf2     17  FC 2 Circular Buffer
                      bu3100_FCCircBuf3     18  FC 3 Circular Buffer
                      bu3100_FCCircBuf4     19  FC 4 Circular Buffer

    interruptHandler

        Variable Type       ViAddr

        Pointer to the interrupt service routine.
        Default Value: VI_NULL

    interruptParameter

        Variable Type       ViAddr

        Pointer to any user-defined data, which will be transferred to the
        Interrupt Service routine in every call.
        Default Value: VI_NULL

Return Value

        Returns the status code of this operation.  The status code  either
        indicates success or describes an error or warning condition.  You
        examine the status code from each call to an instrument driver
```

```
function to determine if an error occurred.
```

```
To obtain a text description of the status code, call the
bu6100_error_message function.  To obtain additional information
about the error condition, call the bu6100_GetError function.  To
clear the error information from the driver, call the
bu6100_ClearError function.

The general meaning of the status code is as follows:

Value                  Meaning
-------------------------------
0                      Success
Positive Values        Warnings
Negative Values        Errors

This instrument driver returns errors and warnings defined by other
sources.  The following table defines the ranges of additional status
codes that this driver can return.  The table lists the different
include files that contain the defined constants for the particular
status codes:

Numeric Range (in Hex)   Status Code Types
-------------------------------------------------
3FFA0000 to 3FFA1FFF     IVI      Warnings
3FFF0000 to 3FFFFFFF     VISA     Warnings
3FFC0000 to 3FFCFFFF     VXIPnP   Driver Warnings

BFFA0000 to BFFA1FFF     IVI      Errors
BFFF0000 to BFFFFFFF     VISA     Errors
BFFC0000 to BFFCFFFF     VXIPnP   Driver Errors
```

5.5.50. bu6100_installIrqWatcher

```
ViStatus bu6100_installIrqWatcher (ViSession instrumentHandle,
                                   ViInt16 source);
```

Purpose

Creates the IRQ watcher for selected Function Card. After IRQ watcher is installed the function bu6100_waitIrqWatcher can be used to catch an interrupt events from given Function Card or from multiple Function Cards at the same time respectively.
This mechanism can be used whenever asynchronous Interrupt Handling is not applicable (for instance, LabVIEW environment).

Parameter List

instrumentHandle

Variable Type ViSession

The Instrument Handle is used to identify the unique session or communication channel between the driver and the instrument.

If more than one instrument of the same model type is used, this Handle will be used to differentiate between them.

source

Variable Type ViInt16

The trigger line for which Interrupt Watcher will be installed.

Valid Values:

bu3100_FCTrigOutA1	0	FC 1 Trigger Output A
bu3100_FCTrigOutA2	1	FC 2 Trigger Output A
bu3100_FCTrigOutA3	2	FC 3 Trigger Output A
bu3100_FCTrigOutA4	3	FC 4 Trigger Output A
bu3100_FCTrigOutB1	8	FC 1 Trigger Output B
bu3100_FCTrigOutB2	9	FC 2 Trigger Output B
bu3100_FCTrigOutB3	10	FC 3 Trigger Output B
bu3100_FCTrigOutB4	11	FC 4 Trigger Output B
bu3100_FCCircBuf1	16	FC 1 Circular Buffer
bu3100_FCCircBuf2	17	FC 2 Circular Buffer
bu3100_FCCircBuf3	18	FC 3 Circular Buffer
bu3100_FCCircBuf4	19	FC 4 Circular Buffer

Return Value

Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You examine the status code from each call to an instrument driver function to determine if an error occurred.

To obtain a text description of the status code, call the bu6100_error_message function. To obtain additional information about the error condition, call the bu6100_GetError function. To clear the error information from the driver, call the bu6100_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

This instrument driver returns errors and warnings defined by other sources. The following table defines the ranges of additional status codes that this driver can return. The table lists the different

```
include files that contain the defined constants for the particular
status codes:
```

```
Numeric Range (in Hex)   Status Code Types
-------------------------------------------------
3FFA0000 to 3FFA1FFF     IVI      Warnings
3FFF0000 to 3FFFFFFF     VISA     Warnings
3FFC0000 to 3FFCFFFF     VXIPnP   Driver Warnings

BFFA0000 to BFFA1FFF     IVI      Errors
BFFF0000 to BFFFFFFF     VISA     Errors
BFFC0000 to BFFCFFFF     VXIPnP   Driver Errors
```

5.5.51. bu6100_InvalidateAllAttributes

```
    ViStatus bu6100_InvalidateAllAttributes (ViSession instrumentHandle);

 Purpose

    This function invalidates the cached values of all attributes for the
    session.

 Parameter List

    instrumentHandle

        Variable Type       ViSession

        The ViSession handle that you obtain from the bu6100_init or
        bu6100_InitWithOptions function.  The handle identifies a particular
        instrument session.

        Default Value:  None


 Return Value

        Returns the status code of this operation.  The status code  either
        indicates success or describes an error or warning condition.  You
        examine the status code from each call to an instrument driver
        function to determine if an error occurred.

        To obtain a text description of the status code, call the
        bu6100_error_message function.  To obtain additional information
        about the error condition, call the bu6100_GetError function.  To
        clear the error information from the driver, call the
        bu6100_ClearError function.

        The general meaning of the status code is as follows:

        Value                  Meaning
        -------------------------------
        0                      Success
        Positive Values        Warnings
        Negative Values        Errors

        This instrument driver returns errors and warnings defined by other
        sources.  The following table defines the ranges of additional status
        codes that this driver can return.  The table lists the different
        include files that contain the defined constants for the particular
        status codes:

        Numeric Range (in Hex)   Status Code Types
        -------------------------------------------------
        3FFA0000 to 3FFA1FFF     IVI      Warnings
        3FFF0000 to 3FFFFFFF     VISA     Warnings
        3FFC0000 to 3FFCFFFF     VXIPnP   Driver Warnings

        BFFA0000 to BFFA1FFF     IVI      Errors
        BFFF0000 to BFFFFFFF     VISA     Errors
        BFFC0000 to BFFCFFFF     VXIPnP   Driver Errors
```

5.5.52. bu6100_IviLxiSync_AddArmAlarm

```
ViStatus bu6100_IviLxiSync_AddArmAlarm (ViSession instrumentHandle,
                                        ViConstString alarmName);
```

```
Purpose

    This function creates a new arm alarm

    ProDAQ 6100 does not support Arm subsystem.

Parameter List

    instrumentHandle

        Variable Type       ViSession

        Instrument handle.

    alarmName

        Variable Type       ViConstString

        Specifies the name of the arm alarm to create.

Return Value

        Returns the status code of this operation.  The status code  either
        indicates success or describes an error or warning condition.  You
        examine the status code from each call to an instrument driver
        function to determine if an error occurred.

        To obtain a text description of the status code, call the
        bu6100_error_message function.  To obtain additional information
        about the error condition, call the bu6100_GetError function.  To
        clear the error information from the driver, call the
        bu6100_ClearError function.

        The general meaning of the status code is as follows:

        Value                  Meaning
        -------------------------------
        0                      Success
        Positive Values        Warnings
        Negative Values        Errors

        This instrument driver returns errors and warnings defined by other
        sources.  The following table defines the ranges of additional status
        codes that this driver can return.  The table lists the different
        include files that contain the defined constants for the particular
        status codes:

        Numeric Range (in Hex)   Status Code Types
        -------------------------------------------------
        3FFA0000 to 3FFA1FFF     IVI      Warnings
        3FFF0000 to 3FFFFFFF     VISA     Warnings
        3FFC0000 to 3FFCFFFF     VXIPnP   Driver Warnings

        BFFA0000 to BFFA1FFF     IVI      Errors
        BFFF0000 to BFFFFFFF     VISA     Errors
        BFFC0000 to BFFCFFFF     VXIPnP   Driver Errors
```

5.5.53. bu6100_IviLxiSync_AddArmSource

```
ViStatus bu6100_IviLxiSync_AddArmSource (ViSession instrumentHandle,
                                         ViConstString sourceName);
```

```
Purpose

    This function creates a new arm source.

    ProDAQ 6100 does not support Arm subsystem.

Parameter List

    instrumentHandle

        Variable Type       ViSession

        Instrument handle.

    sourceName

        Variable Type       ViConstString

        Specifies the name of the arm source to create.

Return Value

        Returns the status code of this operation.  The status code  either
        indicates success or describes an error or warning condition.  You
        examine the status code from each call to an instrument driver
        function to determine if an error occurred.

        To obtain a text description of the status code, call the
        bu6100_error_message function.  To obtain additional information
        about the error condition, call the bu6100_GetError function.  To
        clear the error information from the driver, call the
        bu6100_ClearError function.

        The general meaning of the status code is as follows:

        Value                  Meaning
        -------------------------------
        0                      Success
        Positive Values        Warnings
        Negative Values        Errors

        This instrument driver returns errors and warnings defined by other
        sources.  The following table defines the ranges of additional status
        codes that this driver can return.  The table lists the different
        include files that contain the defined constants for the particular
        status codes:

        Numeric Range (in Hex)   Status Code Types
        -------------------------------------------------
        3FFA0000 to 3FFA1FFF     IVI      Warnings
        3FFF0000 to 3FFFFFFF     VISA     Warnings
        3FFC0000 to 3FFCFFFF     VXIPnP   Driver Warnings

        BFFA0000 to BFFA1FFF     IVI      Errors
        BFFF0000 to BFFFFFFF     VISA     Errors
        BFFC0000 to BFFCFFFF     VXIPnP   Driver Errors
```

5.5.54. bu6100_IviLxiSync_AddEvent

```
ViStatus bu6100_IviLxiSync_AddEvent (ViSession instrumentHandle,
                                     ViConstString eventName);
```

```
Purpose

    This function creates a new event.

Parameter List

    instrumentHandle

        Variable Type       ViSession

        Instrument handle.

    eventName

        Variable Type       ViConstString

        Specifies the name of the event to create.

Return Value

        Returns the status code of this operation.  The status code  either
        indicates success or describes an error or warning condition.  You
        examine the status code from each call to an instrument driver
        function to determine if an error occurred.

        To obtain a text description of the status code, call the
        bu6100_error_message function.  To obtain additional information
        about the error condition, call the bu6100_GetError function.  To
        clear the error information from the driver, call the
        bu6100_ClearError function.

        The general meaning of the status code is as follows:

        Value                  Meaning
        -------------------------------
        0                      Success
        Positive Values        Warnings
        Negative Values        Errors

        This instrument driver returns errors and warnings defined by other
        sources.  The following table defines the ranges of additional status
        codes that this driver can return.  The table lists the different
        include files that contain the defined constants for the particular
        status codes:

        Numeric Range (in Hex)   Status Code Types
        -------------------------------------------------
        3FFA0000 to 3FFA1FFF     IVI      Warnings
        3FFF0000 to 3FFFFFFF     VISA     Warnings
        3FFC0000 to 3FFCFFFF     VXIPnP   Driver Warnings

        BFFA0000 to BFFA1FFF     IVI      Errors
        BFFF0000 to BFFFFFFF     VISA     Errors
        BFFC0000 to BFFCFFFF     VXIPnP   Driver Errors
```

5.5.55. bu6100_IviLxiSync_AddTriggerAlarm

```
    ViStatus bu6100_IviLxiSync_AddTriggerAlarm (ViSession instrumentHandle,
                                                ViConstString alarmName);

Purpose

    This function creates a new trigger alarm

    ProDAQ 6100 does not support custom Trigger Alarms.
    Only "ALARM0" is supported by ProDAQ 6100.

Parameter List

    instrumentHandle

        Variable Type       ViSession

        Instrument handle.

    alarmName

        Variable Type       ViConstString

        Specifies the name of the trigger alarm to create.

Return Value

        Returns the status code of this operation.  The status code  either
        indicates success or describes an error or warning condition.  You
        examine the status code from each call to an instrument driver
        function to determine if an error occurred.

        To obtain a text description of the status code, call the
        bu6100_error_message function.  To obtain additional information
        about the error condition, call the bu6100_GetError function.  To
        clear the error information from the driver, call the
        bu6100_ClearError function.

        The general meaning of the status code is as follows:

        Value                  Meaning
        -------------------------------
        0                      Success
        Positive Values        Warnings
        Negative Values        Errors

        This instrument driver returns errors and warnings defined by other
        sources.  The following table defines the ranges of additional status
        codes that this driver can return.  The table lists the different
        include files that contain the defined constants for the particular
        status codes:

        Numeric Range (in Hex)   Status Code Types
        -------------------------------------------------
        3FFA0000 to 3FFA1FFF     IVI      Warnings
        3FFF0000 to 3FFFFFFF     VISA     Warnings
        3FFC0000 to 3FFCFFFF     VXIPnP   Driver Warnings

        BFFA0000 to BFFA1FFF     IVI      Errors
        BFFF0000 to BFFFFFFF     VISA     Errors
        BFFC0000 to BFFCFFFF     VXIPnP   Driver Errors
```

5.5.56. bu6100_IviLxiSync_AddTriggerSource

```
ViStatus bu6100_IviLxiSync_AddTriggerSource (ViSession instrumentHandle,
                                             ViConstString sourceName);
```

```
Purpose

    This function creates a new trigger source.
    ProDAQ 6100 does not support custom trigger sources.

Parameter List

    instrumentHandle

        Variable Type       ViSession

        Instrument handle.

    sourceName

        Variable Type       ViConstString

        Specifies the name of the trigger source to create.

Return Value

        Returns the status code of this operation.  The status code  either
        indicates success or describes an error or warning condition.  You
        examine the status code from each call to an instrument driver
        function to determine if an error occurred.

        To obtain a text description of the status code, call the
        bu6100_error_message function.  To obtain additional information
        about the error condition, call the bu6100_GetError function.  To
        clear the error information from the driver, call the
        bu6100_ClearError function.

        The general meaning of the status code is as follows:

        Value                  Meaning
        -------------------------------
        0                      Success
        Positive Values        Warnings
        Negative Values        Errors

        This instrument driver returns errors and warnings defined by other
        sources.  The following table defines the ranges of additional status
        codes that this driver can return.  The table lists the different
        include files that contain the defined constants for the particular
        status codes:

        Numeric Range (in Hex)   Status Code Types
        -------------------------------------------------
        3FFA0000 to 3FFA1FFF     IVI      Warnings
        3FFF0000 to 3FFFFFFF     VISA     Warnings
        3FFC0000 to 3FFCFFFF     VXIPnP   Driver Warnings

        BFFA0000 to BFFA1FFF     IVI      Errors
        BFFF0000 to BFFFFFFF     VISA     Errors
        BFFC0000 to BFFCFFFF     VXIPnP   Driver Errors
```

5.5.57. bu6100_IviLxiSync_ArmTrigger

```
    ViStatus bu6100_IviLxiSync_ArmTrigger (ViSession instrumentHandle);

Purpose

    This function arms the trigger.


Parameter List

    instrumentHandle

        Variable Type       ViSession

        Instrument handle.

Return Value

        Returns the status code of this operation.  The status code  either
        indicates success or describes an error or warning condition.  You
        examine the status code from each call to an instrument driver
        function to determine if an error occurred.

        To obtain a text description of the status code, call the
        bu6100_error_message function.  To obtain additional information
        about the error condition, call the bu6100_GetError function.  To
        clear the error information from the driver, call the
        bu6100_ClearError function.

        The general meaning of the status code is as follows:

        Value                  Meaning
        -------------------------------
        0                      Success
        Positive Values        Warnings
        Negative Values        Errors

        This instrument driver returns errors and warnings defined by other
        sources.  The following table defines the ranges of additional status
        codes that this driver can return.  The table lists the different
        include files that contain the defined constants for the particular
        status codes:

        Numeric Range (in Hex)   Status Code Types
        -------------------------------------------------
        3FFA0000 to 3FFA1FFF     IVI      Warnings
        3FFF0000 to 3FFFFFFF     VISA     Warnings
        3FFC0000 to 3FFCFFFF     VXIPnP   Driver Warnings

        BFFA0000 to BFFA1FFF     IVI      Errors
        BFFF0000 to BFFFFFFF     VISA     Errors
        BFFC0000 to BFFCFFFF     VXIPnP   Driver Errors
```

5.5.58. bu6100_IviLxiSync_ClearEventLog

```
    ViStatus bu6100_IviLxiSync_ClearEventLog (ViSession instrumentHandle);

Purpose

    This function removes all existing entries from the event log.

Parameter List

    instrumentHandle

        Variable Type       ViSession

        Instrument handle.

Return Value

        Returns the status code of this operation.  The status code  either
        indicates success or describes an error or warning condition.  You
        examine the status code from each call to an instrument driver
        function to determine if an error occurred.

        To obtain a text description of the status code, call the
        bu6100_error_message function.  To obtain additional information
        about the error condition, call the bu6100_GetError function.  To
        clear the error information from the driver, call the
        bu6100_ClearError function.

        The general meaning of the status code is as follows:

        Value                  Meaning
        -------------------------------
        0                      Success
        Positive Values        Warnings
        Negative Values        Errors

        This instrument driver returns errors and warnings defined by other
        sources.  The following table defines the ranges of additional status
        codes that this driver can return.  The table lists the different
        include files that contain the defined constants for the particular
        status codes:

        Numeric Range (in Hex)   Status Code Types
        -------------------------------------------------
        3FFA0000 to 3FFA1FFF     IVI      Warnings
        3FFF0000 to 3FFFFFFF     VISA     Warnings
        3FFC0000 to 3FFCFFFF     VXIPnP   Driver Warnings

        BFFA0000 to BFFA1FFF     IVI      Errors
        BFFF0000 to BFFFFFFF     VISA     Errors
        BFFC0000 to BFFCFFFF     VXIPnP   Driver Errors
```

5.5.59. bu6100_IviLxiSync_ClearTriggerLog

```
    ViStatus bu6100_IviLxiSync_ClearTriggerLog (ViSession instrumentHandle);

Purpose

    This function removes all existing entries from the trigger log.

Parameter List

    instrumentHandle

        Variable Type       ViSession

        Instrument handle.

Return Value

        Returns the status code of this operation.  The status code  either
        indicates success or describes an error or warning condition.  You
        examine the status code from each call to an instrument driver
        function to determine if an error occurred.

        To obtain a text description of the status code, call the
        bu6100_error_message function.  To obtain additional information
        about the error condition, call the bu6100_GetError function.  To
        clear the error information from the driver, call the
        bu6100_ClearError function.

        The general meaning of the status code is as follows:

        Value                  Meaning
        -------------------------------
        0                      Success
        Positive Values        Warnings
        Negative Values        Errors

        This instrument driver returns errors and warnings defined by other
        sources.  The following table defines the ranges of additional status
        codes that this driver can return.  The table lists the different
        include files that contain the defined constants for the particular
        status codes:

        Numeric Range (in Hex)   Status Code Types
        -------------------------------------------------
        3FFA0000 to 3FFA1FFF     IVI      Warnings
        3FFF0000 to 3FFFFFFF     VISA     Warnings
        3FFC0000 to 3FFCFFFF     VXIPnP   Driver Warnings

        BFFA0000 to BFFA1FFF     IVI      Errors
        BFFF0000 to BFFFFFFF     VISA     Errors
        BFFC0000 to BFFCFFFF     VXIPnP   Driver Errors
```

5.5.60. bu6100_IviLxiSync_ConfigureArmAlarm

```
ViStatus bu6100_IviLxiSync_ConfigureArmAlarm (ViSession instrumentHandle,
                                              ViConstString alarmName,
                                              ViBoolean enabled,
                                              ViReal64 timeSeconds,
                                              ViReal64 timeFraction,
                                              ViReal64 period,
                                              ViInt32 repeatCount);
```

```
Purpose

    This function configures the most commonly configured attributes of the
    arm alarm.

    ProDAQ 6100 does not support Arm subsystem.

Parameter List

    instrumentHandle

        Variable Type       ViSession

        Instrument handle.

    alarmName

        Variable Type       ViConstString

        The name of the alarm.

    enabled

        Variable Type       ViBoolean

        Enables or disables the arm alarm.
        Valid Values:
              VI_FALSE  0  Alarm is disabled
              VI_TRUE   1  Alarm is enabled
        Default Value: VI_FALSE

    timeSeconds

        Variable Type       ViReal64

        Specifies the seconds part of 1588 time.

    timeFraction

        Variable Type       ViReal64

        Specifies the fractional part of 1588 time.

    period

        Variable Type       ViReal64

        Specifies the period of the arm alarm.

    repeatCount

        Variable Type       ViInt32

        Specifies the number of times to repeat the trigger at the period
        specified by the Arm Alarm Repeat Period attribute.

Return Value

        Returns the status code of this operation.  The status code  either
        indicates success or describes an error or warning condition.  You
        examine the status code from each call to an instrument driver
        function to determine if an error occurred.

        To obtain a text description of the status code, call the
        bu6100_error_message function.  To obtain additional information
```

```
about the error condition, call the bu6100_GetError function.  To
clear the error information from the driver, call the
bu6100_ClearError function.
```

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

This instrument driver returns errors and warnings defined by other sources. The following table defines the ranges of additional status codes that this driver can return. The table lists the different include files that contain the defined constants for the particular status codes:

Numeric Range (in Hex)	Status Code Types	
3FFA0000 to 3FFA1FFF	IVI	Warnings
3FFF0000 to 3FFFFFFF	VISA	Warnings
3FFC0000 to 3FFCFFFF	VXIPnP	Driver Warnings
BFFA0000 to BFFA1FFF	IVI	Errors
BFFF0000 to BFFFFFFF	VISA	Errors
BFFC0000 to BFFCFFFF	VXIPnP	Driver Errors

5.5.61. bu6100_IviLxiSync_ConfigureArmSource

```
    ViStatus bu6100_IviLxiSync_ConfigureArmSource
                 (ViSession instrumentHandle, ViConstString sourceName,
                  ViBoolean enabled, ViInt32 detection);

Purpose

    This function configures the most commonly configured attributes of the
    arm source sub-system.

    ProDAQ 6100 does not support Arm subsystem.

Parameter List

    instrumentHandle

        Variable Type       ViSession

        Instrument handle.

    sourceName

        Variable Type       ViConstString

        Name of the arm source to configure.

    enabled

        Variable Type       ViBoolean

        Enables or disables the arm source.

        Valid Values:
                VI_FALSE   0  The Source is disabled;
                VI_TRUE    1  The Source is enabled;

        Default Value: VI_FALSE

    detection

        Variable Type       ViInt32

        Specifies the style of the arm source.

        Valid Values:

        BU6100_VAL_IVILXISYNC_DETECTION_RISE - Configures the LXI device to
        arm on the rising edge of the arm source.

        BU6100_VAL_IVILXISYNC_DETECTION_FALL - Configures the  LXI device  to
        arm on the falling edge of the arm source.

        BU6100_VAL_IVILXISYNC_DETECTION_HIGH - Configures the LXI device to
        arm while the arm source is high, that is, while it remains true.

        BU6100_VAL_IVILXISYNC_DETECTION_LOW - Configures the LXI device to
        arm while the arm source is low, that is, while it remains false.

        Default Value: BU6100_VAL_IVILXISYNC_DETECTION_RISE



Return Value

        Returns the status code of this operation.  The status code  either
        indicates success or describes an error or warning condition.  You
        examine the status code from each call to an instrument driver
        function to determine if an error occurred.

        To obtain a text description of the status code, call the
        bu6100_error_message function.  To obtain additional information
        about the error condition, call the bu6100_GetError function.  To
        clear the error information from the driver, call the
        bu6100_ClearError function.
```

```
The general meaning of the status code is as follows:

Value                  Meaning
-------------------------------
0                      Success
Positive Values        Warnings
Negative Values        Errors

This instrument driver returns errors and warnings defined by other
sources.  The following table defines the ranges of additional status
codes that this driver can return.  The table lists the different
include files that contain the defined constants for the particular
status codes:

Numeric Range (in Hex)   Status Code Types
-------------------------------------------------
3FFA0000 to 3FFA1FFF     IVI      Warnings
3FFF0000 to 3FFFFFFF     VISA     Warnings
3FFC0000 to 3FFCFFFF     VXIPnP   Driver Warnings

BFFA0000 to BFFA1FFF     IVI      Errors
BFFF0000 to BFFFFFFF     VISA     Errors
BFFC0000 to BFFCFFFF     VXIPnP   Driver Errors
```

5.5.62. bu6100_IviLxiSync_ConfigureEvent

```
ViStatus bu6100_IviLxiSync_ConfigureEvent (ViSession instrumentHandle,
                                           ViConstString eventName,
                                           ViInt32 driveMode,
                                           ViConstString source,
                                           ViConstString destinationPath,
                                           ViInt32 slope);
```

Purpose

This function configures the most commonly configured attributes of the event sub-system.

Parameter List

instrumentHandle

Variable Type	ViSession

Instrument handle.

eventName

Variable Type	ViConstString

Specifies the name of the event to configure.

driveMode

Variable Type	ViInt32

Specifies the mode of the event.

Valid Values:

BU6100_VAL_IVILXISYNC_EVENT_DRIVEN - Enables the event in driven mode.

BU6100_VAL_IVILXISYNC_EVENT_OFF - Disables the event.

BU6100_VAL_IVILXISYNC_EVENT_WIREDOR - Enables the event in wired-OR mode.

Default Value: BU6100_VAL_IVILXISYNC_EVENT_DRIVEN

source

Variable Type	ViConstString

Specifies the signal which causes an event to be transmitted.

destinationPath

Variable Type	ViConstString

Specifies a list of places to send the event.

slope

Variable Type	ViInt32

Specifies the slope of the event signal.

Valid Values:

BU6100_VAL_IVILXISYNC_SLOPE_RISE - The event will be transmitted with a rising edge.

BU6100_VAL_IVILXISYNC_SLOPE_FALL - The event will be transmitted with a falling edge.

Default Value: BU6100_VAL_IVILXISYNC_SLOPE_RISE

```
Return Value

        Returns the status code of this operation.  The status code  either
        indicates success or describes an error or warning condition.  You
        examine the status code from each call to an instrument driver
        function to determine if an error occurred.

        To obtain a text description of the status code, call the
        bu6100_error_message function.  To obtain additional information
        about the error condition, call the bu6100_GetError function.  To
        clear the error information from the driver, call the
        bu6100_ClearError function.

        The general meaning of the status code is as follows:

        Value                  Meaning
        -------------------------------
        0                      Success
        Positive Values        Warnings
        Negative Values        Errors

        This instrument driver returns errors and warnings defined by other
        sources.  The following table defines the ranges of additional status
        codes that this driver can return.  The table lists the different
        include files that contain the defined constants for the particular
        status codes:

        Numeric Range (in Hex)   Status Code Types
        -------------------------------------------------
        3FFA0000 to 3FFA1FFF     IVI      Warnings
        3FFF0000 to 3FFFFFFF     VISA     Warnings
        3FFC0000 to 3FFCFFFF     VXIPnP   Driver Warnings

        BFFA0000 to BFFA1FFF     IVI      Errors
        BFFF0000 to BFFFFFFF     VISA     Errors
        BFFC0000 to BFFCFFFF     VXIPnP   Driver Errors
```

5.5.63. bu6100_IviLxiSync_ConfigureTriggerAlarm

```
    ViStatus bu6100_IviLxiSync_ConfigureTriggerAlarm
                 (ViSession instrumentHandle, ViConstString alarmName,
                  ViReal64 timeSeconds, ViReal64 timeFraction,
                  ViReal64 period, ViInt32 repeatCount);

  Purpose

    This function configures the most commonly configured attributes of the
    trigger alarm sub-system.
    Only "ALARM0" is supported by ProDAQ 6100.


  Parameter List

    instrumentHandle

        Variable Type       ViSession

        Instrument handle.

    alarmName

        Variable Type       ViConstString

        The name of the alarm. Only "ALARM0" is supported by ProDAQ 6100.

    timeSeconds

        Variable Type       ViReal64

        Specifies the seconds part of 1588 time.

    timeFraction

        Variable Type       ViReal64

        Specifies the fractional part of 1588 time.

    period

        Variable Type       ViReal64

        Specifies the period of the trigger alarm.

    repeatCount

        Variable Type       ViInt32

        Specifies the number of times to repeat the trigger at the period
        specified by the Trigger Alarm Repeat Period attribute.

  Return Value

        Returns the status code of this operation.  The status code  either
        indicates success or describes an error or warning condition.  You
        examine the status code from each call to an instrument driver
        function to determine if an error occurred.

        To obtain a text description of the status code, call the
        bu6100_error_message function.  To obtain additional information
        about the error condition, call the bu6100_GetError function.  To
        clear the error information from the driver, call the
        bu6100_ClearError function.

        The general meaning of the status code is as follows:

        Value                  Meaning
        -------------------------------
        0                      Success
        Positive Values        Warnings
        Negative Values        Errors

        This instrument driver returns errors and warnings defined by other
        sources.  The following table defines the ranges of additional status
```

```
codes that this driver can return.  The table lists the different
include files that contain the defined constants for the particular
status codes:
```

```
Numeric Range (in Hex)   Status Code Types
-------------------------------------------------
3FFA0000 to 3FFA1FFF     IVI      Warnings
3FFF0000 to 3FFFFFFF     VISA     Warnings
3FFC0000 to 3FFCFFFF     VXIPnP   Driver Warnings

BFFA0000 to BFFA1FFF     IVI      Errors
BFFF0000 to BFFFFFFF     VISA     Errors
BFFC0000 to BFFCFFFF     VXIPnP   Driver Errors
```

5.5.64. bu6100_IviLxiSync_ConfigureTriggerLog

```
    ViStatus bu6100_IviLxiSync_ConfigureTriggerLog
                 (ViSession instrumentHandle, ViInt32 logMode);

  Purpose

    This function configures the trigger log.

  Parameter List

    instrumentHandle

        Variable Type       ViSession

        Instrument handle.

    logMode

        Variable Type       ViInt32

        Selects the mode for the trigger log.

        Valid Values:

        BU6100_LOG_DISABLED  0  No records will be added to the trigger
                                log;
        BU6100_LOG_ENABLED   1  Every trigger assert will be added to
                                the trigger log. When the log queue will
                                be full, the new record will not be
                                added, but the last record will be
                                marked with overflow flag.
        BU6100_LOG_DROP      2  Every trigger assert will be added to
                                the trigger log. When the log queue will
                                be full, the oldest record will be
                                overwritten with the new one.

  Return Value

        Returns the status code of this operation.  The status code  either
        indicates success or describes an error or warning condition.  You
        examine the status code from each call to an instrument driver
        function to determine if an error occurred.

        To obtain a text description of the status code, call the
        bu6100_error_message function.  To obtain additional information
        about the error condition, call the bu6100_GetError function.  To
        clear the error information from the driver, call the
        bu6100_ClearError function.

        The general meaning of the status code is as follows:

        Value                  Meaning
        -------------------------------
        0                      Success
        Positive Values        Warnings
        Negative Values        Errors

        This instrument driver returns errors and warnings defined by other
        sources.  The following table defines the ranges of additional status
        codes that this driver can return.  The table lists the different
        include files that contain the defined constants for the particular
        status codes:

        Numeric Range (in Hex)   Status Code Types
        -------------------------------------------------
        3FFA0000 to 3FFA1FFF     IVI      Warnings
        3FFF0000 to 3FFFFFFF     VISA     Warnings
        3FFC0000 to 3FFCFFFF     VXIPnP   Driver Warnings

        BFFA0000 to BFFA1FFF     IVI      Errors
        BFFF0000 to BFFFFFFF     VISA     Errors
        BFFC0000 to BFFCFFFF     VXIPnP   Driver Errors
```

5.5.65. bu6100_IviLxiSync_ConfigureTriggerSource

```
     ViStatus bu6100_IviLxiSync_ConfigureTriggerSource
                  (ViSession instrumentHandle, ViConstString sourceName,
                   ViReal64 delay, ViInt32 detection);

 Purpose

     This function configures the most commonly configured attributes of the
     trigger source sub-system.

 Parameter List

     instrumentHandle

         Variable Type       ViSession

         Instrument handle.

     sourceName

         Variable Type       ViConstString

         Name of the trigger source to configure.

     delay

         Variable Type       ViReal64

         Specifies the trigger source delay.  The units are seconds.  A
         negative value implies pre-trigger acquisition.

     detection

         Variable Type       ViInt32

         Specifies the slope of the trigger source.

         Valid Values:

         BU6100_VAL_IVILXISYNC_DETECTION_RISE - Configures the LXI device to
         trigger on the rising edge of the trigger source.

         BU6100_VAL_IVILXISYNC_DETECTION_FALL - Configures the LXI device to
         trigger on the falling edge of the trigger source.

         Default Value: BU6100_VAL_IVILXISYNC_DETECTION_RISE


 Return Value

         Returns the status code of this operation.  The status code  either
         indicates success or describes an error or warning condition.  You
         examine the status code from each call to an instrument driver
         function to determine if an error occurred.

         To obtain a text description of the status code, call the
         bu6100_error_message function.  To obtain additional information
         about the error condition, call the bu6100_GetError function.  To
         clear the error information from the driver, call the
         bu6100_ClearError function.

         The general meaning of the status code is as follows:

         Value                  Meaning
         -------------------------------
         0                      Success
         Positive Values        Warnings
         Negative Values        Errors

         This instrument driver returns errors and warnings defined by other
         sources.  The following table defines the ranges of additional status
         codes that this driver can return.  The table lists the different
         include files that contain the defined constants for the particular
         status codes:
```

```
Numeric Range (in Hex)   Status Code Types
-------------------------------------------------
3FFA0000 to 3FFA1FFF     IVI      Warnings
3FFF0000 to 3FFFFFFF     VISA     Warnings
3FFC0000 to 3FFCFFFF     VXIPnP   Driver Warnings

BFFA0000 to BFFA1FFF     IVI      Errors
BFFF0000 to BFFFFFFF     VISA     Errors
BFFC0000 to BFFCFFFF     VXIPnP   Driver Errors
```

5.5.66. bu6100_IviLxiSync_DisableAllArmAlarms

```
    ViStatus bu6100_IviLxiSync_DisableAllArmAlarms
                 (ViSession instrumentHandle);

Purpose

    This function disables all arm alarms.

    ProDAQ 6100 does not support Arm subsystem.

Parameter List

    instrumentHandle

        Variable Type       ViSession

        Instrument handle.

Return Value

        Returns the status code of this operation.  The status code  either
        indicates success or describes an error or warning condition.  You
        examine the status code from each call to an instrument driver
        function to determine if an error occurred.

        To obtain a text description of the status code, call the
        bu6100_error_message function.  To obtain additional information
        about the error condition, call the bu6100_GetError function.  To
        clear the error information from the driver, call the
        bu6100_ClearError function.

        The general meaning of the status code is as follows:

        Value                  Meaning
        -------------------------------
        0                      Success
        Positive Values        Warnings
        Negative Values        Errors

        This instrument driver returns errors and warnings defined by other
        sources.  The following table defines the ranges of additional status
        codes that this driver can return.  The table lists the different
        include files that contain the defined constants for the particular
        status codes:

        Numeric Range (in Hex)   Status Code Types
        -------------------------------------------------
        3FFA0000 to 3FFA1FFF     IVI      Warnings
        3FFF0000 to 3FFFFFFF     VISA     Warnings
        3FFC0000 to 3FFCFFFF     VXIPnP   Driver Warnings

        BFFA0000 to BFFA1FFF     IVI      Errors
        BFFF0000 to BFFFFFFF     VISA     Errors
        BFFC0000 to BFFCFFFF     VXIPnP   Driver Errors
```

5.5.67. bu6100_IviLxiSync_DisableAllArmSources

```
     ViStatus bu6100_IviLxiSync_DisableAllArmSources
                 (ViSession instrumentHandle);

 Purpose

     This function disables all arm sources.

     ProDAQ 6100 does not support Arm subsystem.

 Parameter List

     instrumentHandle

         Variable Type       ViSession

         Instrument handle.

 Return Value

         Returns the status code of this operation.  The status code  either
         indicates success or describes an error or warning condition.  You
         examine the status code from each call to an instrument driver
         function to determine if an error occurred.

         To obtain a text description of the status code, call the
         bu6100_error_message function.  To obtain additional information
         about the error condition, call the bu6100_GetError function.  To
         clear the error information from the driver, call the
         bu6100_ClearError function.

         The general meaning of the status code is as follows:

         Value                  Meaning
         -------------------------------
         0                      Success
         Positive Values        Warnings
         Negative Values        Errors

         This instrument driver returns errors and warnings defined by other
         sources.  The following table defines the ranges of additional status
         codes that this driver can return.  The table lists the different
         include files that contain the defined constants for the particular
         status codes:

         Numeric Range (in Hex)   Status Code Types
         -------------------------------------------------
         3FFA0000 to 3FFA1FFF     IVI      Warnings
         3FFF0000 to 3FFFFFFF     VISA     Warnings
         3FFC0000 to 3FFCFFFF     VXIPnP   Driver Warnings

         BFFA0000 to BFFA1FFF     IVI      Errors
         BFFF0000 to BFFFFFFF     VISA     Errors
         BFFC0000 to BFFCFFFF     VXIPnP   Driver Errors
```

5.5.68. bu6100_IviLxiSync_DisableAllEvents

```
    ViStatus bu6100_IviLxiSync_DisableAllEvents (ViSession instrumentHandle);

Purpose

    This function disables all events.

Parameter List

    instrumentHandle

        Variable Type       ViSession

        Instrument handle.

Return Value

        Returns the status code of this operation.  The status code  either
        indicates success or describes an error or warning condition.  You
        examine the status code from each call to an instrument driver
        function to determine if an error occurred.

        To obtain a text description of the status code, call the
        bu6100_error_message function.  To obtain additional information
        about the error condition, call the bu6100_GetError function.  To
        clear the error information from the driver, call the
        bu6100_ClearError function.

        The general meaning of the status code is as follows:

        Value                  Meaning
        -------------------------------
        0                      Success
        Positive Values        Warnings
        Negative Values        Errors

        This instrument driver returns errors and warnings defined by other
        sources.  The following table defines the ranges of additional status
        codes that this driver can return.  The table lists the different
        include files that contain the defined constants for the particular
        status codes:

        Numeric Range (in Hex)   Status Code Types
        -------------------------------------------------
        3FFA0000 to 3FFA1FFF     IVI      Warnings
        3FFF0000 to 3FFFFFFF     VISA     Warnings
        3FFC0000 to 3FFCFFFF     VXIPnP   Driver Warnings

        BFFA0000 to BFFA1FFF     IVI      Errors
        BFFF0000 to BFFFFFFF     VISA     Errors
        BFFC0000 to BFFCFFFF     VXIPnP   Driver Errors
```

5.5.69. bu6100_IviLxiSync_DisableAllTriggerAlarms

```
    ViStatus bu6100_IviLxiSync_DisableAllTriggerAlarms
                 (ViSession instrumentHandle);

Purpose

    This function disables all trigger alarms.
    Only "ALARM0" is supported by ProDAQ 6100.

Parameter List

    instrumentHandle

        Variable Type       ViSession

        Instrument handle.

Return Value

        Returns the status code of this operation.  The status code  either
        indicates success or describes an error or warning condition.  You
        examine the status code from each call to an instrument driver
        function to determine if an error occurred.

        To obtain a text description of the status code, call the
        bu6100_error_message function.  To obtain additional information
        about the error condition, call the bu6100_GetError function.  To
        clear the error information from the driver, call the
        bu6100_ClearError function.

        The general meaning of the status code is as follows:

        Value                  Meaning
        -------------------------------
        0                      Success
        Positive Values        Warnings
        Negative Values        Errors

        This instrument driver returns errors and warnings defined by other
        sources.  The following table defines the ranges of additional status
        codes that this driver can return.  The table lists the different
        include files that contain the defined constants for the particular
        status codes:

        Numeric Range (in Hex)   Status Code Types
        -------------------------------------------------
        3FFA0000 to 3FFA1FFF     IVI      Warnings
        3FFF0000 to 3FFFFFFF     VISA     Warnings
        3FFC0000 to 3FFCFFFF     VXIPnP   Driver Warnings

        BFFA0000 to BFFA1FFF     IVI      Errors
        BFFF0000 to BFFFFFFF     VISA     Errors
        BFFC0000 to BFFCFFFF     VXIPnP   Driver Errors
```

5.5.70. bu6100_IviLxiSync_GetArmAlarmName

```
ViStatus bu6100_IviLxiSync_GetArmAlarmName (ViSession instrumentHandle,
                                            ViInt32 alarmIndex,
                                            ViInt32 alarmNameBufferSize,
                                            ViChar _VI_FAR alarmName[]);
```

Purpose

This function returns the physical repeated capability identifier that corresponds to the one-based index that the user specifies. If the value that the user passes for the AlarmIndex parameter is less than one or greater than the value of the Arm Alarm Count attribute, the function returns an empty string in the AlarmName parameter and returns an error. For custom arm sources added vith the Add Arm Source function, this function returns the arm source name in the original casing used when Add Arm Source was called.

ProDAQ 6100 does not support Arm subsystem.

Parameter List

instrumentHandle

Variable Type	ViSession

Instrument handle.

alarmIndex

Variable Type	ViInt32

A one-based index that defines which name to return.

alarmNameBufferSize

Variable Type	ViInt32

The number of bytes in the ViChar array that the user specifies for the AlarmName parameter.

alarmName

Variable Type	ViChar[]

The buffer into which the function returns the alarm name that corresponds to the index the user specifies. The caller may pass VI_NULL for this parameter if the AlarmNameBufferSize parameter is 0.

Return Value

Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You examine the status code from each call to an instrument driver function to determine if an error occurred.

To obtain a text description of the status code, call the bu6100_error_message function. To obtain additional information about the error condition, call the bu6100_GetError function. To clear the error information from the driver, call the bu6100_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

This instrument driver returns errors and warnings defined by other sources. The following table defines the ranges of additional status codes that this driver can return. The table lists the different include files that contain the defined constants for the particular status codes:

```
Numeric Range (in Hex)   Status Code Types
-------------------------------------------------
3FFA0000 to 3FFA1FFF     IVI      Warnings
3FFF0000 to 3FFFFFFF     VISA     Warnings
3FFC0000 to 3FFCFFFF     VXIPnP   Driver Warnings

BFFA0000 to BFFA1FFF     IVI      Errors
BFFF0000 to BFFFFFFF     VISA     Errors
BFFC0000 to BFFCFFFF     VXIPnP   Driver Errors
```

5.5.71. bu6100_IviLxiSync_GetArmSourceName

```
ViStatus bu6100_IviLxiSync_GetArmSourceName (ViSession instrumentHandle,
                                             ViInt32 sourceIndex,
                                             ViInt32 sourceNameBufferSize,
                                             ViChar _VI_FAR sourceName[]);
```

Purpose

This function returns the physical repeated capability identifier that corresponds to the one-based index that the user specifies. If the value that the user passes for the SourceIndex parameter is less than one or greater than the value of the Arm Source Count attribute, the function returns an empty string in the SourceName parameter and returns an error. For custom arm sources added vith the Add Arm Source function, this function returns the arm source name in the original casing used when Add Arm Source was called.

ProDAQ 6100 does not support Arm subsystem.

Parameter List

instrumentHandle

Variable Type	ViSession

Instrument handle.

sourceIndex

Variable Type	ViInt32

A one-based index that defines which name to return.

sourceNameBufferSize

Variable Type	ViInt32

The number of bytes in the ViChar array that the user specifies for the SourceName parameter.

sourceName

Variable Type	ViChar[]

The buffer into which the function returns the source name that corresponds to the index the user specifies. The caller may pass VI_NULL for this parameter if the SourceNameBufferSize parameter is 0.

Return Value

Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You examine the status code from each call to an instrument driver function to determine if an error occurred.

To obtain a text description of the status code, call the bu6100_error_message function. To obtain additional information about the error condition, call the bu6100_GetError function. To clear the error information from the driver, call the bu6100_ClearError function.

The general meaning of the status code is as follows:

```
Value                  Meaning
-------------------------------
0                      Success
Positive Values        Warnings
Negative Values        Errors
```

This instrument driver returns errors and warnings defined by other sources. The following table defines the ranges of additional status codes that this driver can return. The table lists the different include files that contain the defined constants for the particular status codes:

```
Numeric Range (in Hex)   Status Code Types
-------------------------------------------------
3FFA0000 to 3FFA1FFF     IVI      Warnings
3FFF0000 to 3FFFFFFF     VISA     Warnings
3FFC0000 to 3FFCFFFF     VXIPnP   Driver Warnings

BFFA0000 to BFFA1FFF     IVI      Errors
BFFF0000 to BFFFFFFF     VISA     Errors
BFFC0000 to BFFCFFFF     VXIPnP   Driver Errors
```

5.5.72. bu6100_IviLxiSync_GetEventName

```
ViStatus bu6100_IviLxiSync_GetEventName (ViSession instrumentHandle,
                                         ViInt32 eventIndex,
                                         ViInt32 eventNameBufferSize,
                                         ViChar _VI_FAR eventName[]);
```

```
Purpose

    This function returns the physical repeated capability identifier that corresponds to the
    one-based index that the user specifies. If the value that the user passes for the
    EventIndex parameter is less than one or greater than the value of the Event Count attribute,
    the function returns an empty string in the EventName parameter and returns an error. For
    custom event sources added vith the Add Event Source function, this function returns the
    event source name in the original casing used when Add Event Source was called.

Parameter List

    instrumentHandle

        Variable Type       ViSession

        Instrument handle.

    eventIndex

        Variable Type       ViInt32

        A one-based index that defines which name to return.

    eventNameBufferSize

        Variable Type       ViInt32

        The number of bytes in the ViChar array that the user specifies for
        the EventName parameter.

    eventName

        Variable Type       ViChar[]

        The buffer into which the function returns the alarm name that
        corresponds to the index the user specifies. The caller may pass
        VI_NULL for this parameter if the EventNameBufferSize parameter is 0.

Return Value

        Returns the status code of this operation.  The status code  either
        indicates success or describes an error or warning condition.  You
        examine the status code from each call to an instrument driver
        function to determine if an error occurred.

        To obtain a text description of the status code, call the
        bu6100_error_message function.  To obtain additional information
        about the error condition, call the bu6100_GetError function.  To
        clear the error information from the driver, call the
        bu6100_ClearError function.

        The general meaning of the status code is as follows:

        Value                  Meaning
        -------------------------------
        0                      Success
        Positive Values        Warnings
        Negative Values        Errors

        This instrument driver returns errors and warnings defined by other
        sources.  The following table defines the ranges of additional status
        codes that this driver can return.  The table lists the different
        include files that contain the defined constants for the particular
        status codes:

        Numeric Range (in Hex)   Status Code Types
        -------------------------------------------------
        3FFA0000 to 3FFA1FFF     IVI      Warnings
        3FFF0000 to 3FFFFFFF     VISA     Warnings
        3FFC0000 to 3FFCFFFF     VXIPnP   Driver Warnings

        BFFA0000 to BFFA1FFF     IVI      Errors
        BFFF0000 to BFFFFFFF     VISA     Errors
        BFFC0000 to BFFCFFFF     VXIPnP   Driver Errors
```

5.5.73. bu6100_IviLxiSync_GetNextEventLogEntry

```
    ViStatus bu6100_IviLxiSync_GetNextEventLogEntry
                (ViSession instrumentHandle, ViInt32 logEntryBufferSize,
                 ViChar _VI_FAR logEntry[]);

Purpose

    This function retrieves and clears the oldest event log entry for the IVI
    session.

Parameter List

    instrumentHandle

        Variable Type       ViSession

        Instrument handle.

    logEntryBufferSize

        Variable Type       ViInt32

        The number of bytes in the ViChar array that the user specifies for
        the LogEntry parameter.

    logEntry

        Variable Type       ViChar[]

        The buffer in which the function returns the oldest event log entry.
        Can be VI_NULL if LogEntryBufferSize is 0.

Return Value

        Returns the status code of this operation.  The status code  either
        indicates success or describes an error or warning condition.  You
        examine the status code from each call to an instrument driver
        function to determine if an error occurred.

        To obtain a text description of the status code, call the
        bu6100_error_message function.  To obtain additional information
        about the error condition, call the bu6100_GetError function.  To
        clear the error information from the driver, call the
        bu6100_ClearError function.

        The general meaning of the status code is as follows:

        Value                  Meaning
        -------------------------------
        0                      Success
        Positive Values        Warnings
        Negative Values        Errors

        This instrument driver returns errors and warnings defined by other
        sources.  The following table defines the ranges of additional status
        codes that this driver can return.  The table lists the different
        include files that contain the defined constants for the particular
        status codes:

        Numeric Range (in Hex)   Status Code Types
        -------------------------------------------------
        3FFA0000 to 3FFA1FFF     IVI      Warnings
        3FFF0000 to 3FFFFFFF     VISA     Warnings
        3FFC0000 to 3FFCFFFF     VXIPnP   Driver Warnings

        BFFA0000 to BFFA1FFF     IVI      Errors
        BFFF0000 to BFFFFFFF     VISA     Errors
        BFFC0000 to BFFCFFFF     VXIPnP   Driver Errors
```

5.5.74. bu6100_IviLxiSync_GetNextTriggerLogEntry

```
    ViStatus bu6100_IviLxiSync_GetNextTriggerLogEntry
                 (ViSession instrumentHandle, ViInt32 logEntryBufferSize,
                  ViChar _VI_FAR logEntry[], ViPReal64 timeSeconds,
                  ViPReal64 timeFractional);
```

```
Purpose

    This function retrieves and clears the oldest trigger log entry for the
    IVI session.

Parameter List

    instrumentHandle

        Variable Type       ViSession

        Instrument handle.

    logEntryBufferSize

        Variable Type       ViInt32

        The number of bytes in the ViChar array that the user specifies for
        the LogEntry parameter.

    logEntry

        Variable Type       ViChar[]

        The buffer in which the function returns the oldest event log entry.
        Can be VI_NULL if LogEntryBufferSize is 0.

    timeSeconds

        Variable Type       ViReal64 (passed by reference)

    timeFractional

        Variable Type       ViReal64 (passed by reference)

Return Value

        Returns the status code of this operation.  The status code  either
        indicates success or describes an error or warning condition.  You
        examine the status code from each call to an instrument driver
        function to determine if an error occurred.

        To obtain a text description of the status code, call the
        bu6100_error_message function.  To obtain additional information
        about the error condition, call the bu6100_GetError function.  To
        clear the error information from the driver, call the
        bu6100_ClearError function.

        The general meaning of the status code is as follows:

        Value                  Meaning
        -------------------------------
        0                      Success
        Positive Values        Warnings
        Negative Values        Errors

        This instrument driver returns errors and warnings defined by other
        sources.  The following table defines the ranges of additional status
        codes that this driver can return.  The table lists the different
        include files that contain the defined constants for the particular
        status codes:

        Numeric Range (in Hex)   Status Code Types
        -------------------------------------------------
        3FFA0000 to 3FFA1FFF     IVI      Warnings
        3FFF0000 to 3FFFFFFF     VISA     Warnings
        3FFC0000 to 3FFCFFFF     VXIPnP   Driver Warnings
```

```
BFFA0000 to BFFA1FFF     IVI      Errors
BFFF0000 to BFFFFFFF     VISA     Errors
BFFC0000 to BFFCFFFF     VXIPnP   Driver Errors
```

5.5.75. bu6100_IviLxiSync_GetNumberOfTriggerLogEntries

```
     ViStatus bu6100_IviLxiSync_GetNumberOfTriggerLogEntries
                  (ViSession instrumentHandle, ViPInt32 number_ofLogEntries);

 Purpose

     Returns the number of records in the trigger log book.

 Parameter List

     instrumentHandle

         Variable Type       ViSession

         Instrument handle.

     number_ofLogEntries

         Variable Type       ViInt32 (passed by reference)

         Returns the number of records in the trigger log book.

 Return Value

         Returns the status code of this operation.  The status code  either
         indicates success or describes an error or warning condition.  You
         examine the status code from each call to an instrument driver
         function to determine if an error occurred.

         To obtain a text description of the status code, call the
         bu6100_error_message function.  To obtain additional information
         about the error condition, call the bu6100_GetError function.  To
         clear the error information from the driver, call the
         bu6100_ClearError function.

         The general meaning of the status code is as follows:

         Value                  Meaning
         -------------------------------
         0                      Success
         Positive Values        Warnings
         Negative Values        Errors

         This instrument driver returns errors and warnings defined by other
         sources.  The following table defines the ranges of additional status
         codes that this driver can return.  The table lists the different
         include files that contain the defined constants for the particular
         status codes:

         Numeric Range (in Hex)   Status Code Types
         -------------------------------------------------
         3FFA0000 to 3FFA1FFF     IVI      Warnings
         3FFF0000 to 3FFFFFFF     VISA     Warnings
         3FFC0000 to 3FFCFFFF     VXIPnP   Driver Warnings

         BFFA0000 to BFFA1FFF     IVI      Errors
         BFFF0000 to BFFFFFFF     VISA     Errors
         BFFC0000 to BFFCFFFF     VXIPnP   Driver Errors
```

5.5.76. bu6100_IviLxiSync_GetSystemTime

```
ViStatus bu6100_IviLxiSync_GetSystemTime (ViSession instrumentHandle,
                                          ViPReal64 timeSeconds,
                                          ViPReal64 timeFractional);
```

```
Purpose

    This function retrieves the current 1588 time.

Parameter List

    instrumentHandle

        Variable Type       ViSession

        Instrument handle.

    timeSeconds

        Variable Type       ViReal64 (passed by reference)

        Indicates the seconds portion of the current 1588 time.

    timeFractional

        Variable Type       ViReal64 (passed by reference)

        Indicates the fractional portion of the current 1588 time.

Return Value

        Returns the status code of this operation.  The status code  either
        indicates success or describes an error or warning condition.  You
        examine the status code from each call to an instrument driver
        function to determine if an error occurred.

        To obtain a text description of the status code, call the
        bu6100_error_message function.  To obtain additional information
        about the error condition, call the bu6100_GetError function.  To
        clear the error information from the driver, call the
        bu6100_ClearError function.

        The general meaning of the status code is as follows:

        Value                  Meaning
        -------------------------------
        0                      Success
        Positive Values        Warnings
        Negative Values        Errors

        This instrument driver returns errors and warnings defined by other
        sources.  The following table defines the ranges of additional status
        codes that this driver can return.  The table lists the different
        include files that contain the defined constants for the particular
        status codes:

        Numeric Range (in Hex)   Status Code Types
        -------------------------------------------------
        3FFA0000 to 3FFA1FFF     IVI      Warnings
        3FFF0000 to 3FFFFFFF     VISA     Warnings
        3FFC0000 to 3FFCFFFF     VXIPnP   Driver Warnings

        BFFA0000 to BFFA1FFF     IVI      Errors
        BFFF0000 to BFFFFFFF     VISA     Errors
        BFFC0000 to BFFCFFFF     VXIPnP   Driver Errors
```

5.5.77. bu6100_IviLxiSync_GetTriggerAlarmName

```
    ViStatus bu6100_IviLxiSync_GetTriggerAlarmName
                 (ViSession instrumentHandle, ViInt32 alarmIndex,
                  ViInt32 alarmNameBufferSize, ViChar _VI_FAR alarmName[]);

  Purpose

    This function returns the physical repeated capability identifier that
    corresponds to the one-based index that the user specifies. If the value
    that the user passes for the AlarmIndex parameter is less than one or
    greater than the value of the Trigger Alarm Count attribute, the function
    returns an empty string in the AlarmName parameter and returns an error.
    For custom trigger sources added with the Add Arm Source function, this
    function returns the arm source name in the original casing used when Add
    Arm Source was called.
    Only "ALARM0" is supported by ProDAQ 6100.

  Parameter List

    instrumentHandle

        Variable Type       ViSession

        Instrument handle.

    alarmIndex

        Variable Type       ViInt32

        A one-based index that defines which name to return.

    alarmNameBufferSize

        Variable Type       ViInt32

        The number of bytes in the ViChar array that the user specifies for
        the AlarmName parameter.

    alarmName

        Variable Type       ViChar[]

        The buffer into which the function returns the alarm name that
        corresponds to the index the user specifies. The caller may pass
        VI_NULL for this parameter if the AlarmNameBufferSize parameter is 0.

  Return Value

        Returns the status code of this operation.  The status code  either
        indicates success or describes an error or warning condition.  You
        examine the status code from each call to an instrument driver
        function to determine if an error occurred.

        To obtain a text description of the status code, call the
        bu6100_error_message function.  To obtain additional information
        about the error condition, call the bu6100_GetError function.  To
        clear the error information from the driver, call the
        bu6100_ClearError function.

        The general meaning of the status code is as follows:

        Value                  Meaning
        -------------------------------
        0                      Success
        Positive Values        Warnings
        Negative Values        Errors

        This instrument driver returns errors and warnings defined by other
        sources.  The following table defines the ranges of additional status
        codes that this driver can return.  The table lists the different
        include files that contain the defined constants for the particular
        status codes:
```

```
Numeric Range (in Hex)   Status Code Types
-------------------------------------------------
3FFA0000 to 3FFA1FFF     IVI      Warnings
3FFF0000 to 3FFFFFFF     VISA     Warnings
3FFC0000 to 3FFCFFFF     VXIPnP   Driver Warnings

BFFA0000 to BFFA1FFF     IVI      Errors
BFFF0000 to BFFFFFFF     VISA     Errors
BFFC0000 to BFFCFFFF     VXIPnP   Driver Errors
```

5.5.78. bu6100_IviLxiSync_GetTriggerSourceName

```
ViStatus bu6100_IviLxiSync_GetTriggerSourceName
             (ViSession instrumentHandle, ViInt32 sourceIndex,
              ViInt32 sourceNameBufferSize,
              ViChar _VI_FAR sourceName[]);
```

```
Purpose

    This function returns the physical repeated capability identifier that
    corresponds to the one-based index that the user specifies. If the value
    that the user passes for the SourceIndex parameter is less than one or
    greater than the value of the Trigger Source Count attribute, the
    function returns an empty string in the SourceName parameter and returns
    an error. For custom trigger sources added vith the Add Trigger Source
    function, this function returns the trigger source name in the original
    casing used when Add Trigger Source was called.

Parameter List

    instrumentHandle

        Variable Type       ViSession

        Instrument handle.

    sourceIndex

        Variable Type       ViInt32

        A one-based index that defines which name to return.

    sourceNameBufferSize

        Variable Type       ViInt32

        The number of bytes in the ViChar array that the user specifies for
        the SourceName parameter.

    sourceName

        Variable Type       ViChar[]

        The buffer into which the function returns the source name that
        corresponds to the index the user specifies. The caller may pass
        VI_NULL for this parameter if the SourceNameBufferSize parameter is
        0.

Return Value

        Returns the status code of this operation.  The status code  either
        indicates success or describes an error or warning condition.  You
        examine the status code from each call to an instrument driver
        function to determine if an error occurred.

        To obtain a text description of the status code, call the
        bu6100_error_message function.  To obtain additional information
        about the error condition, call the bu6100_GetError function.  To
        clear the error information from the driver, call the
        bu6100_ClearError function.

        The general meaning of the status code is as follows:

        Value                  Meaning
        -------------------------------
        0                      Success
        Positive Values        Warnings
        Negative Values        Errors

        This instrument driver returns errors and warnings defined by other
        sources.  The following table defines the ranges of additional status
        codes that this driver can return.  The table lists the different
        include files that contain the defined constants for the particular
        status codes:
```

```
Numeric Range (in Hex)   Status Code Types
-------------------------------------------------
3FFA0000 to 3FFA1FFF     IVI      Warnings
3FFF0000 to 3FFFFFFF     VISA     Warnings
3FFC0000 to 3FFCFFFF     VXIPnP   Driver Warnings

BFFA0000 to BFFA1FFF     IVI      Errors
BFFF0000 to BFFFFFFF     VISA     Errors
BFFC0000 to BFFCFFFF     VXIPnP   Driver Errors
```

5.5.79. bu6100_IviLxiSync_RemoveAllCustomArmAlarms

```
    ViStatus bu6100_IviLxiSync_RemoveAllCustomArmAlarms
                 (ViSession instrumentHandle);

Purpose

    This function removes all of the custom arm alarms that were added using
    the Add Arm Alarm function.

    ProDAQ 6100 does not support Arm subsystem.

Parameter List

    instrumentHandle

        Variable Type       ViSession

        Instrument handle.

Return Value

        Returns the status code of this operation.  The status code  either
        indicates success or describes an error or warning condition.  You
        examine the status code from each call to an instrument driver
        function to determine if an error occurred.

        To obtain a text description of the status code, call the
        bu6100_error_message function.  To obtain additional information
        about the error condition, call the bu6100_GetError function.  To
        clear the error information from the driver, call the
        bu6100_ClearError function.

        The general meaning of the status code is as follows:

        Value                  Meaning
        -------------------------------
        0                      Success
        Positive Values        Warnings
        Negative Values        Errors

        This instrument driver returns errors and warnings defined by other
        sources.  The following table defines the ranges of additional status
        codes that this driver can return.  The table lists the different
        include files that contain the defined constants for the particular
        status codes:

        Numeric Range (in Hex)   Status Code Types
        -------------------------------------------------
        3FFA0000 to 3FFA1FFF     IVI      Warnings
        3FFF0000 to 3FFFFFFF     VISA     Warnings
        3FFC0000 to 3FFCFFFF     VXIPnP   Driver Warnings

        BFFA0000 to BFFA1FFF     IVI      Errors
        BFFF0000 to BFFFFFFF     VISA     Errors
        BFFC0000 to BFFCFFFF     VXIPnP   Driver Errors
```

5.5.80. bu6100_IviLxiSync_RemoveAllCustomArmSources

```
     ViStatus bu6100_IviLxiSync_RemoveAllCustomArmSources
                  (ViSession instrumentHandle);

 Purpose

     This function removes all of the custom arm sources that were added using
     the Add Arm Source function.

     ProDAQ 6100 does not support Arm subsystem.

 Parameter List

     instrumentHandle

         Variable Type       ViSession

         Instrument handle.

 Return Value

         Returns the status code of this operation.  The status code  either
         indicates success or describes an error or warning condition.  You
         examine the status code from each call to an instrument driver
         function to determine if an error occurred.

         To obtain a text description of the status code, call the
         bu6100_error_message function.  To obtain additional information
         about the error condition, call the bu6100_GetError function.  To
         clear the error information from the driver, call the
         bu6100_ClearError function.

         The general meaning of the status code is as follows:

         Value                  Meaning
         -------------------------------
         0                      Success
         Positive Values        Warnings
         Negative Values        Errors

         This instrument driver returns errors and warnings defined by other
         sources.  The following table defines the ranges of additional status
         codes that this driver can return.  The table lists the different
         include files that contain the defined constants for the particular
         status codes:

         Numeric Range (in Hex)   Status Code Types
         -------------------------------------------------
         3FFA0000 to 3FFA1FFF     IVI      Warnings
         3FFF0000 to 3FFFFFFF     VISA     Warnings
         3FFC0000 to 3FFCFFFF     VXIPnP   Driver Warnings

         BFFA0000 to BFFA1FFF     IVI      Errors
         BFFF0000 to BFFFFFFF     VISA     Errors
         BFFC0000 to BFFCFFFF     VXIPnP   Driver Errors
```

5.5.81. bu6100_IviLxiSync_RemoveAllCustomEvents

```
     ViStatus bu6100_IviLxiSync_RemoveAllCustomEvents
                  (ViSession instrumentHandle);

 Purpose

     This function removes all of the custom events that were added using the
     Add Event function.

 Parameter List

     instrumentHandle

         Variable Type       ViSession

         Instrument handle.

 Return Value

         Returns the status code of this operation.  The status code  either
         indicates success or describes an error or warning condition.  You
         examine the status code from each call to an instrument driver
         function to determine if an error occurred.

         To obtain a text description of the status code, call the
         bu6100_error_message function.  To obtain additional information
         about the error condition, call the bu6100_GetError function.  To
         clear the error information from the driver, call the
         bu6100_ClearError function.

         The general meaning of the status code is as follows:

         Value                  Meaning
         -------------------------------
         0                      Success
         Positive Values        Warnings
         Negative Values        Errors

         This instrument driver returns errors and warnings defined by other
         sources.  The following table defines the ranges of additional status
         codes that this driver can return.  The table lists the different
         include files that contain the defined constants for the particular
         status codes:

         Numeric Range (in Hex)   Status Code Types
         -------------------------------------------------
         3FFA0000 to 3FFA1FFF     IVI      Warnings
         3FFF0000 to 3FFFFFFF     VISA     Warnings
         3FFC0000 to 3FFCFFFF     VXIPnP   Driver Warnings

         BFFA0000 to BFFA1FFF     IVI      Errors
         BFFF0000 to BFFFFFFF     VISA     Errors
         BFFC0000 to BFFCFFFF     VXIPnP   Driver Errors
```

5.5.82. bu6100_IviLxiSync_RemoveAllCustomTriggerSources

```
     ViStatus bu6100_IviLxiSync_RemoveAllCustomTriggerSources
                  (ViSession instrumentHandle);

 Purpose

     This function removes all of the custom trigger sources that were added
     using the Add Trigger Source function.

 Parameter List

     instrumentHandle

         Variable Type       ViSession

         Instrument handle.

 Return Value

         Returns the status code of this operation.  The status code  either
         indicates success or describes an error or warning condition.  You
         examine the status code from each call to an instrument driver
         function to determine if an error occurred.

         To obtain a text description of the status code, call the
         bu6100_error_message function.  To obtain additional information
         about the error condition, call the bu6100_GetError function.  To
         clear the error information from the driver, call the
         bu6100_ClearError function.

         The general meaning of the status code is as follows:

         Value                  Meaning
         -------------------------------
         0                      Success
         Positive Values        Warnings
         Negative Values        Errors

         This instrument driver returns errors and warnings defined by other
         sources.  The following table defines the ranges of additional status
         codes that this driver can return.  The table lists the different
         include files that contain the defined constants for the particular
         status codes:

         Numeric Range (in Hex)   Status Code Types
         -------------------------------------------------
         3FFA0000 to 3FFA1FFF     IVI      Warnings
         3FFF0000 to 3FFFFFFF     VISA     Warnings
         3FFC0000 to 3FFCFFFF     VXIPnP   Driver Warnings

         BFFA0000 to BFFA1FFF     IVI      Errors
         BFFF0000 to BFFFFFFF     VISA     Errors
         BFFC0000 to BFFCFFFF     VXIPnP   Driver Errors
```

5.5.83. bu6100_IviLxiSync_RemoveAllTriggerAlarms

```
    ViStatus bu6100_IviLxiSync_RemoveAllTriggerAlarms
                 (ViSession instrumentHandle);

Purpose

    This function removes all of the trigger alarms that were added using the
    Add Trigger Alarm function.

    Only "ALARM0" is supported by ProDAQ 6100.

Parameter List

    instrumentHandle

        Variable Type       ViSession

        Instrument handle.

Return Value

        Returns the status code of this operation.  The status code  either
        indicates success or describes an error or warning condition.  You
        examine the status code from each call to an instrument driver
        function to determine if an error occurred.

        To obtain a text description of the status code, call the
        bu6100_error_message function.  To obtain additional information
        about the error condition, call the bu6100_GetError function.  To
        clear the error information from the driver, call the
        bu6100_ClearError function.

        The general meaning of the status code is as follows:

        Value                  Meaning
        -------------------------------
        0                      Success
        Positive Values        Warnings
        Negative Values        Errors

        This instrument driver returns errors and warnings defined by other
        sources.  The following table defines the ranges of additional status
        codes that this driver can return.  The table lists the different
        include files that contain the defined constants for the particular
        status codes:

        Numeric Range (in Hex)   Status Code Types
        -------------------------------------------------
        3FFA0000 to 3FFA1FFF     IVI      Warnings
        3FFF0000 to 3FFFFFFF     VISA     Warnings
        3FFC0000 to 3FFCFFFF     VXIPnP   Driver Warnings

        BFFA0000 to BFFA1FFF     IVI      Errors
        BFFF0000 to BFFFFFFF     VISA     Errors
        BFFC0000 to BFFCFFFF     VXIPnP   Driver Errors
```

5.5.84. bu6100_IviLxiSync_RemoveArmAlarm

```
    ViStatus bu6100_IviLxiSync_RemoveArmAlarm (ViSession instrumentHandle,
                                           ViConstString alarmName);

Purpose

    This function removes an Arm Alarm.

    ProDAQ 6100 does not support Arm subsystem.

Parameter List

    instrumentHandle

        Variable Type       ViSession

        Instrument handle.

    alarmName

        Variable Type       ViConstString

        Specifies the name of the arm alarm to remove.

Return Value

        Returns the status code of this operation.  The status code  either
        indicates success or describes an error or warning condition.  You
        examine the status code from each call to an instrument driver
        function to determine if an error occurred.

        To obtain a text description of the status code, call the
        bu6100_error_message function.  To obtain additional information
        about the error condition, call the bu6100_GetError function.  To
        clear the error information from the driver, call the
        bu6100_ClearError function.

        The general meaning of the status code is as follows:

        Value                  Meaning
        -------------------------------
        0                      Success
        Positive Values        Warnings
        Negative Values        Errors

        This instrument driver returns errors and warnings defined by other
        sources.  The following table defines the ranges of additional status
        codes that this driver can return.  The table lists the different
        include files that contain the defined constants for the particular
        status codes:

        Numeric Range (in Hex)   Status Code Types
        -------------------------------------------------
        3FFA0000 to 3FFA1FFF     IVI      Warnings
        3FFF0000 to 3FFFFFFF     VISA     Warnings
        3FFC0000 to 3FFCFFFF     VXIPnP   Driver Warnings

        BFFA0000 to BFFA1FFF     IVI      Errors
        BFFF0000 to BFFFFFFF     VISA     Errors
        BFFC0000 to BFFCFFFF     VXIPnP   Driver Errors
```

5.5.85. bu6100_IviLxiSync_RemoveArmSource

```
    ViStatus bu6100_IviLxiSync_RemoveArmSource (ViSession instrumentHandle,
                                                ViConstString sourceName);

Purpose

    This function removes an arm source.

    ProDAQ 6100 does not support Arm subsystem.

Parameter List

    instrumentHandle

        Variable Type       ViSession

        Instrument handle.

    sourceName

        Variable Type       ViConstString

        Specifies the name of the arm source to remove.

Return Value

        Returns the status code of this operation.  The status code  either
        indicates success or describes an error or warning condition.  You
        examine the status code from each call to an instrument driver
        function to determine if an error occurred.

        To obtain a text description of the status code, call the
        bu6100_error_message function.  To obtain additional information
        about the error condition, call the bu6100_GetError function.  To
        clear the error information from the driver, call the
        bu6100_ClearError function.

        The general meaning of the status code is as follows:

        Value                  Meaning
        -------------------------------
        0                      Success
        Positive Values        Warnings
        Negative Values        Errors

        This instrument driver returns errors and warnings defined by other
        sources.  The following table defines the ranges of additional status
        codes that this driver can return.  The table lists the different
        include files that contain the defined constants for the particular
        status codes:

        Numeric Range (in Hex)   Status Code Types
        -------------------------------------------------
        3FFA0000 to 3FFA1FFF     IVI      Warnings
        3FFF0000 to 3FFFFFFF     VISA     Warnings
        3FFC0000 to 3FFCFFFF     VXIPnP   Driver Warnings

        BFFA0000 to BFFA1FFF     IVI      Errors
        BFFF0000 to BFFFFFFF     VISA     Errors
        BFFC0000 to BFFCFFFF     VXIPnP   Driver Errors
```

5.5.86. bu6100_IviLxiSync_RemoveEvent

```
ViStatus bu6100_IviLxiSync_RemoveEvent (ViSession instrumentHandle,
                                        ViConstString eventName);
```

```
Purpose

    This function removes an event.

Parameter List

    instrumentHandle

        Variable Type       ViSession

        Instrument handle.

    eventName

        Variable Type       ViConstString

        Specifies the name of the event to remove.

Return Value

        Returns the status code of this operation.  The status code  either
        indicates success or describes an error or warning condition.  You
        examine the status code from each call to an instrument driver
        function to determine if an error occurred.

        To obtain a text description of the status code, call the
        bu6100_error_message function.  To obtain additional information
        about the error condition, call the bu6100_GetError function.  To
        clear the error information from the driver, call the
        bu6100_ClearError function.

        The general meaning of the status code is as follows:

        Value                  Meaning
        -------------------------------
        0                      Success
        Positive Values        Warnings
        Negative Values        Errors

        This instrument driver returns errors and warnings defined by other
        sources.  The following table defines the ranges of additional status
        codes that this driver can return.  The table lists the different
        include files that contain the defined constants for the particular
        status codes:

        Numeric Range (in Hex)   Status Code Types
        -------------------------------------------------
        3FFA0000 to 3FFA1FFF     IVI      Warnings
        3FFF0000 to 3FFFFFFF     VISA     Warnings
        3FFC0000 to 3FFCFFFF     VXIPnP   Driver Warnings

        BFFA0000 to BFFA1FFF     IVI      Errors
        BFFF0000 to BFFFFFFF     VISA     Errors
        BFFC0000 to BFFCFFFF     VXIPnP   Driver Errors
```

5.5.87. bu6100_IviLxiSync_RemoveTriggerAlarm

```
     ViStatus bu6100_IviLxiSync_RemoveTriggerAlarm
                  (ViSession instrumentHandle, ViConstString alarmName);

 Purpose

     This function removes a trigger alarm.

     Only "ALARM0" is supported by ProDAQ 6100.

 Parameter List

     instrumentHandle

         Variable Type       ViSession

         Instrument handle.

     alarmName

         Variable Type       ViConstString

         Specifies the name of the trigger alarm to remove.

 Return Value

         Returns the status code of this operation.  The status code  either
         indicates success or describes an error or warning condition.  You
         examine the status code from each call to an instrument driver
         function to determine if an error occurred.

         To obtain a text description of the status code, call the
         bu6100_error_message function.  To obtain additional information
         about the error condition, call the bu6100_GetError function.  To
         clear the error information from the driver, call the
         bu6100_ClearError function.

         The general meaning of the status code is as follows:

         Value                  Meaning
         -------------------------------
         0                      Success
         Positive Values        Warnings
         Negative Values        Errors

         This instrument driver returns errors and warnings defined by other
         sources.  The following table defines the ranges of additional status
         codes that this driver can return.  The table lists the different
         include files that contain the defined constants for the particular
         status codes:

         Numeric Range (in Hex)   Status Code Types
         -------------------------------------------------
         3FFA0000 to 3FFA1FFF     IVI      Warnings
         3FFF0000 to 3FFFFFFF     VISA     Warnings
         3FFC0000 to 3FFCFFFF     VXIPnP   Driver Warnings

         BFFA0000 to BFFA1FFF     IVI      Errors
         BFFF0000 to BFFFFFFF     VISA     Errors
         BFFC0000 to BFFCFFFF     VXIPnP   Driver Errors
```

5.5.88. bu6100_IviLxiSync_RemoveTriggerSource

```
    ViStatus bu6100_IviLxiSync_RemoveTriggerSource
                 (ViSession instrumentHandle, ViConstString sourceName);

Purpose

    This function removes a trigger source.

Parameter List

    instrumentHandle

        Variable Type       ViSession

        Instrument handle.

    sourceName

        Variable Type       ViConstString

        Specifies the name of the trigger source to remove.

Return Value

        Returns the status code of this operation.  The status code  either
        indicates success or describes an error or warning condition.  You
        examine the status code from each call to an instrument driver
        function to determine if an error occurred.

        To obtain a text description of the status code, call the
        bu6100_error_message function.  To obtain additional information
        about the error condition, call the bu6100_GetError function.  To
        clear the error information from the driver, call the
        bu6100_ClearError function.

        The general meaning of the status code is as follows:

        Value                  Meaning
        -------------------------------
        0                      Success
        Positive Values        Warnings
        Negative Values        Errors

        This instrument driver returns errors and warnings defined by other
        sources.  The following table defines the ranges of additional status
        codes that this driver can return.  The table lists the different
        include files that contain the defined constants for the particular
        status codes:

        Numeric Range (in Hex)   Status Code Types
        -------------------------------------------------
        3FFA0000 to 3FFA1FFF     IVI      Warnings
        3FFF0000 to 3FFFFFFF     VISA     Warnings
        3FFC0000 to 3FFCFFFF     VXIPnP   Driver Warnings

        BFFA0000 to BFFA1FFF     IVI      Errors
        BFFF0000 to BFFFFFFF     VISA     Errors
        BFFC0000 to BFFCFFFF     VXIPnP   Driver Errors
```

5.5.89. bu6100_IviLxiSync_SetSystemTime

```
ViStatus bu6100_IviLxiSync_SetSystemTime (ViSession instrumentHandle,
                                          ViReal64 timeSeconds,
                                          ViReal64 timeFractional);
```

```
Purpose

    This function sets the 1588 time if the board operates as a 1588 clock
    master or as a not synchronized slave.

Parameter List

    instrumentHandle

        Variable Type       ViSession

        The Instrument Handle is used to identify the unique session or
        communication channel between the driver and the instrument.

        If more than one instrument of the same model type is used, this
        Handle will be used to differentiate between them.


    timeSeconds

        Variable Type       ViReal64

        Sets the seconds portion of the 1588 time.

    timeFractional

        Variable Type       ViReal64

        Sets the fractional portion of the 1588 time.

Return Value

        Returns the status code of this operation.  The status code  either
        indicates success or describes an error or warning condition.  You
        examine the status code from each call to an instrument driver
        function to determine if an error occurred.

        To obtain a text description of the status code, call the
        bu6100_error_message function.  To obtain additional information
        about the error condition, call the bu6100_GetError function.  To
        clear the error information from the driver, call the
        bu6100_ClearError function.

        The general meaning of the status code is as follows:

        Value                  Meaning
        -------------------------------
        0                      Success
        Positive Values        Warnings
        Negative Values        Errors

        This instrument driver returns errors and warnings defined by other
        sources.  The following table defines the ranges of additional status
        codes that this driver can return.  The table lists the different
        include files that contain the defined constants for the particular
        status codes:

        Numeric Range (in Hex)   Status Code Types
        -------------------------------------------------
        3FFA0000 to 3FFA1FFF     IVI      Warnings
        3FFF0000 to 3FFFFFFF     VISA     Warnings
        3FFC0000 to 3FFCFFFF     VXIPnP   Driver Warnings

        BFFA0000 to BFFA1FFF     IVI      Errors
        BFFF0000 to BFFFFFFF     VISA     Errors
        BFFC0000 to BFFCFFFF     VXIPnP   Driver Errors
```

5.5.90. bu6100_killList

```
ViStatus bu6100_killList (ViSession instrumentHandle,
                          ViInt16 functionCard);
```

Purpose

Kills the List currently running by DSP for the specified Function Card.

Parameter List

instrumentHandle

Variable Type	ViSession

The Instrument Handle is used to identify the unique session or communication channel between the driver and the instrument.

If more than one instrument of the same model type is used, this Handle will be used to differentiate between them.

functionCard

Variable Type	ViInt16

Specifies the Function Card for which List should be killed.

Valid Values: 1, 2, 3, 4

Default Value: 1

Return Value

Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You examine the status code from each call to an instrument driver function to determine if an error occurred.

To obtain a text description of the status code, call the bu6100_error_message function. To obtain additional information about the error condition, call the bu6100_GetError function. To clear the error information from the driver, call the bu6100_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

This instrument driver returns errors and warnings defined by other sources. The following table defines the ranges of additional status codes that this driver can return. The table lists the different include files that contain the defined constants for the particular status codes:

Numeric Range (in Hex)	Status Code Types	
3FFA0000 to 3FFA1FFF	IVI	Warnings
3FFF0000 to 3FFFFFFF	VISA	Warnings
3FFC0000 to 3FFCFFFF	VXIPnP	Driver Warnings
BFFA0000 to BFFA1FFF	IVI	Errors
BFFF0000 to BFFFFFFF	VISA	Errors
BFFC0000 to BFFCFFFF	VXIPnP	Driver Errors

5.5.91. bu6100_loadList

```
ViStatus bu6100_loadList (ViSession instrumentHandle,
                          ViInt16 functionCard, ViInt32 n_ofOperations,
                          ViInt32 _VI_FAR opIndexes[],
                          ViInt32 _VI_FAR varIndexes[],
                          ViInt32 _VI_FAR constants[]);
```

Purpose

Loads the List of Commands into theDSP's table of lists. DSP precompiles the List and store it in its internal memory.

Parameter List

instrumentHandle

Variable Type   ViSession

The Instrument Handle is used to identify the unique session or communication channel between the driver and the instrument.

If more than one instrument of the same model type is used, this Handle will be used to differentiate between them.

functionCard

Variable Type   ViInt16

Specifies the Function Card for which the List will be loaded.

Valid Values: 1, 2, 3, 4

Default Value: 1

n_ofOperations

Variable Type   ViInt32

Number of operations contained in the list to be loaded. The possible values are from 1 to BU6100_MAX_LIST_LENGTH (0x40)

Default Value: 0

opIndexes

Variable Type   ViInt32[]

This array should contain indexes of operations in the list.

varIndexes

Variable Type   ViInt32[]

This array should contain the indexes of the List variables.

constants

Variable Type   ViInt32[]

This array should contain constants.

Return Value

Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You examine the status code from each call to an instrument driver function to determine if an error occurred.

To obtain a text description of the status code, call the bu6100_error_message function. To obtain additional information about the error condition, call the bu6100_GetError function. To clear the error information from the driver, call the bu6100_ClearError function.

```
The general meaning of the status code is as follows:

Value                  Meaning
-------------------------------
0                      Success
Positive Values        Warnings
Negative Values        Errors

This instrument driver returns errors and warnings defined by other
sources.  The following table defines the ranges of additional status
codes that this driver can return.  The table lists the different
include files that contain the defined constants for the particular
status codes:

Numeric Range (in Hex)   Status Code Types
-------------------------------------------------
3FFA0000 to 3FFA1FFF     IVI      Warnings
3FFF0000 to 3FFFFFFF     VISA     Warnings
3FFC0000 to 3FFCFFFF     VXIPnP   Driver Warnings

BFFA0000 to BFFA1FFF     IVI      Errors
BFFF0000 to BFFFFFFF     VISA     Errors
BFFC0000 to BFFCFFFF     VXIPnP   Driver Errors
```

5.5.92. bu6100_LockSession

```
ViStatus bu6100_LockSession (ViSession instrumentHandle,
                             ViPBoolean callerHasLock);
```

```
Purpose

    This function obtains a multithread lock on the instrument session.
    Before it does so, it waits until all other execution threads have
    released their locks on the instrument session.

    Other threads might have obtained a lock on this session in the following
    ways:

    - The user's application called bu6100_LockSession.

    - A call to the instrument driver locked the session.

    - A call to the IVI engine locked the session.

    After your call to bu6100_LockSession returns successfully, no other
    threads can access the instrument session until you call
    bu6100_UnlockSession.

    Use bu6100_LockSession and bu6100_UnlockSession around a sequence of
    calls to instrument driver functions if you require that the instrument
    retain its settings through the end of the sequence.

    You can safely make nested calls to bu6100_LockSession within the same
    thread.  To completely unlock the session, you must balance each call to
    bu6100_LockSession with a call to bu6100_UnlockSession.  If, however, you
    use the Caller Has Lock parameter in all calls to bu6100_LockSession and
    bu6100_UnlockSession within a function, the IVI Library locks the session
    only once within the function regardless of the number of calls you make
    to bu6100_LockSession.  This allows you to call bu6100_UnlockSession just
    once at the end of the function.


Parameter List

    instrumentHandle

        Variable Type       ViSession

        The ViSession handle that you obtain from the bu6100_init or
        bu6100_InitWithOptions function.  The handle identifies a particular
        instrument session.

        Default Value:  None


    callerHasLock

        Variable Type       ViBoolean (passed by reference)

        This parameter serves as a convenience.  If you do not want to use
        this parameter, pass VI_NULL.

        Use this parameter in complex functions to keep track of whether you
        obtain a lock and therefore need to unlock the session.  Pass the
        address of a local ViBoolean variable.  In the declaration of the
        local variable, initialize it to VI_FALSE.  Pass the address of the
        same local variable to any other calls you make to bu6100_LockSession
        or bu6100_UnlockSession in the same function.

        The parameter is an input/output parameter.  bu6100_LockSession and
        bu6100_UnlockSession each inspect the current value and take the
        following actions:

        - If the value is VI_TRUE, bu6100_LockSession does not lock the
        session again.  If the value is VI_FALSE, bu6100_LockSession obtains
        the lock and sets the value of the parameter to VI_TRUE.

        - If the value is VI_FALSE, bu6100_UnlockSession does not attempt to
        unlock the session.  If the value is VI_TRUE, bu6100_UnlockSession
        releases the lock and sets the value of the parameter to VI_FALSE.
```

```
Thus, you can, call bu6100_UnlockSession at the end of your function
without worrying about whether you actually have the lock.
```

```
Example:

ViStatus TestFunc (ViSession vi, ViInt32 flags)
{
    ViStatus error = VI_SUCCESS;
    ViBoolean haveLock = VI_FALSE;

    if (flags & BIT_1)
        {
        viCheckErr( bu6100_LockSession(vi, &haveLock));
        viCheckErr( TakeAction1(vi));
        if (flags & BIT_2)
            {
            viCheckErr( bu6100_UnlockSession(vi, &haveLock));
            viCheckErr( TakeAction2(vi));
            viCheckErr( bu6100_LockSession(vi, &haveLock);
            }
        if (flags & BIT_3)
            viCheckErr( TakeAction3(vi));
        }

Error:
    /*
       At this point, you cannot really be sure that
       you have the lock.  Fortunately, the haveLock
       variable takes care of that for you.
    */
    bu6100_UnlockSession(vi, &haveLock);
    return error;
}
```

Return Value

Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You examine the status code from each call to an instrument driver function to determine if an error occurred.

To obtain a text description of the status code, call the bu6100_error_message function. To obtain additional information about the error condition, call the bu6100_GetError function. To clear the error information from the driver, call the bu6100_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

This instrument driver returns errors and warnings defined by other sources. The following table defines the ranges of additional status codes that this driver can return. The table lists the different include files that contain the defined constants for the particular status codes:

Numeric Range (in Hex)	Status Code Types	
3FFA0000 to 3FFA1FFF	IVI	Warnings
3FFF0000 to 3FFFFFFF	VISA	Warnings
3FFC0000 to 3FFCFFFF	VXIPnP	Driver Warnings
BFFA0000 to BFFA1FFF	IVI	Errors
BFFF0000 to BFFFFFFF	VISA	Errors
BFFC0000 to BFFCFFFF	VXIPnP	Driver Errors

5.5.93. bu6100_pulseTrig

```
    ViStatus bu6100_pulseTrig (ViSession instrumentHandle,
                               ViInt16 destination, ViInt16 function);

  Purpose

    Send a pulse to the specified destination.

  Parameter List

    instrumentHandle

        Variable Type       ViSession

        The Instrument Handle is used to identify the unique session or
        communication channel between the driver and the instrument.

        If more than one instrument of the same model type is used, this
        Handle will be used to differentiate between them.


    destination

        Variable Type       ViInt16

        Selects the trigger destination.

        Valid Values:
                     BU6100_FCTrigInA1      0   FC 1 Trigger Input A
                     BU6100_FCTrigInA2      1   FC 2 Trigger Input A
                     BU6100_FCTrigInA3      2   FC 3 Trigger Input A
                     BU6100_FCTrigInA4      3   FC 4 Trigger Input A

                     BU6100_FCTrigInB1      8   FC 1 Trigger Input B
                     BU6100_FCTrigInB2      9   FC 2 Trigger Input B
                     BU6100_FCTrigInB3      10  FC 3 Trigger Input B
                     BU6100_FCTrigInB4      11  FC 4 Trigger Input B

                     BU6100_lviTrigOut0     16  LXITrig0
                     BU6100_lviTrigOut1     17  LXITrig1
                     BU6100_lviTrigOut2     18  LXITrig2
                     BU6100_lviTrigOut3     19  LXITrig3
                     BU6100_lviTrigOut4     20  LXITrig4
                     BU6100_lviTrigOut5     21  LXITrig5
                     BU6100_lviTrigOut6     22  LXITrig6
                     BU6100_lviTrigOut7     23  LXITrig7

        Default Value: 0 (BU6100_FCTrigInA1)


    function

        Variable Type       ViInt16

        Specifies the function which will be performed on the selected
        trigger line.

        Valid Values:

        BU6100_TRIG_DEASSERT  0  Put the trigger line to inactive state
        BU6100_TRIG_ASSERT    1  Put the trigger line to active state
        BU6100_TRIG_PULSE     2  Deasserts, asserts and the deasserts
                                 again the trigger line, forming a short
                                 pulse.
        Default Value: 0 (BU6100_TRIG_DEASSERT)

  Return Value

        Returns the status code of this operation.  The status code  either
        indicates success or describes an error or warning condition.  You
        examine the status code from each call to an instrument driver
        function to determine if an error occurred.

        To obtain a text description of the status code, call the
```

```
bu6100_error_message function.  To obtain additional information
about the error condition, call the bu6100_GetError function.  To
clear the error information from the driver, call the
bu6100_ClearError function.
```

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

This instrument driver returns errors and warnings defined by other sources. The following table defines the ranges of additional status codes that this driver can return. The table lists the different include files that contain the defined constants for the particular status codes:

Numeric Range (in Hex)	Status Code Types	
3FFA0000 to 3FFA1FFF	IVI	Warnings
3FFF0000 to 3FFFFFFF	VISA	Warnings
3FFC0000 to 3FFCFFFF	VXIPnP	Driver Warnings
BFFA0000 to BFFA1FFF	IVI	Errors
BFFF0000 to BFFFFFFF	VISA	Errors
BFFC0000 to BFFCFFFF	VXIPnP	Driver Errors

5.5.94. bu6100_pulseTrigSynch

```
    ViStatus bu6100_pulseTrigSynch (ViSession instrumentHandle,
                                    ViInt32 triggerMask, ViInt16 function);

Purpose

    Asserts/Deasserts or Sends a pulse to the specified trigger lines.

Parameter List

    instrumentHandle

        Variable Type       ViSession

        The Instrument Handle is used to identify the unique session or
        communication channel between the driver and the instrument.

        If more than one instrument of the same model type is used, this
        Handle will be used to differentiate between them.


    triggerMask

        Variable Type       ViInt32

        Selects the trigger lines to be operated. This value is a bitmask of
        the following bits:

           Bit 0   FC 1 Trigger Input A
           ...
           Bit 3   FC 4 Trigger Input A

           Bit 8   FC 1 Trigger Input B
           ...
           Bit 11  FC 4 Trigger Input B

           Bit 16  LXITrig0
           ..
           Bit 23  LXITrig7

        Default Value: 0

    function

        Variable Type       ViInt16

        Specifies the function which will be performed on the selected
        trigger lines.

        Valid Values:

        BU6100_TRIG_DEASSERT  0  Put the trigger line to inactive state
        BU6100_TRIG_ASSERT    1  Put the trigger line to active state
        BU6100_TRIG_PULSE     2  Deasserts, asserts and the deasserts
                                 again the trigger line, forming a short
                                 pulse.
        Default Value: 0 (BU6100_TRIG_DEASSERT)

Return Value

        Returns the status code of this operation.  The status code  either
        indicates success or describes an error or warning condition.  You
        examine the status code from each call to an instrument driver
        function to determine if an error occurred.

        To obtain a text description of the status code, call the
        bu6100_error_message function.  To obtain additional information
        about the error condition, call the bu6100_GetError function.  To
        clear the error information from the driver, call the
        bu6100_ClearError function.

        The general meaning of the status code is as follows:

        Value                 Meaning
        -------------------------------
```

0	Success
Positive Values	Warnings
Negative Values	Errors

```
This instrument driver returns errors and warnings defined by other
sources.  The following table defines the ranges of additional status
codes that this driver can return.  The table lists the different
include files that contain the defined constants for the particular
status codes:

Numeric Range (in Hex)   Status Code Types
-------------------------------------------------
3FFA0000 to 3FFA1FFF     IVI      Warnings
3FFF0000 to 3FFFFFFF     VISA     Warnings
3FFC0000 to 3FFCFFFF     VXIPnP   Driver Warnings

BFFA0000 to BFFA1FFF     IVI      Errors
BFFF0000 to BFFFFFFF     VISA     Errors
BFFC0000 to BFFCFFFF     VXIPnP   Driver Errors
```

5.5.95. bu6100_readBoardTemp

```
ViStatus bu6100_readBoardTemp (ViSession instrumentHandle,
                               ViPReal64 temperature1,
                               ViPReal64 temperature2);
```

```
Purpose

    This function reads the temperature from the temperature sensors fitted
    on the module.

Parameter List

    instrumentHandle

        Variable Type       ViSession

        The Instrument Handle is used to identify the unique session or
        communication channel between the driver and the instrument.

        If more than one instrument of the same model type is used, this
        Handle will be used to differentiate between them.


    temperature1

        Variable Type       ViReal64 (passed by reference)

        This parameter return the temperature in degrees Celsius from the
        on-board temperature sensor 1.

    temperature2

        Variable Type       ViReal64 (passed by reference)

        This parameter return the temperature in degrees Celsius from the
        on-board temperature sensor 2.

Return Value

        Returns the status code of this operation.  The status code  either
        indicates success or describes an error or warning condition.  You
        examine the status code from each call to an instrument driver
        function to determine if an error occurred.

        To obtain a text description of the status code, call the
        bu6100_error_message function.  To obtain additional information
        about the error condition, call the bu6100_GetError function.  To
        clear the error information from the driver, call the
        bu6100_ClearError function.

        The general meaning of the status code is as follows:

        Value                  Meaning
        -------------------------------
        0                      Success
        Positive Values        Warnings
        Negative Values        Errors

        This instrument driver returns errors and warnings defined by other
        sources.  The following table defines the ranges of additional status
        codes that this driver can return.  The table lists the different
        include files that contain the defined constants for the particular
        status codes:

        Numeric Range (in Hex)   Status Code Types
        -------------------------------------------------
        3FFA0000 to 3FFA1FFF     IVI      Warnings
        3FFF0000 to 3FFFFFFF     VISA     Warnings
        3FFC0000 to 3FFCFFFF     VXIPnP   Driver Warnings

        BFFA0000 to BFFA1FFF     IVI      Errors
        BFFF0000 to BFFFFFFF     VISA     Errors
        BFFC0000 to BFFCFFFF     VXIPnP   Driver Errors
```

5.5.96. bu6100_readCB

```
    ViStatus bu6100_readCB (ViSession instrumentHandle,
                            ViInt16 functionCard, ViInt32 n_ofSamples,
                            ViInt32 _VI_FAR data[]);
```

```
Purpose

    This function retrieves the specified amount of data from the Circular
    Buffer for particular Function Card. The function doesn't check for data
    availability, the program should be sure that the requested data is
    available by bu6100_getCBstatus() function call.

Parameter List

    instrumentHandle

        Variable Type       ViSession

        The Instrument Handle is used to identify the unique session or
        communication channel between the driver and the instrument.

        If more than one instrument of the same model type is used, this
        Handle will be used to differentiate between them.


    functionCard

        Variable Type       ViInt16

        Specifies the Function Card for which the Circular Buffer will be
        configured.

        Valid Values: 1, 2, 3, 4

        Default Value: 1

    n_ofSamples

        Variable Type       ViInt32

        This parameter specifies the number of 32-bit words to read from the
        Circular Buffer.

    data

        Variable Type       ViInt32[]

        This parameter returns the data retrieved from the Circular Buffer.
        This array should be allocated prior to function call with
        appropriate size to hold all requested data.

Return Value

        Returns the status code of this operation.  The status code  either
        indicates success or describes an error or warning condition.  You
        examine the status code from each call to an instrument driver
        function to determine if an error occurred.

        To obtain a text description of the status code, call the
        bu6100_error_message function.  To obtain additional information
        about the error condition, call the bu6100_GetError function.  To
        clear the error information from the driver, call the
        bu6100_ClearError function.

        The general meaning of the status code is as follows:

        Value                  Meaning
        -------------------------------
        0                      Success
        Positive Values        Warnings
        Negative Values        Errors

        This instrument driver returns errors and warnings defined by other
        sources.  The following table defines the ranges of additional status
        codes that this driver can return.  The table lists the different
```

```
include files that contain the defined constants for the particular
status codes:
```

```
Numeric Range (in Hex)   Status Code Types
-------------------------------------------------
3FFA0000 to 3FFA1FFF     IVI      Warnings
3FFF0000 to 3FFFFFFF     VISA     Warnings
3FFC0000 to 3FFCFFFF     VXIPnP   Driver Warnings

BFFA0000 to BFFA1FFF     IVI      Errors
BFFF0000 to BFFFFFFF     VISA     Errors
BFFC0000 to BFFCFFFF     VXIPnP   Driver Errors
```

5.5.97. bu6100_readDram

```
ViStatus bu6100_readDram (ViSession instrumentHandle, ViInt32 offset,
                          ViInt32 count, ViInt32 _VI_FAR readData[]);
```

```
Purpose

    Reads the block of data from specified address of DRAM module.

Parameter List

    instrumentHandle

        Variable Type       ViSession

        The Instrument Handle is used to identify the unique session or
        communication channel between the driver and the instrument.

        If more than one instrument of the same model type is used, this
        Handle will be used to differentiate between them.


    offset

        Variable Type       ViInt32

        Selects the start address in DRAM module.

        Default Value: 0

    count

        Variable Type       ViInt32

        Number of elements of data (32-bit words) to read from the DRAM
        module.

        Default Value: 1

    readData

        Variable Type       ViInt32[]

        An array to receive the block of data read from the DRAM module.

        This array must be declared at least as large as the number of
        elements to be read from the instrument - failure to do so may result
        in a system failure.

Return Value

        Returns the status code of this operation.  The status code  either
        indicates success or describes an error or warning condition.  You
        examine the status code from each call to an instrument driver
        function to determine if an error occurred.

        To obtain a text description of the status code, call the
        bu6100_error_message function.  To obtain additional information
        about the error condition, call the bu6100_GetError function.  To
        clear the error information from the driver, call the
        bu6100_ClearError function.

        The general meaning of the status code is as follows:

        Value                  Meaning
        -------------------------------
        0                      Success
        Positive Values        Warnings
        Negative Values        Errors

        This instrument driver returns errors and warnings defined by other
        sources.  The following table defines the ranges of additional status
        codes that this driver can return.  The table lists the different
        include files that contain the defined constants for the particular
        status codes:
```

```
Numeric Range (in Hex)   Status Code Types
-------------------------------------------------
3FFA0000 to 3FFA1FFF     IVI      Warnings
3FFF0000 to 3FFFFFFF     VISA     Warnings
3FFC0000 to 3FFCFFFF     VXIPnP   Driver Warnings

BFFA0000 to BFFA1FFF     IVI      Errors
BFFF0000 to BFFFFFFF     VISA     Errors
BFFC0000 to BFFCFFFF     VXIPnP   Driver Errors
```

5.5.98. bu6100_ReadInstrData

```
    ViStatus bu6100_ReadInstrData (ViSession instrumentHandle,
                                   ViInt32 number_ofBytesToRead,
                                   ViChar _VI_FAR readBuffer[],
                                   ViPInt32 numBytesRead);

  Purpose

    This function reads data from the instrument.

  Parameter List

    instrumentHandle

        Variable Type       ViSession

        The ViSession handle that you obtain from the bu6100_init or
        bu6100_InitWithOptions function.  The handle identifies a particular
        instrument session.

        Default Value:  None


    number_ofBytesToRead

        Variable Type       ViInt32

        Pass the maximum number of bytes to read from the instruments.

        Valid Range:  0 to the number of elements in the Read Buffer
        Default Value: 256

    readBuffer

        Variable Type       ViChar[]

        After this function executes, this parameter contains the data that
        was read from the instrument.

    numBytesRead

        Variable Type       ViInt32 (passed by reference)

        Returns the number of bytes actually read from the instrument and
        stored in the Read Buffer.

  Return Value

        Returns the status code of this operation.  The status code  either
        indicates success or describes an error or warning condition.  You
        examine the status code from each call to an instrument driver
        function to determine if an error occurred.
        To obtain a text description of the status code, call the
        bu6100_error_message function.  To obtain additional information
        about the error condition, call the bu6100_GetError function.  To
        clear the error information from the driver, call the
        bu6100_ClearError function.

        The general meaning of the status code is as follows:

        Value                  Meaning
        -------------------------------
        0                      Success
        Positive Values        Warnings
        Negative Values        Errors

        This instrument driver returns errors and warnings defined by other
        sources.  The following table defines the ranges of additional status
        codes that this driver can return.  The table lists the different
        include files that contain the defined constants for the particular
        status codes:

        Numeric Range (in Hex)   Status Code Types
        -------------------------------------------------
        3FFA0000 to 3FFA1FFF     IVI      Warnings
        3FFF0000 to 3FFFFFFF     VISA     Warnings
        3FFC0000 to 3FFCFFFF     VXIPnP   Driver Warnings

        BFFA0000 to BFFA1FFF     IVI      Errors
        BFFF0000 to BFFFFFFF     VISA     Errors
        BFFC0000 to BFFCFFFF     VXIPnP   Driver Errors
```

5.5.99. bu6100_readSigLinesStat

```
    ViStatus bu6100_readSigLinesStat (ViSession instrumentHandle,
                                  ViPInt32 FCError,
                                  ViPInt32 FCDirectInterrupt);

  Purpose

    Returns the status of the Function Card special signal lines
    coming to the LIST processor board.

  Parameter List

    instrumentHandle

        Variable Type       ViSession

        The Instrument Handle is used to identify the unique session or
        communication channel between the driver and the instrument.

        If more than one instrument of the same model type is used, this
        Handle will be used to differentiate between them.


    FCError

        Variable Type       ViInt32 (passed by reference)

        Returns the status of the FC_DE signal lines.

        bit 0 corresponds to FC_DE line of Function Card 1
        ...
        bit 3 corresponds to FC_DE line of Function Card 4

    FCDirectInterrupt

        Variable Type       ViInt32 (passed by reference)

        Returns the status of the FC_DI signal lines.

        bit 0 corresponds to FC_DI line of Function Card 1
        ...
        bit 3 corresponds to FC_DI line of Function Card 4

  Return Value

        Returns the status code of this operation.  The status code  either
        indicates success or describes an error or warning condition.  You
        examine the status code from each call to an instrument driver
        function to determine if an error occurred.

        To obtain a text description of the status code, call the
        bu6100_error_message function.  To obtain additional information
        about the error condition, call the bu6100_GetError function.  To
        clear the error information from the driver, call the
        bu6100_ClearError function.

        The general meaning of the status code is as follows:

        Value                  Meaning
        -------------------------------
        0                      Success
        Positive Values        Warnings
        Negative Values        Errors

        This instrument driver returns errors and warnings defined by other
        sources.  The following table defines the ranges of additional status
        codes that this driver can return.  The table lists the different
        include files that contain the defined constants for the particular
        status codes:
```

```
Numeric Range (in Hex)   Status Code Types
-------------------------------------------------
3FFA0000 to 3FFA1FFF     IVI      Warnings
3FFF0000 to 3FFFFFFF     VISA     Warnings
3FFC0000 to 3FFCFFFF     VXIPnP   Driver Warnings

BFFA0000 to BFFA1FFF     IVI      Errors
BFFF0000 to BFFFFFFF     VISA     Errors
BFFC0000 to BFFCFFFF     VXIPnP   Driver Errors
```

5.5.100. bu6100_readVRTemp

```
ViStatus bu6100_readVRTemp (ViSession instrumentHandle,
                            ViPReal64 temperature);
```

Purpose

This function reads the temperature from the sensor fitted to voltage reference module.

Parameter List

instrumentHandle

Variable Type   ViSession

The Instrument Handle is used to identify the unique session or communication channel between the driver and the instrument.

If more than one instrument of the same model type is used, this Handle will be used to differentiate between them.

temperature

Variable Type   ViReal64 (passed by reference)

This parameter return the temperature in degrees Celsius.

Return Value

Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You examine the status code from each call to an instrument driver function to determine if an error occurred.

To obtain a text description of the status code, call the bu6100_error_message function. To obtain additional information about the error condition, call the bu6100_GetError function. To clear the error information from the driver, call the bu6100_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

This instrument driver returns errors and warnings defined by other sources. The following table defines the ranges of additional status codes that this driver can return. The table lists the different include files that contain the defined constants for the particular status codes:

Numeric Range (in Hex)	Status Code Types	
3FFA0000 to 3FFA1FFF	IVI	Warnings
3FFF0000 to 3FFFFFFF	VISA	Warnings
3FFC0000 to 3FFCFFFF	VXIPnP	Driver Warnings
BFFA0000 to BFFA1FFF	IVI	Errors
BFFF0000 to BFFFFFFF	VISA	Errors
BFFC0000 to BFFCFFFF	VXIPnP	Driver Errors

5.5.101. bu6100_removeIrqService

```
    ViStatus bu6100_removeIrqService (ViSession instrumentHandle,
                                      ViInt16 source);

Purpose

    Removes the IRQ watcher or IRQ handler from the Interrupt Service
    dispatching mechanism and releases the system resources.

Parameter List

    instrumentHandle

        Variable Type       ViSession

        The Instrument Handle is used to identify the unique session or
        communication channel between the driver and the instrument.

        If more than one instrument of the same model type is used, this
        Handle will be used to differentiate between them

    source

        Variable Type       ViInt16

        The trigger line for which Interrupt Service will be removed.

        Valid Values:
                     bu3100_FCTrigOutA1    0   FC 1 Trigger Output A
                     bu3100_FCTrigOutA2    1   FC 2 Trigger Output A
                     bu3100_FCTrigOutA3    2   FC 3 Trigger Output A
                     bu3100_FCTrigOutA4    3   FC 4 Trigger Output A

                     bu3100_FCTrigOutB1    8   FC 1 Trigger Output B
                     bu3100_FCTrigOutB2    9   FC 2 Trigger Output B
                     bu3100_FCTrigOutB3    10  FC 3 Trigger Output B
                     bu3100_FCTrigOutB4    11  FC 4 Trigger Output B

                     bu3100_FCCircBuf1     16  FC 1 Circular Buffer
                     bu3100_FCCircBuf2     17  FC 2 Circular Buffer
                     bu3100_FCCircBuf3     18  FC 3 Circular Buffer
                     bu3100_FCCircBuf4     19  FC 4 Circular Buffer

Return Value

        Returns the status code of this operation.  The status code  either
        indicates success or describes an error or warning condition.  You
        examine the status code from each call to an instrument driver
        function to determine if an error occurred.

        To obtain a text description of the status code, call the
        bu6100_error_message function.  To obtain additional information
        about the error condition, call the bu6100_GetError function.  To
        clear the error information from the driver, call the
        bu6100_ClearError function.

        The general meaning of the status code is as follows:

        Value                  Meaning
        -------------------------------
        0                      Success
        Positive Values        Warnings
        Negative Values        Errors

        This instrument driver returns errors and warnings defined by other
        sources.  The following table defines the ranges of additional status
        codes that this driver can return.  The table lists the different
        include files that contain the defined constants for the particular
        status codes:

        Numeric Range (in Hex)   Status Code Types
        -------------------------------------------------
        3FFA0000 to 3FFA1FFF     IVI      Warnings
        3FFF0000 to 3FFFFFFF     VISA     Warnings
        3FFC0000 to 3FFCFFFF     VXIPnP   Driver Warnings
```

```
BFFA0000 to BFFA1FFF     IVI      Errors
BFFF0000 to BFFFFFFF     VISA     Errors
BFFC0000 to BFFCFFFF     VXIPnP   Driver Errors
```

5.5.102. bu6100_reset

```
    ViStatus bu6100_reset (ViSession instrumentHandle);

Purpose

    This function resets the instrument to a known state and sends
    initialization commands to the instrument.  The initialization commands
    set instrument settings such as Headers Off, Short Command form, and Data
    Transfer Binary to the state necessary for the operation of the
    instrument driver.


Parameter List

    instrumentHandle

        Variable Type       ViSession

        The ViSession handle that you obtain from the bu6100_init or
        bu6100_InitWithOptions function.  The handle identifies a particular
        instrument session.

        Default Value:  None


Return Value

        Returns the status code of this operation.  The status code  either
        indicates success or describes an error or warning condition.  You
        examine the status code from each call to an instrument driver
        function to determine if an error occurred.

        To obtain a text description of the status code, call the
        bu6100_error_message function.  To obtain additional information
        about the error condition, call the bu6100_GetError function.  To
        clear the error information from the driver, call the
        bu6100_ClearError function.

        The general meaning of the status code is as follows:

        Value                  Meaning
        -------------------------------
        0                      Success
        Positive Values        Warnings
        Negative Values        Errors

        This instrument driver returns errors and warnings defined by other
        sources.  The following table defines the ranges of additional status
        codes that this driver can return.  The table lists the different
        include files that contain the defined constants for the particular
        status codes:

        Numeric Range (in Hex)   Status Code Types
        -------------------------------------------------
        3FFA0000 to 3FFA1FFF     IVI      Warnings
        3FFF0000 to 3FFFFFFF     VISA     Warnings
        3FFC0000 to 3FFCFFFF     VXIPnP   Driver Warnings

        BFFA0000 to BFFA1FFF     IVI      Errors
        BFFF0000 to BFFFFFFF     VISA     Errors
        BFFC0000 to BFFCFFFF     VXIPnP   Driver Errors
```

5.5.103. bu6100_ResetInterchangeCheck

```
    ViStatus bu6100_ResetInterchangeCheck (ViSession instrumentHandle);
```

Purpose

When developing a complex test system that consists of multiple test modules, it is generally a good idea to design the test modules so that they can run in any order. To do so requires ensuring that each test module completely configures the state of each instrument it uses. If a particular test module does not completely configure the state of an instrument, the state of the instrument depends on the configuration from a previously executed test module. If you execute the test modules in a different order, the behavior of the instrument and therefore the entire test module is likely to change. This change in behavior is generally instrument specific and represents an interchangeability problem.

You can use this function to test for such cases. After you call this function, the interchangeability checking algorithms in the specific driver ignore all previous configuration operations. By calling this function at the beginning of a test module, you can determine whether the test module has dependencies on the operation of previously executed test modules.

This function does not clear the interchangeability warnings from the list of previously recorded interchangeability warnings. If you want to guarantee that the bu6100_GetNextInterchangeWarning function only returns those interchangeability warnings that are generated after calling this function, you must clear the list of interchangeability warnings. You can clear the interchangeability warnings list by repeatedly calling the bu6100_GetNextInterchangeWarning function until no more interchangeability warnings are returned. If you are not interested in the content of those warnings, you can call the bu6100_ClearInterchangeWarnings function.

Parameter List

instrumentHandle

Variable Type	ViSession

The ViSession handle that you obtain from the bu6100_init or bu6100_InitWithOptions function. The handle identifies a particular instrument session.

Default Value: None

Return Value

Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You examine the status code from each call to an instrument driver function to determine if an error occurred.

To obtain a text description of the status code, call the bu6100_error_message function. To obtain additional information about the error condition, call the bu6100_GetError function. To clear the error information from the driver, call the bu6100_ClearError function.

The general meaning of the status code is as follows:

```
Value                  Meaning
-------------------------------
0                      Success
Positive Values        Warnings
Negative Values        Errors
```

This instrument driver returns errors and warnings defined by other sources. The following table defines the ranges of additional status codes that this driver can return. The table lists the different include files that contain the defined constants for the particular status codes:

```
Numeric Range (in Hex)   Status Code Types
-------------------------------------------------
3FFA0000 to 3FFA1FFF     IVI      Warnings
3FFF0000 to 3FFFFFFF     VISA     Warnings
3FFC0000 to 3FFCFFFF     VXIPnP   Driver Warnings

BFFA0000 to BFFA1FFF     IVI      Errors
BFFF0000 to BFFFFFFF     VISA     Errors
BFFC0000 to BFFCFFFF     VXIPnP   Driver Errors
```

5.5.104. bu6100_ResetWithDefaults

```
    ViStatus bu6100_ResetWithDefaults (ViSession instrumentHandle);

Purpose

    This function resets the instrument and applies initial user specified
    settings from the Logical Name which was used to initialize the session.
    If the session was created without a Logical Name, this function is
    equivalent to the bu6100_reset function.

Parameter List

    instrumentHandle

        Variable Type       ViSession

        The ViSession handle that you obtain from the bu6100_init or
        bu6100_InitWithOptions function.  The handle identifies a particular
        instrument session.

        Default Value:  None


Return Value

        Returns the status code of this operation.  The status code  either
        indicates success or describes an error or warning condition.  You
        examine the status code from each call to an instrument driver
        function to determine if an error occurred.

        To obtain a text description of the status code, call the
        bu6100_error_message function.  To obtain additional information
        about the error condition, call the bu6100_GetError function.  To
        clear the error information from the driver, call the
        bu6100_ClearError function.

        The general meaning of the status code is as follows:

        Value                  Meaning
        -------------------------------
        0                      Success
        Positive Values        Warnings
        Negative Values        Errors

        This instrument driver returns errors and warnings defined by other
        sources.  The following table defines the ranges of additional status
        codes that this driver can return.  The table lists the different
        include files that contain the defined constants for the particular
        status codes:

        Numeric Range (in Hex)   Status Code Types
        -------------------------------------------------
        3FFA0000 to 3FFA1FFF     IVI      Warnings
        3FFF0000 to 3FFFFFFF     VISA     Warnings
        3FFC0000 to 3FFCFFFF     VXIPnP   Driver Warnings

        BFFA0000 to BFFA1FFF     IVI      Errors
        BFFF0000 to BFFFFFFF     VISA     Errors
        BFFC0000 to BFFCFFFF     VXIPnP   Driver Errors
```

5.5.105. bu6100_revision_query

```
    ViStatus bu6100_revision_query (ViSession instrumentHandle,
                                    ViChar _VI_FAR instrumentDriverRevision[],
                                    ViChar _VI_FAR firmwareRevision[]);

 Purpose

    This function returns the revision numbers of the instrument driver and
    instrument firmware.



 Parameter List

    instrumentHandle

        Variable Type       ViSession

        The ViSession handle that you obtain from the bu6100_init or
        bu6100_InitWithOptions function.  The handle identifies a particular
        instrument session.

        Default Value:  None

    instrumentDriverRevision

        Variable Type       ViChar[]

        Returns the instrument driver software revision numbers in the form
        of a string.
        You must pass a ViChar array with at least 256 bytes.

    firmwareRevision

        Variable Type       ViChar[]

        Returns the instrument firmware revision numbers in the form of a
        string.
        You must pass a ViChar array with at least 256 bytes.

 Return Value

        Returns the status code of this operation.  The status code  either
        indicates success or describes an error or warning condition.  You
        examine the status code from each call to an instrument driver
        function to determine if an error occurred.

        To obtain a text description of the status code, call the
        bu6100_error_message function.  To obtain additional information
        about the error condition, call the bu6100_GetError function.  To
        clear the error information from the driver, call the
        bu6100_ClearError function.

        The general meaning of the status code is as follows:

        Value                  Meaning
        -------------------------------
        0                      Success
        Positive Values        Warnings
        Negative Values        Errors

        This instrument driver returns errors and warnings defined by other
        sources.  The following table defines the ranges of additional status
        codes that this driver can return.  The table lists the different
        include files that contain the defined constants for the particular
        status codes:

        Numeric Range (in Hex)   Status Code Types
        -------------------------------------------------
        3FFA0000 to 3FFA1FFF     IVI      Warnings
        3FFF0000 to 3FFFFFFF     VISA     Warnings
        3FFC0000 to 3FFCFFFF     VXIPnP   Driver Warnings

        BFFA0000 to BFFA1FFF     IVI      Errors
```

```
BFFF0000 to BFFFFFFF     VISA     Errors
BFFC0000 to BFFCFFFF     VXIPnP   Driver Errors
```

5.5.106. bu6100_self_test

```
ViStatus bu6100_self_test (ViSession instrumentHandle,
                           ViPInt16 selfTestResult,
                           ViChar _VI_FAR selfTestMessage[]);
```

```
Purpose

    This function runs the instrument's self test routine and returns the
    test result(s).



Parameter List

    instrumentHandle

        Variable Type       ViSession

        The ViSession handle that you obtain from the bu6100_init or
        bu6100_InitWithOptions function.  The handle identifies a particular
        instrument session.

        Default Value:  None


    selfTestResult

        Variable Type       ViInt16 (passed by reference)

        This control contains the value returned from the instrument self
        test.  Zero means success.  For any other code, see the device's
        operator's manual.

        Self-Test Code    Description
        ---------------------------------------
           0              Passed self test
           1              Self test failed



    selfTestMessage

        Variable Type       ViChar[]

        Returns the self-test response string from the instrument. See the
        device's operation manual for an explanation of the string's
        contents.

        You must pass a ViChar array with at least 256 bytes.

Return Value

        Returns the status code of this operation.  The status code  either
        indicates success or describes an error or warning condition.  You
        examine the status code from each call to an instrument driver
        function to determine if an error occurred.

        To obtain a text description of the status code, call the
        bu6100_error_message function.  To obtain additional information
        about the error condition, call the bu6100_GetError function.  To
        clear the error information from the driver, call the
        bu6100_ClearError function.

        The general meaning of the status code is as follows:

        Value                  Meaning
        -------------------------------
        0                      Success
        Positive Values        Warnings
        Negative Values        Errors

        This instrument driver returns errors and warnings defined by other
        sources.  The following table defines the ranges of additional status
        codes that this driver can return.  The table lists the different
```

```
include files that contain the defined constants for the particular
status codes:
```

```
Numeric Range (in Hex)   Status Code Types
-------------------------------------------------
3FFA0000 to 3FFA1FFF     IVI      Warnings
3FFF0000 to 3FFFFFFF     VISA     Warnings
3FFC0000 to 3FFCFFFF     VXIPnP   Driver Warnings

BFFA0000 to BFFA1FFF     IVI      Errors
BFFF0000 to BFFFFFFF     VISA     Errors
BFFC0000 to BFFCFFFF     VXIPnP   Driver Errors
```

5.5.107. bu6100_set1588config

```
ViStatus bu6100_set1588config (ViSession instrumentHandle,
                               ViInt32 enabled, ViInt32 threshold,
                               ViInt32 clockDivider, ViInt32 pid_p,
                               ViInt32 pid_i, ViInt32 pid_d);
```

Purpose

This function sets the configuration of the 1588 interface of the ProDAQ 6100 module.

Parameter List

instrumentHandle

Variable Type   ViSession

The Instrument Handle is used to identify the unique session or communication channel between the driver and the instrument.

If more than one instrument of the same model type is used, this Handle will be used to differentiate between them.

enabled

Variable Type   ViInt32

This parameter sets whether the 1588 interface of the ProDAQ 6100 module is disabled, enabled or enabled in slave-only mode.
In the slave-only mode the 6100 will never take a role of the 1588 clock master.

Possible values are:

BU6100_1588_DISABLED	0	1588 interface is disabled;
BU6100_1588_ENABLED	1	1588 interface is enabled;
BU6100_1588_ENABLED_SLAVE_ONLY	2	1588 interface is enabled as a slave-only device;

threshold

Variable Type   ViInt32

This parameter sets the synchronization thershold (in nanoseconds). It is used when the 6100 module operates as a 1588 slave device. If the offset from the 1588 Master is less then the specified threshold, the module is considered as synchronized. If the offset is bigger than the threshold, the module status will be "not synchronized".

clockDivider

Variable Type   ViInt32

This parameter sets the content of the TCLK divider of the 1588 interface.

pid_p

Variable Type   ViInt32

This parameter sets the 'p' coefficient of the PTP PID control loop.

pid_i

Variable Type   ViInt32

This parameter sets the 'i' coefficient of the PTP PID control loop.

pid_d

Variable Type   ViInt32

This parameter sets the 'd' coefficient of the PTP PID control loop.

Return Value

Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You examine the status code from each call to an instrument driver function to determine if an error occurred.

To obtain a text description of the status code, call the bu6100_error_message function. To obtain additional information about the error condition, call the bu6100_GetError function. To clear the error information from the driver, call the bu6100_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

This instrument driver returns errors and warnings defined by other sources. The following table defines the ranges of additional status codes that this driver can return. The table lists the different include files that contain the defined constants for the particular status codes:

Numeric Range (in Hex)	Status Code Types	
3FFA0000 to 3FFA1FFF	IVI	Warnings
3FFF0000 to 3FFFFFFF	VISA	Warnings
3FFC0000 to 3FFCFFFF	VXIPnP	Driver Warnings
BFFA0000 to BFFA1FFF	IVI	Errors
BFFF0000 to BFFFFFFF	VISA	Errors
BFFC0000 to BFFCFFFF	VXIPnP	Driver Errors

5.5.108. bu6100_SetAttributeViBoolean

```
ViStatus bu6100_SetAttributeViBoolean (ViSession instrumentHandle,
                                       ViChar _VI_FAR channelName[],
                                       ViAttr attributeID,
                                       ViBoolean attributeValue);
```

Purpose

This function sets the value of a ViBoolean attribute.

This is a low-level function that you can use to set the values of instrument-specific attributes and inherent IVI attributes. If the attribute represents an instrument state, this function performs instrument I/O in the following cases:

- State caching is disabled for the entire session or for the particular attribute.

- State caching is enabled and the currently cached value is invalid or is different than the value you specify.

This instrument driver contains high-level functions that set most of the instrument attributes. It is best to use the high-level driver functions as much as possible. They handle order dependencies and multithread locking for you. In addition, they perform status checking only after setting all of the attributes. In contrast, when you set multiple attributes using the SetAttribute functions, the functions check the instrument status after each call.

Also, when state caching is enabled, the high-level functions that configure multiple attributes perform instrument I/O only for the attributes whose value you change. Thus, you can safely call the high-level functions without the penalty of redundant instrument I/O.

Parameter List

instrumentHandle

Variable Type	ViSession

The ViSession handle that you obtain from the bu6100_init or bu6100_InitWithOptions function. The handle identifies a particular instrument session.

Default Value: None

channelName

Variable Type	ViChar[]

If the attribute is channel-based, this parameter specifies the name of the channel on which to set the value of the attribute. If the attribute is not channel-based, then pass VI_NULL or an empty string.

Valid Channel Names: 1

Default Value: ""

attributeID

Variable Type	ViAttr

Pass the ID of an attribute.

From the function panel window, you can use this control as follows.

- Click on the control or press <ENTER>, <spacebar>, or <ctrl-down arrow>, to display a dialog box containing a hierarchical list of the available attributes. Attributes whose value cannot be set are dim. Help text is shown for

each attribute. Select an attribute by double-clicking on it or by selecting it and then pressing <ENTER>.

Read-only attributes appear dim in the list box. If you select a read-only attribute, an error message appears.

A ring control at the top of the dialog box allows you to see all IVI attributes or only the attributes of the ViBoolean type. If you choose to see all IVI attributes, the data types appear to the right of the attribute names in the list box. Attributes with data types other than ViBoolean are dim. If you select an attribute data type that is dim, LabWindows/CVI transfers you to the function panel for the corresponding function that is consistent with the data type.

- If you want to enter a variable name, press <CTRL-T> to change this ring control to a manual input box.

- If the attribute in this ring control has named constants as valid values, you can view the constants by moving to the Attribute Value control and pressing <ENTER>.

attributeValue

Variable Type   ViBoolean

Pass the value to which you want to set the attribute.

From the function panel window, you can use this control as follows.

- If the attribute currently showing in the Attribute ID ring control has constants as valid values, you can view a list of the constants by pressing <ENTER> on this control. Select a value by double-clicking on it or by selecting it and then pressing <ENTER>.

  Note: Some of the values might not be valid depending on the current settings of the instrument session.

Default Value: none

Return Value

Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You examine the status code from each call to an instrument driver function to determine if an error occurred.

To obtain a text description of the status code, call the bu6100_error_message function. To obtain additional information about the error condition, call the bu6100_GetError function. To clear the error information from the driver, call the bu6100_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

This instrument driver returns errors and warnings defined by other sources. The following table defines the ranges of additional status codes that this driver can return. The table lists the different include files that contain the defined constants for the particular status codes:

Numeric Range (in Hex)	Status Code Types	
3FFA0000 to 3FFA1FFF	IVI	Warnings
3FFF0000 to 3FFFFFFF	VISA	Warnings
3FFC0000 to 3FFCFFFF	VXIPnP	Driver Warnings
BFFA0000 to BFFA1FFF	IVI	Errors

```
BFFF0000 to BFFFFFFF     VISA     Errors
BFFC0000 to BFFCFFFF     VXIPnP   Driver Errors
```

5.5.109. bu6100_SetAttributeViInt32

```
ViStatus bu6100_SetAttributeViInt32 (ViSession instrumentHandle,
                                     ViChar _VI_FAR channelName[],
                                     ViAttr attributeID,
                                     ViInt32 attributeValue);
```

```
Purpose

    This function sets the value of a ViInt32 attribute.

    This is a low-level function that you can use to set the values of
    instrument-specific attributes and inherent IVI attributes.  If the
    attribute represents an instrument state, this function performs
    instrument I/O in the following cases:

    - State caching is disabled for the entire session or for the particular
    attribute.

    - State caching is enabled and the currently cached value is invalid or
    is different than the value you specify.

    This instrument driver contains high-level functions that set most of the
    instrument attributes.  It is best to use the high-level driver functions
    as much as possible.  They handle order dependencies and multithread
    locking for you.  In addition, they perform status checking only after
    setting all of the attributes.  In contrast, when you set multiple
    attributes using the SetAttribute functions, the functions check the
    instrument status after each call.

    Also, when state caching is enabled, the high-level functions that
    configure multiple attributes perform instrument I/O only for the
    attributes whose value you change.  Thus, you can safely call the
    high-level functions without the penalty of redundant instrument I/O.



Parameter List

    instrumentHandle

        Variable Type       ViSession

        The ViSession handle that you obtain from the bu6100_init or
        bu6100_InitWithOptions function.  The handle identifies a particular
        instrument session.

        Default Value:  None


    channelName

        Variable Type       ViChar[]

        If the attribute is channel-based, this parameter specifies the name
        of the channel on which to set the value of the attribute. If the
        attribute is not channel-based, then pass VI_NULL or an empty string.

        Valid Channel Names:  1

        Default Value:  ""


    attributeID

        Variable Type       ViAttr

        Pass the ID of an attribute.

        From the function panel window, you can use this control as follows.

        - Click on the control or press <ENTER>, <spacebar>, or
          <ctrl-down arrow>, to display a dialog box containing a
          hierarchical list of the available attributes.  Attributes
          whose value cannot be set are dim.  Help text is shown for
          each attribute.  Select an attribute by double-clicking on it
          or by selecting it and then pressing <ENTER>.

          Read-only attributes appear dim in the list box.  If you
          select a read-only attribute, an error message appears.

          A ring control at the top of the dialog box allows you to see
          all IVI attributes or only the attributes of the ViInt32 type.
          If you choose to see all IVI attributes, the data types appear
          to the right of the attribute names in the list box.
          Attributes with data types other than ViInt32 are dim.  If
          you select an attribute data type that is dim, LabWindows/CVI
          transfers you to the function panel for the corresponding
          function that is consistent with the data type.

        - If you want to enter a variable name, press <CTRL-T> to change
          this ring control to a manual input box.

        - If the attribute in this ring control has named constants as
          valid values, you can view the constants by moving to the
          Attribute Value control and pressing <ENTER>.


    attributeValue

        Variable Type       ViInt32

        Pass the value to which you want to set the attribute.

        From the function panel window, you can use this control as follows.

        - If the attribute currently showing in the Attribute ID ring
          control has constants as valid values, you can view a list of
          the constants by pressing <ENTER> on this control.  Select a
          value by double-clicking on it or by selecting it and then
          pressing <ENTER>.

          Note:  Some of the values might not be valid depending on the
          current settings of the instrument session.

        Default Value: none

Return Value

        Returns the status code of this operation.  The status code  either
        indicates success or describes an error or warning condition.  You
        examine the status code from each call to an instrument driver
        function to determine if an error occurred.

        To obtain a text description of the status code, call the
        bu6100_error_message function.  To obtain additional information
        about the error condition, call the bu6100_GetError function.  To
        clear the error information from the driver, call the
        bu6100_ClearError function.

        The general meaning of the status code is as follows:

        Value                  Meaning
        -------------------------------
        0                      Success
        Positive Values        Warnings
        Negative Values        Errors

        This instrument driver returns errors and warnings defined by other
        sources.  The following table defines the ranges of additional status
        codes that this driver can return.  The table lists the different
        include files that contain the defined constants for the particular
        status codes:

        Numeric Range (in Hex)   Status Code Types
        -------------------------------------------------
        3FFA0000 to 3FFA1FFF     IVI      Warnings
        3FFF0000 to 3FFFFFFF     VISA     Warnings
        3FFC0000 to 3FFCFFFF     VXIPnP   Driver Warnings

        BFFA0000 to BFFA1FFF     IVI      Errors
```

```
BFFF0000 to BFFFFFFF     VISA     Errors
BFFC0000 to BFFCFFFF     VXIPnP   Driver Errors
```

5.5.110. bu6100_SetAttributeViReal64

```
    ViStatus bu6100_SetAttributeViReal64 (ViSession instrumentHandle,
                                          ViChar _VI_FAR channelName[],
                                          ViAttr attributeID,
                                          ViReal64 attributeValue);
```

Purpose

This function sets the value of a ViReal64 attribute.

This is a low-level function that you can use to set the values of instrument-specific attributes and inherent IVI attributes. If the attribute represents an instrument state, this function performs instrument I/O in the following cases:

- State caching is disabled for the entire session or for the particular attribute.

- State caching is enabled and the currently cached value is invalid or is different than the value you specify.

This instrument driver contains high-level functions that set most of the instrument attributes. It is best to use the high-level driver functions as much as possible. They handle order dependencies and multithread locking for you. In addition, they perform status checking only after setting all of the attributes. In contrast, when you set multiple attributes using the SetAttribute functions, the functions check the instrument status after each call.

Also, when state caching is enabled, the high-level functions that configure multiple attributes perform instrument I/O only for the attributes whose value you change. Thus, you can safely call the high-level functions without the penalty of redundant instrument I/O.

Parameter List

instrumentHandle

Variable Type	ViSession

The ViSession handle that you obtain from the bu6100_init or bu6100_InitWithOptions function. The handle identifies a particular instrument session.

Default Value: None

channelName

Variable Type	ViChar[]

If the attribute is channel-based, this parameter specifies the name of the channel on which to set the value of the attribute. If the attribute is not channel-based, then pass VI_NULL or an empty string.

Valid Channel Names: 1

Default Value: ""

attributeID

Variable Type	ViAttr

Pass the ID of an attribute.

From the function panel window, you can use this control as follows.

- Click on the control or press <ENTER>, <spacebar>, or <ctrl-down arrow>, to display a dialog box containing a hierarchical list of the available attributes. Attributes whose value cannot be set are dim. Help text is shown for

```
          each attribute.  Select an attribute by double-clicking on it
          or by selecting it and then pressing <ENTER>.

          Read-only attributes appear dim in the list box.  If you
          select a read-only attribute, an error message appears.

          A ring control at the top of the dialog box allows you to see
          all IVI attributes or only the attributes of the ViReal64
          type.  If you choose to see all IVI attributes, the data types
          appear to the right of the attribute names in the list box.
          Attributes with data types other than ViReal64 are dim.  If
          you select an attribute data type that is dim, LabWindows/CVI
          transfers you to the function panel for the corresponding
          function that is consistent with the data type.

        - If you want to enter a variable name, press <CTRL-T> to change
          this ring control to a manual input box.

        - If the attribute in this ring control has named constants as
          valid values, you can view the constants by moving to the
          Attribute Value control and pressing <ENTER>.


    attributeValue

        Variable Type       ViReal64

        Pass the value to which you want to set the attribute.

        From the function panel window, you can use this control as follows.

        - If the attribute currently showing in the Attribute ID ring
          control has constants as valid values, you can view a list of
          the constants by pressing <ENTER> on this control.  Select a
          value by double-clicking on it or by selecting it and then
          pressing <ENTER>.

          Note:  Some of the values might not be valid depending on the
          current settings of the instrument session.

        Default Value: none

Return Value

        Returns the status code of this operation.  The status code  either
        indicates success or describes an error or warning condition.  You
        examine the status code from each call to an instrument driver
        function to determine if an error occurred.

        To obtain a text description of the status code, call the
        bu6100_error_message function.  To obtain additional information
        about the error condition, call the bu6100_GetError function.  To
        clear the error information from the driver, call the
        bu6100_ClearError function.

        The general meaning of the status code is as follows:

        Value                  Meaning
        -------------------------------
        0                      Success
        Positive Values        Warnings
        Negative Values        Errors

        This instrument driver returns errors and warnings defined by other
        sources.  The following table defines the ranges of additional status
        codes that this driver can return.  The table lists the different
        include files that contain the defined constants for the particular
        status codes:

        Numeric Range (in Hex)   Status Code Types
        -------------------------------------------------
        3FFA0000 to 3FFA1FFF     IVI      Warnings
        3FFF0000 to 3FFFFFFF     VISA     Warnings
        3FFC0000 to 3FFCFFFF     VXIPnP   Driver Warnings

        BFFA0000 to BFFA1FFF     IVI      Errors
```

```
BFFF0000 to BFFFFFFF     VISA     Errors
BFFC0000 to BFFCFFFF     VXIPnP   Driver Errors
```

5.5.111. bu6100_SetAttributeViSession

```
ViStatus bu6100_SetAttributeViSession (ViSession instrumentHandle,
                                       ViChar _VI_FAR channelName[],
                                       ViAttr attributeID,
                                       ViSession attributeValue);
```

Purpose

This function sets the value of a ViSession attribute.

This is a low-level function that you can use to set the values of instrument-specific attributes and inherent IVI attributes. If the attribute represents an instrument state, this function performs instrument I/O in the following cases:

- State caching is disabled for the entire session or for the particular attribute.

- State caching is enabled and the currently cached value is invalid or is different than the value you specify.

This instrument driver contains high-level functions that set most of the instrument attributes. It is best to use the high-level driver functions as much as possible. They handle order dependencies and multithread locking for you. In addition, they perform status checking only after setting all of the attributes. In contrast, when you set multiple attributes using the SetAttribute functions, the functions check the instrument status after each call.

Also, when state caching is enabled, the high-level functions that configure multiple attributes perform instrument I/O only for the attributes whose value you change. Thus, you can safely call the high-level functions without the penalty of redundant instrument I/O.

Parameter List

instrumentHandle

Variable Type	ViSession

The ViSession handle that you obtain from the bu6100_init or bu6100_InitWithOptions function. The handle identifies a particular instrument session.

Default Value: None

channelName

Variable Type	ViChar[]

If the attribute is channel-based, this parameter specifies the name of the channel on which to set the value of the attribute. If the attribute is not channel-based, then pass VI_NULL or an empty string.

Valid Channel Names: 1

Default Value: ""

attributeID

Variable Type	ViAttr

Pass the ID of an attribute.

From the function panel window, you can use this control as follows.

- Click on the control or press <ENTER>, <spacebar>, or <ctrl-down arrow>, to display a dialog box containing a hierarchical list of the available attributes. Attributes whose value cannot be set are dim. Help text is shown for each attribute. Select an attribute by double-clicking on it or by selecting it and then pressing <ENTER>.

Read-only attributes appear dim in the list box. If you select a read-only attribute, an error message appears.

A ring control at the top of the dialog box allows you to see all IVI attributes or only the attributes of the ViSession type. If you choose to see all IVI attributes, the data types appear to the right of the attribute names in the list box. Attributes with data types other than ViSession are dim. If you select an attribute data type that is dim, LabWindows/CVI transfers you to the function panel for the corresponding function that is consistent with the data type.

- If you want to enter a variable name, press <CTRL-T> to change this ring control to a manual input box.

- If the attribute in this ring control has named constants as valid values, you can view the constants by moving to the Attribute Value control and pressing <ENTER>.

attributeValue

Variable Type ViSession

Pass the value to which you want to set the attribute.

From the function panel window, you can use this control as follows.

- If the attribute currently showing in the Attribute ID ring control has constants as valid values, you can view a list of the constants by pressing <ENTER> on this control. Select a value by double-clicking on it or by selecting it and then pressing <ENTER>.

  Note: Some of the values might not be valid depending on the current settings of the instrument session.

Default Value: none

Return Value

Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You examine the status code from each call to an instrument driver function to determine if an error occurred.

To obtain a text description of the status code, call the bu6100_error_message function. To obtain additional information about the error condition, call the bu6100_GetError function. To clear the error information from the driver, call the bu6100_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

This instrument driver returns errors and warnings defined by other sources. The following table defines the ranges of additional status codes that this driver can return. The table lists the different include files that contain the defined constants for the particular status codes:

Numeric Range (in Hex)	Status Code Types	
3FFA0000 to 3FFA1FFF	IVI	Warnings
3FFF0000 to 3FFFFFFF	VISA	Warnings
3FFC0000 to 3FFCFFFF	VXIPnP	Driver Warnings
BFFA0000 to BFFA1FFF	IVI	Errors

```
BFFF0000 to BFFFFFFF     VISA     Errors
BFFC0000 to BFFCFFFF     VXIPnP   Driver Errors
```

5.5.112. bu6100_SetAttributeViString

```
ViStatus bu6100_SetAttributeViString (ViSession instrumentHandle,
                                      ViChar _VI_FAR channelName[],
                                      ViAttr attributeID,
                                      ViChar _VI_FAR attributeValue[]);
```

Purpose

This function sets the value of a ViString attribute.

This is a low-level function that you can use to set the values of instrument-specific attributes and inherent IVI attributes. If the attribute represents an instrument state, this function performs instrument I/O in the following cases:

- State caching is disabled for the entire session or for the particular attribute.

- State caching is enabled and the currently cached value is invalid or is different than the value you specify.

This instrument driver contains high-level functions that set most of the instrument attributes. It is best to use the high-level driver functions as much as possible. They handle order dependencies and multithread locking for you. In addition, they perform status checking only after setting all of the attributes. In contrast, when you set multiple attributes using the SetAttribute functions, the functions check the instrument status after each call.

Also, when state caching is enabled, the high-level functions that configure multiple attributes perform instrument I/O only for the attributes whose value you change. Thus, you can safely call the high-level functions without the penalty of redundant instrument I/O.

Parameter List

instrumentHandle

Variable Type	ViSession

The ViSession handle that you obtain from the bu6100_init or bu6100_InitWithOptions function. The handle identifies a particular instrument session.

Default Value: None

channelName

Variable Type	ViChar[]

If the attribute is channel-based, this parameter specifies the name of the channel on which to set the value of the attribute. If the attribute is not channel-based, then pass VI_NULL or an empty string.

Valid Channel Names: 1

Default Value: ""

attributeID

Variable Type	ViAttr

Pass the ID of an attribute.

From the function panel window, you can use this control as follows.

- Click on the control or press <ENTER>, <spacebar>, or <ctrl-down arrow>, to display a dialog box containing a hierarchical list of the available attributes. Attributes whose value cannot be set are dim. Help text is shown for

each attribute. Select an attribute by double-clicking on it or by selecting it and then pressing <ENTER>.

Read-only attributes appear dim in the list box. If you select a read-only attribute, an error message appears.

A ring control at the top of the dialog box allows you to see all IVI attributes or only the attributes of the ViString type. If you choose to see all IVI attributes, the data types appear to the right of the attribute names in the list box. Attributes with data types other than ViString are dim. If you select an attribute data type that is dim, LabWindows/CVI transfers you to the function panel for the corresponding function that is consistent with the data type.

- If you want to enter a variable name, press <CTRL-T> to change this ring control to a manual input box.

- If the attribute in this ring control has named constants as valid values, you can view the constants by moving to the Attribute Value control and pressing <ENTER>.

attributeValue

Variable Type ViChar[]

Pass the value to which you want to set the attribute.

From the function panel window, you can use this control as follows.

- If the attribute currently showing in the Attribute ID ring control has constants as valid values, you can view a list of the constants by pressing <ENTER> on this control. Select a value by double-clicking on it or by selecting it and then pressing <ENTER>.

  Note: Some of the values might not be valid depending on the current settings of the instrument session.

Default Value: none

Return Value

Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You examine the status code from each call to an instrument driver function to determine if an error occurred.

To obtain a text description of the status code, call the bu6100_error_message function. To obtain additional information about the error condition, call the bu6100_GetError function. To clear the error information from the driver, call the bu6100_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

This instrument driver returns errors and warnings defined by other sources. The following table defines the ranges of additional status codes that this driver can return. The table lists the different include files that contain the defined constants for the particular status codes:

Numeric Range (in Hex)	Status Code Types	
3FFA0000 to 3FFA1FFF	IVI	Warnings
3FFF0000 to 3FFFFFFF	VISA	Warnings
3FFC0000 to 3FFCFFFF	VXIPnP	Driver Warnings
BFFA0000 to BFFA1FFF	IVI	Errors

```
BFFF0000 to BFFFFFFF     VISA     Errors
BFFC0000 to BFFCFFFF     VXIPnP   Driver Errors
```

5.5.113. bu6100_setConfigCB

```
ViStatus bu6100_setConfigCB (ViSession instrumentHandle,
                             ViInt16 functionCard, ViInt32 function,
                             ViInt32 FIFOAddress, ViInt32 address,
                             ViInt32 length, ViInt32 threshold);
```

Purpose

Configures the Circular Buffer for particular Function Card.

Parameter List

instrumentHandle

Variable Type ViSession

The Instrument Handle is used to identify the unique session or communication channel between the driver and the instrument.

If more than one instrument of the same model type is used, this Handle will be used to differentiate between them.

functionCard

Variable Type ViInt16

Specifies the Function Card for which the Circular Buffer will be configured.
Valid Values: 1, 2, 3, 4.

function

Variable Type ViInt32

This parameter specifies the type of access to Circular Buffer which will be performed by DSP. If user doesn't want to change the function setting, the value bu3100_NO_CHANGE (-1) should be used.

Valid Values:

bu3100_NO_CHANGE	-1	
bu3100_LIST_READ16	0	16-bit Packed block read
bu3100_LIST_READ32	1	32-bit block read
bu3100_LIST_READ64	2	64-bit block read
bu3100_LIST_WRITE16	3	16-bit packed block write
bu3100_LIST_WRITE32	4	32-bit block write
bu3100_LIST_WRITE64	5	64-bit block write

Default Value: -1

FIFOAddress

Variable Type ViInt32

This parameter specifies the Address of Function Card FIFO in the Function Card address space.If user doesn't want to change the FIFO address setting, the value bu3100_NO_CHANGE (-1) should be used.

Default Value: -1

address

Variable Type ViInt32

This parameter specifies the Starting Address of the Circular Buffer in bu6100 DRAM. If other than default Circular buffer will be used, this address might be obtained by bu6100_allocDram() function call. If user doesn't want to change the address setting, the value bu3100_NO_CHANGE (-1) should be used.

Default Value: -1

length

```
        Variable Type       ViInt32

        This parameter specifies the length of the Circular Buffer in 32-bit
        words. If user doesn't want to change the length setting, the value
        bu3100_NO_CHANGE (-1) should be used.

        Default Value: -1

    threshold

        Variable Type       ViInt32

        This parameter specifies the threshold value - the number of 32-bit
        words, after collection of which the DSP will send a signal
        (interrupt) to the VXI host. If no asynchronuous mechanism supposed
        to be used, value '0' should be written. If user doesn't want to
        change the threshold setting, the value bu3100_NO_CHANGE (-1) should
        be used.

        Default Value: -1

Return Value

        Returns the status code of this operation.  The status code  either
        indicates success or describes an error or warning condition.  You
        examine the status code from each call to an instrument driver
        function to determine if an error occurred.

        To obtain a text description of the status code, call the
        bu6100_error_message function.  To obtain additional information
        about the error condition, call the bu6100_GetError function.  To
        clear the error information from the driver, call the
        bu6100_ClearError function.

        The general meaning of the status code is as follows:

        Value                  Meaning
        -------------------------------
        0                      Success
        Positive Values        Warnings
        Negative Values        Errors

        This instrument driver returns errors and warnings defined by other
        sources.  The following table defines the ranges of additional status
        codes that this driver can return.  The table lists the different
        include files that contain the defined constants for the particular
        status codes:

        Numeric Range (in Hex)   Status Code Types
        -------------------------------------------------
        3FFA0000 to 3FFA1FFF     IVI      Warnings
        3FFF0000 to 3FFFFFFF     VISA     Warnings
        3FFC0000 to 3FFCFFFF     VXIPnP   Driver Warnings

        BFFA0000 to BFFA1FFF     IVI      Errors
        BFFF0000 to BFFFFFFF     VISA     Errors
        BFFC0000 to BFFCFFFF     VXIPnP   Driver Errors
```

5.5.114. bu6100_setFcCclk

```
ViStatus bu6100_setFcCclk (ViSession instrumentHandle,
                           ViInt16 functionCard, ViInt32 CCLKSource);
```

```
Purpose

    This function selects the source of the Common Clock (CCLK) for the given
    Function Card.

Parameter List

    instrumentHandle

        Variable Type       ViSession

        The Instrument Handle is used to identify the unique session or
        communication channel between the driver and the instrument.

        If more than one instrument of the same model type is used, this
        Handle will be used to differentiate between them.


    functionCard

        Variable Type       ViInt16

        The function card to access.
        Valid Values: 1, 2, 3, 4.

    CCLKSource

        Variable Type       ViInt32

        Specifies the source for the Function Card Common Clock.

        Valid Values:

        BU6100_CCLK_0           0  CCLK Disabled and forced to '0'
        BU6100_CCLK_1           1  CCLK Disabled and forced to '1'
        BU6100_CCLK_10          2  CCLK Connected to CLK10
        BU6100_CCLK_5           3  CCLK Connected to CLK10/2
        BU6100_CCLK_2           4  CCLK Connected to CLK10/5
        BU6100_CCLK_1588_PPP    5  CCLK Connected to IEEE1588_PPP

        BU6100_CCLK_LXI_TRG_0   8  CCLK Connected to LXI Trig 0
        BU6100_CCLK_LXI_TRG_1   9  CCLK Connected to LXI Trig 1
        BU6100_CCLK_LXI_TRG_2   10 CCLK Connected to LXI Trig 2
        BU6100_CCLK_LXI_TRG_3   11 CCLK Connected to LXI Trig 3
        BU6100_CCLK_LXI_TRG_4   12 CCLK Connected to LXI Trig 4
        BU6100_CCLK_LXI_TRG_5   13 CCLK Connected to LXI Trig 5
        BU6100_CCLK_LXI_TRG_6   14 CCLK Connected to LXI Trig 6
        BU6100_CCLK_LXI_TRG_7   15 CCLK Connected to LXI Trig 7


Return Value

        Returns the status code of this operation.  The status code  either
        indicates success or describes an error or warning condition.  You
        examine the status code from each call to an instrument driver
        function to determine if an error occurred.

        To obtain a text description of the status code, call the
        bu6100_error_message function.  To obtain additional information
        about the error condition, call the bu6100_GetError function.  To
        clear the error information from the driver, call the
        bu6100_ClearError function.

        The general meaning of the status code is as follows:

        Value                  Meaning
        -------------------------------
        0                      Success
        Positive Values        Warnings
        Negative Values        Errors
```

This instrument driver returns errors and warnings defined by other sources. The following table defines the ranges of additional status codes that this driver can return. The table lists the different include files that contain the defined constants for the particular status codes:

Numeric Range (in Hex)	Status Code Types	
3FFA0000 to 3FFA1FFF	IVI	Warnings
3FFF0000 to 3FFFFFFF	VISA	Warnings
3FFC0000 to 3FFCFFFF	VXIPnP	Driver Warnings
BFFA0000 to BFFA1FFF	IVI	Errors
BFFF0000 to BFFFFFFF	VISA	Errors
BFFC0000 to BFFCFFFF	VXIPnP	Driver Errors

5.5.115. bu6100_setVoltRefOutput

```
    ViStatus bu6100_setVoltRefOutput (ViSession instrumentHandle,
                                      ViReal64 voltage);

  Purpose

    Sets the Voltage Reference module output voltage.

  Parameter List

    instrumentHandle

        Variable Type       ViSession

        The Instrument Handle is used to identify the unique session or
        communication channel between the driver and the instrument.

        If more than one instrument of the same model type is used, this
        Handle will be used to differentiate between them.


    voltage

        Variable Type       ViReal64

        Specifies the output voltage of the Voltage Reference module.
        The available set of voltages depends on the type of Voltage
        Reference module fitted to motherboard. If module is not able
        generate requested message, the function will return error status.

        Default Value: 9.0 Volt

  Return Value

        Returns the status code of this operation.  The status code  either
        indicates success or describes an error or warning condition.  You
        examine the status code from each call to an instrument driver
        function to determine if an error occurred.

        To obtain a text description of the status code, call the
        bu6100_error_message function.  To obtain additional information
        about the error condition, call the bu6100_GetError function.  To
        clear the error information from the driver, call the
        bu6100_ClearError function.

        The general meaning of the status code is as follows:

        Value                  Meaning
        -------------------------------
        0                      Success
        Positive Values        Warnings
        Negative Values        Errors

        This instrument driver returns errors and warnings defined by other
        sources.  The following table defines the ranges of additional status
        codes that this driver can return.  The table lists the different
        include files that contain the defined constants for the particular
        status codes:

        Numeric Range (in Hex)   Status Code Types
        -------------------------------------------------
        3FFA0000 to 3FFA1FFF     IVI      Warnings
        3FFF0000 to 3FFFFFFF     VISA     Warnings
        3FFC0000 to 3FFCFFFF     VXIPnP   Driver Warnings

        BFFA0000 to BFFA1FFF     IVI      Errors
        BFFF0000 to BFFFFFFF     VISA     Errors
        BFFC0000 to BFFCFFFF     VXIPnP   Driver Errors
```

5.5.116. bu6100_startList

```
ViStatus bu6100_startList (ViSession instrumentHandle,
                           ViInt16 functionCard, ViInt16 n_ofVriables,
                           ViInt32 _VI_FAR variables[]);
```

```
Purpose

    Starts the execution of specified List on the DSP. The List must be
    loaded into DSP memory by using bu6100_loadList() function prior to
    function call. If The List uses the Circular Buffer, it should be
    configured prior this function call with bu6100_setConfigCB() function.



Parameter List

    instrumentHandle

        Variable Type       ViSession

        The Instrument Handle is used to identify the unique session or
        communication channel between the driver and the instrument.

        If more than one instrument of the same model type is used, this
        Handle will be used to differentiate between them.


    functionCard

        Variable Type       ViInt16

        Specifies the Function Card for which the List will be started.

        Valid Values: 1, 2, 3, 4

        Default Value: 1

    n_ofVriables

        Variable Type       ViInt16

        Number of arguments to be passed to the starting List.

        Default Value: 0

    variables

        Variable Type       ViInt32[]

        This array contains the arguments for the starting List.

Return Value

        Returns the status code of this operation.  The status code  either
        indicates success or describes an error or warning condition.  You
        examine the status code from each call to an instrument driver
        function to determine if an error occurred.

        To obtain a text description of the status code, call the
        bu6100_error_message function.  To obtain additional information
        about the error condition, call the bu6100_GetError function.  To
        clear the error information from the driver, call the
        bu6100_ClearError function.

        The general meaning of the status code is as follows:

        Value                  Meaning
        -------------------------------
        0                      Success
        Positive Values        Warnings
        Negative Values        Errors

        This instrument driver returns errors and warnings defined by other
        sources.  The following table defines the ranges of additional status
        codes that this driver can return.  The table lists the different
```

```
include files that contain the defined constants for the particular
status codes:
```

```
Numeric Range (in Hex)   Status Code Types
-------------------------------------------------
3FFA0000 to 3FFA1FFF     IVI      Warnings
3FFF0000 to 3FFFFFFF     VISA     Warnings
3FFC0000 to 3FFCFFFF     VXIPnP   Driver Warnings

BFFA0000 to BFFA1FFF     IVI      Errors
BFFF0000 to BFFFFFFF     VISA     Errors
BFFC0000 to BFFCFFFF     VXIPnP   Driver Errors
```

5.5.117. bu6100_synchronizeListVars

```
     ViStatus bu6100_synchronizeListVars (ViSession instrumentHandle,
                                          ViInt16 functionCard,
                                          ViInt32 space, ViInt32 nVarsRead,
                                          ViInt32 startIndexRead,
                                          ViInt32 _VI_FAR readBuffer[],
                                          ViInt32 nVarsWrite,
                                          ViInt32 startIndexWrite,
                                          ViInt32 _VI_FAR writeBuffer[]);

 Purpose

     Retrieves and sends data for the currently running List.
     This function first reads current values of List data and
     then sets the new values. It is performed in synchronous way, i.e. List
     execution will be suspended between reading data and setting of new
     values.

 Parameter List

     instrumentHandle

         Variable Type       ViSession

         The Instrument Handle is used to identify the unique session or
         communication channel between the driver and the instrument.

         If more than one instrument of the same model type is used, this
         Handle will be used to differentiate between them.


     functionCard

         Variable Type       ViInt16

         Specifies the Function Card for which the List data will be
         exchanged.

         Valid Values: 1, 2, 3, 4

         Default Value: 1

     space

         Variable Type       ViInt32

         Specifies whether local or global variables will be synchronized.

         Valid Values:

         BU6100_LIST_SPACE_LOCAL  0
         BU6100_LIST_SPACE_GLOBAL 1

         Default Value: 0 (BU6100_LIST_SPACE_LOCAL)

     nVarsRead

         Variable Type       ViInt32

         Number of variables to retrieve.

         Default Value: 0

     startIndexRead

         Variable Type       ViInt32

         Start Index of variables to be retrieved.

         Default Value: 0

     readBuffer

         Variable Type       ViInt32[]
```

```
        This array will contain data retrieved form the List.

    nVarsWrite

        Variable Type       ViInt32

        Number of variables to set.

        Default Value: 0

    startIndexWrite

        Variable Type       ViInt32

        Start Index of variables to set.

        Default Value: 0

    writeBuffer

        Variable Type       ViInt32[]

        This array should contain the values of variables to set.

Return Value

        Returns the status code of this operation.  The status code  either
        indicates success or describes an error or warning condition.  You
        examine the status code from each call to an instrument driver
        function to determine if an error occurred.

        To obtain a text description of the status code, call the
        bu6100_error_message function.  To obtain additional information
        about the error condition, call the bu6100_GetError function.  To
        clear the error information from the driver, call the
        bu6100_ClearError function.

        The general meaning of the status code is as follows:

        Value                  Meaning
        -------------------------------
        0                      Success
        Positive Values        Warnings
        Negative Values        Errors

        This instrument driver returns errors and warnings defined by other
        sources.  The following table defines the ranges of additional status
        codes that this driver can return.  The table lists the different
        include files that contain the defined constants for the particular
        status codes:

        Numeric Range (in Hex)   Status Code Types
        -------------------------------------------------
        3FFA0000 to 3FFA1FFF     IVI      Warnings
        3FFF0000 to 3FFFFFFF     VISA     Warnings
        3FFC0000 to 3FFCFFFF     VXIPnP   Driver Warnings

        BFFA0000 to BFFA1FFF     IVI      Errors
        BFFF0000 to BFFFFFFF     VISA     Errors
        BFFC0000 to BFFCFFFF     VXIPnP   Driver Errors
```

5.5.118. bu6100_UnlockSession

```
ViStatus bu6100_UnlockSession (ViSession instrumentHandle,
                               ViPBoolean callerHasLock);
```

```
Purpose

    This function releases a lock that you acquired on an instrument session
    using bu6100_LockSession.  Refer to bu6100_LockSession for additional
    information on session locks.


Parameter List

    instrumentHandle

        Variable Type       ViSession

        The ViSession handle that you obtain from the bu6100_init or
        bu6100_InitWithOptions function.  The handle identifies a particular
        instrument session.

        Default Value:  None


    callerHasLock

        Variable Type       ViBoolean (passed by reference)

        This parameter serves as a convenience.  If you do not want to use
        this parameter, pass VI_NULL.

        Use this parameter in complex functions to keep track of whether you
        obtain a lock and therefore need to unlock the session.
        Pass the address of a local ViBoolean variable.  In the declaration
        of the local variable, initialize it to VI_FALSE.  Pass the address
        of the same local variable to any other calls you make to
        bu6100_LockSession or bu6100_UnlockSession in the same function.

        The parameter is an input/output parameter.  bu6100_LockSession and
        bu6100_UnlockSession each inspect the current value and take the
        following actions:

        - If the value is VI_TRUE, bu6100_LockSession does not lock the
        session again.  If the value is VI_FALSE, bu6100_LockSession obtains
        the lock and sets the value of the parameter to VI_TRUE.

        - If the value is VI_FALSE, bu6100_UnlockSession does not attempt to
        unlock the session.  If the value is VI_TRUE, bu6100_UnlockSession
        releases the lock and sets the value of the parameter to VI_FALSE.

        Thus, you can, call bu6100_UnlockSession at the end of your function
        without worrying about whether you actually have the lock.

        Example:

        ViStatus TestFunc (ViSession vi, ViInt32 flags)
        {
            ViStatus error = VI_SUCCESS;
            ViBoolean haveLock = VI_FALSE;

            if (flags & BIT_1)
                {
                viCheckErr( bu6100_LockSession(vi, &haveLock));
                viCheckErr( TakeAction1(vi));
                if (flags & BIT_2)
                    {
                    viCheckErr( bu6100_UnlockSession(vi, &haveLock));
                    viCheckErr( TakeAction2(vi));
                    viCheckErr( bu6100_LockSession(vi, &haveLock);
                    }
                if (flags & BIT_3)
                    viCheckErr( TakeAction3(vi));
                }
        Error:
            /*
               At this point, you cannot really be sure that
               you have the lock.  Fortunately, the haveLock
               variable takes care of that for you.
            */
            bu6100_UnlockSession(vi, &haveLock);
            return error;
        }
```

Return Value

```
        Returns the status code of this operation.  The status code  either
        indicates success or describes an error or warning condition.  You
        examine the status code from each call to an instrument driver
        function to determine if an error occurred.

        To obtain a text description of the status code, call the
        bu6100_error_message function.  To obtain additional information
        about the error condition, call the bu6100_GetError function.  To
        clear the error information from the driver, call the
        bu6100_ClearError function.

        The general meaning of the status code is as follows:

        Value                  Meaning
        -------------------------------
        0                      Success
        Positive Values        Warnings
        Negative Values        Errors

        This instrument driver returns errors and warnings defined by other
        sources.  The following table defines the ranges of additional status
        codes that this driver can return.  The table lists the different
        include files that contain the defined constants for the particular
        status codes:

        Numeric Range (in Hex)   Status Code Types
        -------------------------------------------------
        3FFA0000 to 3FFA1FFF     IVI      Warnings
        3FFF0000 to 3FFFFFFF     VISA     Warnings
        3FFC0000 to 3FFCFFFF     VXIPnP   Driver Warnings

        BFFA0000 to BFFA1FFF     IVI      Errors
        BFFF0000 to BFFFFFFF     VISA     Errors
        BFFC0000 to BFFCFFFF     VXIPnP   Driver Errors
```

5.5.119. bu6100_waitIrqWatcher

```
ViStatus bu6100_waitIrqWatcher (ViSession instrumentHandle,
                                ViInt16 source,
                                ViInt32 timeoutValue_msec);
```

```
Purpose

    Waits for the Interrupt watched by the given IRQ Watcher specified number
    of milliseconds. Before use this function the IRQ Watcher has to be
    created using bu6100_installIrqWatcher() function.

Parameter List

    instrumentHandle

        Variable Type       ViSession

        The Instrument Handle is used to identify the unique session or
        communication channel between the driver and the instrument.

        If more than one instrument of the same model type is used, this
        Handle will be used to differentiate between them.


    source

        Variable Type       ViInt16

        The trigger line for which Interrupt Watcher the function will wait.

        Valid Values:
                      BU6100_FCTrigOutA1    0   FC 1 Trigger Output A
                      BU6100_FCTrigOutA2    1   FC 2 Trigger Output A
                      BU6100_FCTrigOutA3    2   FC 3 Trigger Output A
                      BU6100_FCTrigOutA4    3   FC 4 Trigger Output A

                      BU6100_FCTrigOutB1    8   FC 1 Trigger Output B
                      BU6100_FCTrigOutB2    9   FC 2 Trigger Output B
                      BU6100_FCTrigOutB3    10  FC 3 Trigger Output B
                      BU6100_FCTrigOutB4    11  FC 4 Trigger Output B

                      BU6100_FCCircBuf1     16  FC 1 Circular Buffer
                      BU6100_FCCircBuf2     17  FC 2 Circular Buffer
                      BU6100_FCCircBuf3     18  FC 3 Circular Buffer
                      BU6100_FCCircBuf4     19  FC 4 Circular Buffer

        Default Value: 0 (BU6100_FCTrigOutA1)

    timeoutValue_msec

        Variable Type       ViInt32

        Specifies the timeout in milliseconds of the wait operation.

        Default Value:  1000 ms

Return Value

        Returns the status code of this operation.  The status code  either
        indicates success or describes an error or warning condition.  You
        examine the status code from each call to an instrument driver
        function to determine if an error occurred.

        To obtain a text description of the status code, call the
        bu6100_error_message function.  To obtain additional information
        about the error condition, call the bu6100_GetError function.  To
        clear the error information from the driver, call the
        bu6100_ClearError function.

        The general meaning of the status code is as follows:

        Value                  Meaning
        -------------------------------
        0                      Success
```

```
Positive Values        Warnings
Negative Values        Errors

This instrument driver returns errors and warnings defined by other
sources.  The following table defines the ranges of additional status
codes that this driver can return.  The table lists the different
include files that contain the defined constants for the particular
status codes:

Numeric Range (in Hex)   Status Code Types
-------------------------------------------------
3FFA0000 to 3FFA1FFF     IVI      Warnings
3FFF0000 to 3FFFFFFF     VISA     Warnings
3FFC0000 to 3FFCFFFF     VXIPnP   Driver Warnings

BFFA0000 to BFFA1FFF     IVI      Errors
BFFF0000 to BFFFFFFF     VISA     Errors
BFFC0000 to BFFCFFFF     VXIPnP   Driver Errors
```

5.5.120. bu6100_waitList

```
ViStatus bu6100_waitList (ViSession instrumentHandle,
                          ViInt16 functionCard, ViInt32 timeout_ms);
```

```
Purpose

    Waits until the currently active List is terminated or timeout reached.

Parameter List

    instrumentHandle

        Variable Type       ViSession

        The Instrument Handle is used to identify the unique session or
        communication channel between the driver and the instrument.

        If more than one instrument of the same model type is used, this
        Handle will be used to differentiate between them.


    functionCard

        Variable Type       ViInt16

        The function card for which the active list will be waited.

        Valid Values: 1, 2, 3, 4

        Default Value: 1

    timeout_ms

        Variable Type       ViInt32

        Specifies the timeout of the wait operation.

        Default Value: 10 ms

Return Value

        Returns the status code of this operation.  The status code  either
        indicates success or describes an error or warning condition.  You
        examine the status code from each call to an instrument driver
        function to determine if an error occurred.

        To obtain a text description of the status code, call the
        bu6100_error_message function.  To obtain additional information
        about the error condition, call the bu6100_GetError function.  To
        clear the error information from the driver, call the
        bu6100_ClearError function.

        The general meaning of the status code is as follows:

        Value                  Meaning
        -------------------------------
        0                      Success
        Positive Values        Warnings
        Negative Values        Errors

        This instrument driver returns errors and warnings defined by other
        sources.  The following table defines the ranges of additional status
        codes that this driver can return.  The table lists the different
        include files that contain the defined constants for the particular
        status codes:

        Numeric Range (in Hex)   Status Code Types
        -------------------------------------------------
        3FFA0000 to 3FFA1FFF     IVI      Warnings
        3FFF0000 to 3FFFFFFF     VISA     Warnings
        3FFC0000 to 3FFCFFFF     VXIPnP   Driver Warnings

        BFFA0000 to BFFA1FFF     IVI      Errors
        BFFF0000 to BFFFFFFF     VISA     Errors
```

```
BFFC0000 to BFFCFFFF     VXIPnP   Driver Errors
```

5.5.121. bu6100_writeCB

```
ViStatus bu6100_writeCB (ViSession instrumentHandle,
                         ViInt16 functionCard, ViInt32 n_ofSamples,
                         ViInt32 _VI_FAR data[]);
```

```
Purpose

    This function writes the specified amount of data to the Circular Buffer
    for particular Function Card.

Parameter List

    instrumentHandle

        Variable Type       ViSession

        The Instrument Handle is used to identify the unique session or
        communication channel between the driver and the instrument.

        If more than one instrument of the same model type is used, this
        Handle will be used to differentiate between them.


    functionCard

        Variable Type       ViInt16

        Specifies the Function Card for which the Circular Buffer will be
        configured.

        Valid Values: 1, 2, 3, 4

        Default Value: 1

    n_ofSamples

        Variable Type       ViInt32

        This parameter specifies the number of 32-bit words to be written to
        the Circular Buffer.

    data

        Variable Type       ViInt32[]

        This array should contain the data to be written to the Circular
        Buffer.

Return Value

        Returns the status code of this operation.  The status code  either
        indicates success or describes an error or warning condition.  You
        examine the status code from each call to an instrument driver
        function to determine if an error occurred.

        To obtain a text description of the status code, call the
        bu6100_error_message function.  To obtain additional information
        about the error condition, call the bu6100_GetError function.  To
        clear the error information from the driver, call the
        bu6100_ClearError function.

        The general meaning of the status code is as follows:

        Value                  Meaning
        -------------------------------
        0                      Success
        Positive Values        Warnings
        Negative Values        Errors

        This instrument driver returns errors and warnings defined by other
        sources.  The following table defines the ranges of additional status
        codes that this driver can return.  The table lists the different
        include files that contain the defined constants for the particular
        status codes:
```

```
Numeric Range (in Hex)   Status Code Types
-------------------------------------------------
3FFA0000 to 3FFA1FFF     IVI      Warnings
3FFF0000 to 3FFFFFFF     VISA     Warnings
3FFC0000 to 3FFCFFFF     VXIPnP   Driver Warnings

BFFA0000 to BFFA1FFF     IVI      Errors
BFFF0000 to BFFFFFFF     VISA     Errors
BFFC0000 to BFFCFFFF     VXIPnP   Driver Errors
```

5.5.122. bu6100_writeDram

```
    ViStatus bu6100_writeDram (ViSession instrumentHandle, ViInt32 offset,
                               ViInt32 count, ViInt32 _VI_FAR writeData[]);

Purpose

    Writes the data to the specified address of DRAM module.

Parameter List

    instrumentHandle

        Variable Type       ViSession

        The Instrument Handle is used to identify the unique session or
        communication channel between the driver and the instrument.

        If more than one instrument of the same model type is used, this
        Handle will be used to differentiate between them.

    offset

        Variable Type       ViInt32

        Selects the start address in DRAM module.

        Default Value: 0

    count

        Variable Type       ViInt32

        Number of elements of data (32-bit words) to write to the DRAM
        module.

        Default Value: 1

    writeData

        Variable Type       ViInt32[]

        An array of data to write to the DRAM module.

Return Value

        Returns the status code of this operation.  The status code  either
        indicates success or describes an error or warning condition.  You
        examine the status code from each call to an instrument driver
        function to determine if an error occurred.

        To obtain a text description of the status code, call the
        bu6100_error_message function.  To obtain additional information
        about the error condition, call the bu6100_GetError function.  To
        clear the error information from the driver, call the
        bu6100_ClearError function.

        The general meaning of the status code is as follows:

        Value                  Meaning
        -------------------------------
        0                      Success
        Positive Values        Warnings
        Negative Values        Errors

        This instrument driver returns errors and warnings defined by other
        sources.  The following table defines the ranges of additional status
        codes that this driver can return.  The table lists the different
        include files that contain the defined constants for the particular
        status codes:

        Numeric Range (in Hex)   Status Code Types
        -------------------------------------------------
        3FFA0000 to 3FFA1FFF     IVI      Warnings
        3FFF0000 to 3FFFFFFF     VISA     Warnings
        3FFC0000 to 3FFCFFFF     VXIPnP   Driver Warnings
```

```
BFFA0000 to BFFA1FFF     IVI      Errors
BFFF0000 to BFFFFFFF     VISA     Errors
BFFC0000 to BFFCFFFF     VXIPnP   Driver Errors
```

5.5.123. bu6100_WriteInstrData

```
    ViStatus bu6100_WriteInstrData (ViSession instrumentHandle,
                                    ViChar _VI_FAR writeBuffer[]);

Purpose

    This function writes a user-specified string to the instrument.

    Note:  This function bypasses IVI attribute state caching.  Therefore,
    when you call this function, the cached values for all attributes will be
    invalidated.

Parameter List

    instrumentHandle

        Variable Type       ViSession

        The ViSession handle that you obtain from the bu6100_init or
        bu6100_InitWithOptions function.  The handle identifies a particular
        instrument session.

        Default Value:  None


    writeBuffer

        Variable Type       ViChar[]

        Pass the string to be written to the instrument.

Return Value

        Returns the status code of this operation.  The status code  either
        indicates success or describes an error or warning condition.  You
        examine the status code from each call to an instrument driver
        function to determine if an error occurred.

        To obtain a text description of the status code, call the
        bu6100_error_message function.  To obtain additional information
        about the error condition, call the bu6100_GetError function.  To
        clear the error information from the driver, call the
        bu6100_ClearError function.

        The general meaning of the status code is as follows:

        Value                  Meaning
        -------------------------------
        0                      Success
        Positive Values        Warnings
        Negative Values        Errors

        This instrument driver returns errors and warnings defined by other
        sources.  The following table defines the ranges of additional status
        codes that this driver can return.  The table lists the different
        include files that contain the defined constants for the particular
        status codes:

        Numeric Range (in Hex)   Status Code Types
        -------------------------------------------------
        3FFA0000 to 3FFA1FFF     IVI      Warnings
        3FFF0000 to 3FFFFFFF     VISA     Warnings
        3FFC0000 to 3FFCFFFF     VXIPnP   Driver Warnings

        BFFA0000 to BFFA1FFF     IVI      Errors
        BFFF0000 to BFFFFFFF     VISA     Errors
        BFFC0000 to BFFCFFFF     VXIPnP   Driver Errors
```

5.6. IVI-C Driver Attributes

Attribute Information for the Following Functions:

```
        bu6100_SetAttributeViInt32
        bu6100_GetAttributeViInt32
        bu6100_CheckAttributeViInt32
        bu6100_SetAttributeViReal64
        bu6100_GetAttributeViReal64
        bu6100_CheckAttributeViReal64
        bu6100_SetAttributeViSession
        bu6100_GetAttributeViSession
        bu6100_CheckAttributeViSession
        bu6100_SetAttributeViBoolean
        bu6100_GetAttributeViBoolean
        bu6100_CheckAttributeViBoolean
        bu6100_SetAttributeViString
        bu6100_GetAttributeViString
        bu6100_CheckAttributeViString

   LxiSync Attributes
      Arm
         Arm Count                         BU6100_ATTR_IVILXISYNC_ARM_COUNT
         LxiSync Arm Delay                 BU6100_ATTR_IVILXISYNC_ARM_DELAY
         Alarm
            LxiSync Arm Alarm Count        BU6100_ATTR_IVILXISYNC_ARM_ALARM_COUNT
            LxiSync Arm Alarm Enabled      BU6100_ATTR_IVILXISYNC_ARM_ALARM_ENABLED
            LxiSync Arm Alarm Period       BU6100_ATTR_IVILXISYNC_ARM_ALARM_PERIOD
            LxiSync Arm Alarm Repaet Count BU6100_ATTR_IVILXISYNC_ARM_ALARM_REPEAT_COUNT
            LxiSync Arm Alarm Time Seconds BU6100_ATTR_IVILXISYNC_ARM_ALARM_TIME_SECONDS
            LxiSync Arm Alarm Time Fraction BU6100_ATTR_IVILXISYNC_ARM_ALARM_TIME_FRACTION
         Source
            LxiSync Arm Source Count       BU6100_ATTR_IVILXISYNC_ARM_SOURCE_COUNT
            LxiSync Arm Source Detection   BU6100_ATTR_IVILXISYNC_ARM_SOURCE_DETECTION
            LxiSync Arm Source Enabled     BU6100_ATTR_IVILXISYNC_ARM_SOURCE_ENABLED
            LxiSync Arm Source Event ID    BU6100_ATTR_IVILXISYNC_ARM_SOURCE_EVENTID
            LxiSync Arm Source Filter      BU6100_ATTR_IVILXISYNC_ARM_SOURCE_FILTER
            LxiSync Arm Source Or Enabled  BU6100_ATTR_IVILXISYNC_ARM_SOURCE_OR_ENABLED
      Trigger
         LxiSync Trigger Count             BU6100_ATTR_IVILXISYNC_TRIGGER_COUNT
         LxiSync Trigger Source            BU6100_ATTR_IVILXISYNC_TRIGGER_SOURCE
         Alarm
            LxiSync Trigger Alarm Count    BU6100_ATTR_IVILXISYNC_TRIGGER_ALARM_COUNT
            LxiSync Trigger Alarm Enabled  BU6100_ATTR_IVILXISYNC_TRIGGER_ALARM_ENABLED
            LxiSync TRigger Alarm Period   BU6100_ATTR_IVILXISYNC_TRIGGER_ALARM_PERIOD
            LxiSync Trigger Alarm Repeat Count BU6100_ATTR_IVILXISYNC_TRIGGER_ALARM_REPEAT_COUNT
            LxiSync Trigger Alarm Time Seconds BU6100_ATTR_IVILXISYNC_TRIGGER_ALARM_TIME_SECONDS
            LxiSync Trigger Alarm Time Fractional
BU6100_ATTR_IVILXISYNC_TRIGGER_ALARM_TIME_FRACTION
         Source
            LxiSync Trigger Source Count   BU6100_ATTR_IVILXISYNC_TRIGGER_SOURCE_COUNT
            LxiSync Trigger Source Delay   BU6100_ATTR_IVILXISYNC_TRIGGER_SOURCE_DELAY
            LxiSync Trigger Source Detection BU6100_ATTR_IVILXISYNC_TRIGGER_SOURCE_DETECTION
            LxiSync Trigger Source Event ID BU6100_ATTR_IVILXISYNC_TRIGGER_SOURCE_EVENTID
            LxiSync Trigger Source Filter  BU6100_ATTR_IVILXISYNC_TRIGGER_SOURCE_FILTER
      Event
         LxiSync Event Count               BU6100_ATTR_IVILXISYNC_EVENT_COUNT
         LxiSync Event Destination Path    BU6100_ATTR_IVILXISYNC_EVENT_DESTINATION_PATH
         LxiSync Event DRive Mode          BU6100_ATTR_IVILXISYNC_EVENT_DRIVE_MODE
         LxiSync Event Slope               BU6100_ATTR_IVILXISYNC_EVENT_SLOPE
         LxiSync Event Source              BU6100_ATTR_IVILXISYNC_EVENT_SOURCE
         LxiSync Event Wired OR Bias Mode  BU6100_ATTR_IVILXISYNC_EVENT_WIRED_OR_BIAS_MODE
      Event Log
         LxiSync Event Log Entry Count     BU6100_ATTR_IVILXISYNC_EVENT_LOG_ENTRY_COUNT
         LxiSync Event Log Enabled         BU6100_ATTR_IVILXISYNC_EVENT_LOG_ENABLED
      Time
         LxiSync Is Time Master            BU6100_ATTR_IVILXISYNC_IS_TIME_MASTER
         LxiSync Is Time Synchronized      BU6100_ATTR_IVILXISYNC_IS_TIME_SYNCHRONIZED
   Inherent IVI Attributes
      User Options
         Range Check                       BU6100_ATTR_RANGE_CHECK
         Query Instrument Status           BU6100_ATTR_QUERY_INSTRUMENT_STATUS
         Cache                             BU6100_ATTR_CACHE
         Simulate                          BU6100_ATTR_SIMULATE
         Record Value Coercions                    BU6100_ATTR_RECORD_COERCIONS
         Interchange Check                         BU6100_ATTR_INTERCHANGE_CHECK
      Driver Identification
         Description                               BU6100_ATTR_SPECIFIC_DRIVER_DESCRIPTION
         Driver Prefix                             BU6100_ATTR_SPECIFIC_DRIVER_PREFIX
         Driver Vendor                             BU6100_ATTR_SPECIFIC_DRIVER_VENDOR
         Revision                                  BU6100_ATTR_SPECIFIC_DRIVER_REVISION
         Class Specification Major Version    BU6100_ATTR_SPECIFIC_DRIVER_CLASS_SPEC_MAJOR_VERSION
         Class Specification Minor Version    BU6100_ATTR_SPECIFIC_DRIVER_CLASS_SPEC_MINOR_VERSION
      Driver Capabilities
         Supported Instrument Models               BU6100_ATTR_SUPPORTED_INSTRUMENT_MODELS
         Class Group Capabilities                  BU6100_ATTR_GROUP_CAPABILITIES
      Instrument Identification
         Manufacturer                              BU6100_ATTR_INSTRUMENT_MANUFACTURER
         Model                                     BU6100_ATTR_INSTRUMENT_MODEL
         Firmware Revision                         BU6100_ATTR_INSTRUMENT_FIRMWARE_REVISION
      Advanced Session Information
         Logical Name                              BU6100_ATTR_LOGICAL_NAME
         I/O Resource Descriptor                   BU6100_ATTR_IO_RESOURCE_DESCRIPTOR
         Driver Setup                              BU6100_ATTR_DRIVER_SETUP
   Miscellaneous Attributes
      ID Query Response                            BU6100_ATTR_ID_QUERY_RESPONSE
```

```
BU6100_ATTR_CACHE
Data Type:     ViBoolean
Description:   Specifies whether to cache the value of attributes.  When
caching is enabled, the instrument driver keeps track of the current instrument
settings and avoids sending redundant commands to the instrument.  Thus, you
can significantly increase execution speed.
    The instrument driver can choose always to cache or never to cache
particular attributes regardless of the setting of this attribute.
    The default value is VI_TRUE.  Use the bu6100_InitWithOptions function to
override this value.


BU6100_ATTR_DRIVER_SETUP
Data Type:     ViString
Restrictions:  Not settable.
Description:   Some cases exist where you must specify instrument driver
options at initialization time.  An example of this is specifying a particular
instrument model from among a family of instruments that the driver supports.
This is useful when using simulation.  You can specify driver-specific options
through the DriverSetup keyword in the optionsString parameter to the
bu6100_InitWithOptions function.  If you open an instrument using a logical
name, you can also specify the options through the IVI Configuration Utility.
    The default value is an empty string.


BU6100_ATTR_GROUP_CAPABILITIES
Data Type:     ViString
Restrictions:  Not settable.
Description:   A string that contains a comma-separated list of class-extension
groups that this driver implements.


BU6100_ATTR_ID_QUERY_RESPONSE
Data Type:     ViString
Restrictions:  Not settable.
Description:   Returns the ID Query response string.  The instrument driver
gets the value of this attribute when you pass VI_TRUE for the ID Query
parameter to the bu6100_init or bu6100_InitWithOptions function.


BU6100_ATTR_INSTRUMENT_FIRMWARE_REVISION
Data Type:     ViString
Restrictions:  Not settable.
Description:   A string that contains the firmware revision information for the
instrument you are currently using.


BU6100_ATTR_INSTRUMENT_MANUFACTURER
Data Type:     ViString
Restrictions:  Not settable.
Description:   A string that contains the name of the instrument manufacturer
you are currently using.
```

```
BU6100_ATTR_INSTRUMENT_MODEL
Data Type:     ViString
Restrictions:  Not settable.
Description:   A string that contains the model number or name of the
instrument that you are currently using.


BU6100_ATTR_INTERCHANGE_CHECK
Data Type:     ViBoolean
Description:   Specifies whether to perform interchangeability checking and
retrieve interchangeability warnings.

The default value is VI_FALSE.

Interchangeability warnings indicate that using your application with a
different instrument might cause different behavior.  You call bu6100 Get Next
Interchange Warning to extract interchange warnings.  Call the
bu6100_ClearInterchangeWarnings function to clear the list of
interchangeability warnings without reading them.
Interchangeability checking logs a warning for each attribute you have not set
that affects the behavior of the instrument.


BU6100_ATTR_IO_RESOURCE_DESCRIPTOR
Data Type:     ViString
Restrictions:  Not settable.
Description:   Indicates the resource descriptor the driver uses to identify
the physical device.
    If you initialize the driver with a logical name, this attribute contains
the resource descriptor that corresponds to the entry in the IVI Configuration
utility.
    If you initialize the instrument driver with the resource descriptor, this
attribute contains that value.


BU6100_ATTR_IVILXISYNC_ARM_ALARM_COUNT
Data Type:     ViInt32
Description:   Returns the number of arm alarms created with Arm Add Alarm


BU6100_ATTR_IVILXISYNC_ARM_ALARM_ENABLED
Data Type:     ViBoolean
Description:   Enables or disables the arm alarm.


BU6100_ATTR_IVILXISYNC_ARM_ALARM_PERIOD
Data Type:     ViReal64
Description:   Specifies the period of the arm alarm in seconds; that is, the
amount of time in seconds that transpire before the alarm repeats. A period of
zero means there is no repeat ans a single alarm generated.


BU6100_ATTR_IVILXISYNC_ARM_ALARM_REPEAT_COUNT
Data Type:     ViInt32
Description:   Specifies the number of times the trigger will occur at the
period specified by the Arm Alarm Period attribute. If Arm Alarm Repeat Counter
is zero, then the alarm shall repeat forever at the Arm Alarm Period.


BU6100_ATTR_IVILXISYNC_ARM_ALARM_TIME_FRACTION
Data Type:     ViReal64
Description:   Specifies the fractional portion of time at which the alarm will
go off. Note that the actual time of the alarm is the sum of Arm Alarm Time
Seconds and Arm Alarm Time Fraction. The time is specified as the sum of two
values because a single double-precision floating-point number does not have
sufficient range and resolution to specify the time.
Once the alarm goes off, it will repeat at the period set by Arm Alarm Period
the number of times set by Arm Alarm Count.


BU6100_ATTR_IVILXISYNC_ARM_ALARM_TIME_SECONDS
Data Type:     ViReal64
Description:   Specifies the seconds portion of time at which the alarm will go
off. Note that the actual time of the alarm is the sum of Arm Alarm Time
Seconds and Arm Alarm Time Fraction. The time is specified as the sum of two
```

values because a single double-precision floating-point number does not have
sufficient range and resolution to specify the time.
Once the alarm goes off, it will repeat at the period set by Arm Alarm Period
the number of times set by Arm Alarm Count.

BU6100_ATTR_IVILXISYNC_ARM_COUNT
Data Type: ViInt32
Description: Specifies the number of times the arm has to occur to complete
the arm loop; that is, the number of arms that are accepted before the
measurement must be initiated again.

BU6100_ATTR_IVILXISYNC_ARM_DELAY
Data Type: ViReal64
Description: Specifies the delay from when the arm logic satisfied until the
waiting for the trigger state is enetered. The units are seconds

BU6100_ATTR_IVILXISYNC_ARM_SOURCE_COUNT
Data Type: ViInt32
Restrictions: Not settable.
Description: Returns the number of currently available arm sources.

BU6100_ATTR_IVILXISYNC_ARM_SOURCE_DETECTION
Data Type: ViInt32
Description: Specifies the style of arm source detection.
If the source is a LAN event and the source detection is set to rise, this Arm
repeated capability will be satisfied when the designated LAN packet arrives
with a True indication. If the source detection is set to fall, this Arm
repeated capability will be satisfied when a LAN packet arrives with a False
indication. If the detection is set to high, the source will be satisfied when
the designated LAN packet arrives with a True indication and remain satisfied
until the designated LAN packet arrives with a False indication.   If the
detection is to low, the source will be satisfied when the designated LAN
packet arrives with a False indication and remain satisfied until the
designated LAN packet arrives with a True indication.
Defined values:
bu6100_VAL_IVILXISYNC_DETECTION_RISE - Configures the LXI device to arm on the
rising edge of the arm source.
bu6100_VAL_IVILXISYNC_DETECTION_FALL - Configures the LXI device to arm on the
falling edge of the arm source.
bu6100_VAL_IVILXISYNC_DETECTION_HIGH - Configures the LXI device to arm while
the arm source is high, that is, while it remains true
bu6100_VAL_IVILXISYNC_DETECTION_LOW - Configures the LXI device to arm while
the arm source is low, that is, while it remains low

BU6100_ATTR_IVILXISYNC_ARM_SOURCE_ENABLED
Data Type: ViBoolean
Description: Enables or disables the arm source. If a source is disabled, it
has no affect on the summary arm signal.

BU6100_ATTR_IVILXISYNC_ARM_SOURCE_EVENTID
Data Type: ViString
Description: This specifies the LAN event identifier that is associated with
this arm source. LAN Events with this identifier are accepted from the source
described in the filter.
The default value for EventId is the repeated capability specifier for this arm
source.

BU6100_ATTR_IVILXISYNC_ARM_SOURCE_FILTER
Data Type: ViString
Description: Specifies a filter for restricting arm sources. The filter
specified by this attribute denotes the accepted sources. The syntax for
specifying a filter is as follows:
<Filter> == [(<tcp> | <udp> | <any>) [, <Filter>]]
<tcp> == <host> [:<port>]
<udp> == ALL [: <port>]
<any> == : <port>
<host> is either a hostname or host number. Note that the hostname can not be
"ALL" since that would indicate the <udp> construct.
<port> is a series of decimal digits indicating the port number.
Specifying an empty string or VI_NULL means that LXI arm packets are accepted

via either TCP or UDP multicast from any host. Note that ":5044" is equivalent to the empty string since 5044 is the IANA registered port for LXI events (lxi-evntsvc).

In the <tcp>, <udp> and <any> constructs, <port> refers to the port the device receives the LAN message on. If <port> is omitted from <tcp> or <udp>, packets are only accepted on the IANA registered port for LXI events (lxi-evntsvc).

Specifying the <host> (<tcp> construct) indicates that packets via TCP on the port indicated are accepted.

Specifying ALL (<udp> construct) indicates that UDP multicast packets are accepted if they are directed to the IANA registered port for LXI events (lxi-evntsvc) on the IANA registered multicast address (LXI-EVENT). No TCP packets are accepted unless a <tcp> syntax is also included in the filter. The multicast address can not be altered with this syntax.

Specifying any protocol (<any> construct) indicates that both packets via TCP and UDP multicast packets are accepted if they are directed to the specified port. UDP multicast packets must be received at the IANA registered multicast addres (LXI-EVENT).

The send port is not monitored. This allows the transmitter to use any available port.

Drivers (and the corresponding instruments) that support this syntax are permitted to not support all possible filters syntaxes.

White space shall be ignored. The <Filter> string is case insensitive.

Conventional devices should consider restricting the <port> to only the IANA registered port for LXI events (lxi-evntsvc) and not accepting the generalized syntax.

```
BU6100_ATTR_IVILXISYNC_ARM_SOURCE_OR_ENABLED
Data Type:     ViBoolean
Description:   Enables or disables the OR-summing of the arm sources.  When set
to True, the arm sources use OR-summing.  When set to False, the arm sources
use AND-summing.
```

```
BU6100_ATTR_IVILXISYNC_EVENT_COUNT
Data Type:     ViInt32
Restrictions:  Not settable.
Description:   Returns the number of defined events.  The count returned
includes any of the supported reserved repeated capability names defined in
Reserved Repeated Capability Identifiers as well as any custom repeated
capability identifiers.
```

```
BU6100_ATTR_IVILXISYNC_EVENT_DESTINATION_PATH
Data Type:     ViString
Description:   Specifies a list of places to send the event.
```

The default value for this attribute is the repeated capability name.
The grammar for the parameter is:

```
<DestinationPath>== [(<tcp>|<udp>|<TriggerBus>)[, <DestinationPath>]]
<tcp> == <host> [: <port>] [/<LANIdent>]
<udp> == [ALL] [: <port> ] [/<LANIdent>]
<TriggerBus> == LXI0|LXI1|LXI2|LXI3|LXI4|LXI5|LXI6|LXI7
```

host is either a hostname or host number, and port is a series of decimal digits indicating the port number. Note that the hostname can not be "ALL" or one of the <TriggerBus> designations since that would indicate the <udp> or <TriggerBus> construct.
<LANIdent> is a string indicating the LAN identifier that will be sent in the LAN message. The <LANIdent> is not case sensitive. <LANIdent> is from one to 16 ASCII characters inclusive. The characters may be numeric or underscore or hyphen or upper or lower-case alphabetic characters.
Defaults:
The default <DestinationPath> is the repeated capability name. This may either be a <TriggerBus> identifier or a <LANIdent> identifier.
<LANIdent> defaults to the repeated capability name.
<port> defaults to the IANA registered port for LXI events (lxi-evntsvc).
If the repeated capability name is not a trigger bus specifier then the default <DestinationPath> is ‘ALL’ with the <LANIdent> as the repeated capability name.

If the repeated capability name corresponds to a <TriggerBus>, the default <DestinationPath> is the repeated capability name.
If multiple <DestinationPath>s are specified, the event is transmitted to each.
The <tcp> construct specifies that a TCP message will be sent to the destination when the bound event occurs.
The <udp> construct specifies that a UDP multicast message will be sent to the IANA registered multi-cast address (LXI-EVENT) on the designated port. UDP unicast and UDP broadcasts are not supported by this syntax.
The <TriggerBus> construct specifies that a physical LXI wired trigger bus is used to signal the event.
Note that the LXI specification reserves event identifiers that begin with the characters "LXI" for LXI use. The strings "LXI0", "LXI1", ... ,"LXI7" refer to the 8 LXI wired trigger bus triggers.
White space shall be ignored. The <Destination> string is case insensitive.
Drivers may accept additional vendor-defined syntaxes
Drivers (and the corresponding instruments) that support this syntax are permitted to not support all possible destination syntaxes.
Conventional devices should consider restricting the port to only the IANA registered port for LXI events (lxi-evntsvc) and not accepting the generalized syntax.

BU6100_ATTR_IVILXISYNC_EVENT_DRIVE_MODE
Data Type: ViInt32
Description: Specifies how this event is transmitted

It is an error to turn on the Wired OR Bias Mode for this device for a particular LXI trigger line and then set the Event Enabled attribute to On instead of Wired OR for an event whose destination path includes that LXI trigger line.

BU6100_ATTR_IVILXISYNC_EVENT_LOG_ENABLED
Data Type: ViBoolean
Description: Enables or disables the event logging feature.

BU6100_ATTR_IVILXISYNC_EVENT_LOG_ENTRY_COUNT
Data Type: ViInt32
Restrictions: Not settable.
Description: Returns the number of event log entries available.

BU6100_ATTR_IVILXISYNC_EVENT_SLOPE
Data Type: ViInt32
Description: Specifies the slope of the event that is inbound to the event subsystem that will cause the generation of an outbound event. The outbound event shall be transmitted with the same slope as the inbound event.
Possible values are:
bu6100_VAL_IVILXISYNC_SLOPE_POSITIVE - The event will be transmitted with a rising edge.
bu6100_VAL_IVILXISYNC_SLOPE_NEGATIVE - The event will be transmitted with a falling edge.

BU6100_ATTR_IVILXISYNC_EVENT_SOURCE
Data Type: ViString
Description: Specifies the signal which causes an event to be transmitted. This attribute is case-insensitive but case-preserving.

BU6100_ATTR_IVILXISYNC_EVENT_WIRED_OR_BIAS_MODE
Data Type: ViInt32
Description: Specifies whether this LXI device will serve as the wired-OR bias for specific LXI trigger bus lines.
The allowed values for this attribute are 0 to 255. This attribute is a bit field, where bit 0 represents LXI0, bit 1 represents LXI1, and so on. A value of one in a particular bit indicates that the LXI device shall serve as the bias for the corresponding trigger bus line. A value of zero in a particular bit disables the bias for the corresponding trigger bus line. To use a trigger bus line in driven mode, the bias must be disabled.
Enabling wired-OR bias has no impact on the device’s ability to either respond to signals on trigger bus lines or to send events on trigger bus lines.
One and only one LXI device can serve as the wired-OR bias for a particular trigger bus line, although different devices can serve as the wired-OR bias for different trigger bus lines.

```
BU6100_ATTR_IVILXISYNC_IS_TIME_MASTER
Data Type:     ViBoolean
Restrictions:  Not settable.
Description:   Indicates if this device is the 1588 master.


BU6100_ATTR_IVILXISYNC_IS_TIME_SYNCHRONIZED
Data Type:     ViBoolean
Restrictions:  Not settable.
Description:   Indicates if the device is synchronized.


BU6100_ATTR_IVILXISYNC_TRIGGER_ALARM_COUNT
Data Type:     ViInt32
Restrictions:  Not settable.
Description:   Returns the number of currently available trigger alarms.  The
count returned includes the reserved repeated capability named "ALARM0" as well
as any custom repeated capability identifiers.


BU6100_ATTR_IVILXISYNC_TRIGGER_ALARM_ENABLED
Data Type:     ViBoolean
Description:   Enables or disables the trigger alarm.


BU6100_ATTR_IVILXISYNC_TRIGGER_ALARM_PERIOD
Data Type:     ViReal64
Description:   Specifies the period of the trigger alarm in seconds; that is,
the amount of time in seconds that transpire before the alarm repeats.  A
period of zero means there is no repeat and a single trigger is generated.


BU6100_ATTR_IVILXISYNC_TRIGGER_ALARM_REPEAT_COUNT
Data Type:     ViInt32
Description:   Specifies the number of times the trigger will occur at the
period specified by the Trigger Alarm Period attribute.  If Trigger Alarm
Repeat Period is non-zero and Trigger Alarm Repeat Count is zero, then the
alarm shall repeat forever at the Trigger Alarm Period.

bu6100_VAL_IVILXISYNC_REPEAT_CONTINUOUS is provided to set the repeat count to
forever.


BU6100_ATTR_IVILXISYNC_TRIGGER_ALARM_TIME_FRACTION
Data Type:     ViReal64
Description:   Specifies the fractional seconds portion of the time at which
the alarm will go off.  Note that the actual time of the alarm is the sum of
Trigger Alarm Time Seconds and Trigger Alarm Time Fraction.  The time is
specified as the sum of two values because a single double-precision
floating-point does not have sufficient range and resolution to specify the
time.

Once the alarm goes off, it will repeat at the period set by Trigger Alarm
Period the number of times set by Trigger Alarm Count.


BU6100_ATTR_IVILXISYNC_TRIGGER_ALARM_TIME_SECONDS
Data Type:     ViReal64
Description:   Specifies the seconds portion of the time at which the alarm
will go off.  Note that the actual time of the alarm is the sum of Trigger
Alarm Time Seconds and Trigger Alarm Time Fraction.  The time is specified as
the sum of two values because a single double-precision floating-point does not
have sufficient range and resolution to specify the time.

Once the alarm goes off, it will repeat at the period set by Trigger Alarm
Period the number of times set by Trigger Alarm Count.


BU6100_ATTR_IVILXISYNC_TRIGGER_COUNT
Data Type:     ViInt32
Description:   Specifies the number of times a trigger has to occur to complete
a measurement; that is, the number of triggers that are accepted before the
measurement must be armed again.
```

```
BU6100_ATTR_IVILXISYNC_TRIGGER_SOURCE
Data Type:     ViString
Description:   Specifies which of the available trigger sources to use as the
signal for triggering the device-specific operation (for example, a
measurement).
The value specified for this attribute may be one of the names in the
IviLxiSyncTriggerSource repeated capability collection as returned from the
GetTriggerSourceName function.
The value specified for this attribute may also be one of the names in the
IviLxiSyncTriggerAlarm repeated capability collection as returned from the
GetTriggerAlarmName function.
The name specified here may also correspond to a non-LXI trigger event.  For
instance, the caller can use this attribute to program the trigger source to
external or immediate triggering, by specifying values such as "EXT" or "INT".
Such trigger source names are device-dependent.
If the device trigger source has been programmed to a non-LXI event using an
attribute or function other than the Trigger Source attribute, then this
attribute shall return that value when read.  For instance, if the specific
driver implements an IVI instrument class and the class-compliant API is used
to set the trigger source to external, then this property shall return a string
that reflects the value set through the class-compliant API.
This attribute is case-insensitive but case-preserving.  For more information
on this requirement,


BU6100_ATTR_IVILXISYNC_TRIGGER_SOURCE_COUNT
Data Type:     ViInt32
Restrictions:  Not settable.
Description:   Returns the number of currently available trigger sources.  The
count returned includes any of the supported reserved repeated capability names
defined in Section 2.1.6, Reserved Repeated Capability Identifiers as well as
any custom repeated capability identifiers.


BU6100_ATTR_IVILXISYNC_TRIGGER_SOURCE_DELAY
Data Type:     ViReal64
Description:   Specifies the trigger source delay from when the trigger logic
is satisfied until the device specific action (for instance a measurement) is
triggered.  A negative value implies pre-trigger acquisition. The units are
seconds.


BU6100_ATTR_IVILXISYNC_TRIGGER_SOURCE_DETECTION
Data Type:     ViInt32
Description:   Specifies the slope of the trigger source.
If the source is a LAN event and the source slope is set to positive, this
Trigger repeated capability will be satisfied when the designated LAN packet
arrives with a true indication.  If the source slope is set to negative, this
Trigger repeated capability will be satisfied when a LAN packet arrives with a
false indication.

Popssible values are:
bu6100_VAL_IVILXISYNC_DETECTION_RISE - Configures the LXI device to trigger on
the rising edge of the trigger source.
bu6100_VAL_IVILXISYNC_DETECTION_FALL - Configures the LXI device to trigger on
the falling edge of the trigger source.


BU6100_ATTR_IVILXISYNC_TRIGGER_SOURCE_EVENTID
Data Type:     ViString
Description:   This specifies the LAN event identifier that is associated with
this trigger source.  LAN Events with this identifier are accepted from  the
source described in the filter.
The default value for EventId is the repeated capability specifier for this
trigger source.


BU6100_ATTR_IVILXISYNC_TRIGGER_SOURCE_FILTER
Data Type:     ViString
Description:   Specifies a filter for restricting trigger sources.  The filter
specified by this attribute denotes the accepted sources. The syntax for
specifying a filter is as follows:
<Filter> == [( <tcp> | <udp> | <any>) [, <Filter> ]]
<tcp> == <host> [:<port>]
<udp> == ALL  [: <port>]
<any> == : <port>
<host> is either a hostname or host number.  Note that the hostname can not be
```

“ALL” since that would indicate the <udp> construct.
<port> is a series of decimal digits indicating the port number.
Specifying an empty string or VI_NULL means that LXI trigger packets are accepted via either TCP or UDP multicast from any host. “:5044” is equivalent to the empty string since 5044 is the IANA registered port for LXI events (lxi-evntsvc).

In the <tcp>, <udp> and <any> constructs, <port> refers to the port the device receives the LAN message on. If <port> is omitted from <tcp> or <udp>, packets are only accepted on the IANA registered port for LXI events (lxi-evntsvc).

Specifying the <host> (<tcp> construct) indicates that packets via TCP on the port indicated are accepted.

Specifying ALL (<udp> construct) indicates that UDP multicast packets are accepted if they are directed to the IANA registered port for LXI events (lxi-evntsvc) on the IANA registered multicast address (LXI-EVENT). No TCP packets are accepted unless a <tcp> syntax is also included in the filter. The multicast address can not be altered with this syntax.

Specifying any protocol (<any> construct) indicates that both TCP and UDP multicast packets are accepted if they are directed to the specified port. UDP multicast packets must be received at the IANA registered multicast addres (LXI-EVENT).

The send port is not monitored. This allows the transmitter to use any available port.

Drivers (and the corresponding instruments) that support this syntax are permitted to not support all possible filters syntaxes.
White space shall be ignored. The <Filter> string is case insensitive.
Conventional devices should consider restricting the <port> to only the IANA registered port for LXI events (lxi-eventsvc) and not accepting the generalized syntax.

BU6100_ATTR_LOGICAL_NAME
Data Type: ViString
Restrictions: Not settable.
Description: A string containing the logical name you specified when opening the current IVI session.
You may pass a logical name to the bu6100_init or bu6100_InitWithOptions functions. The IVI Configuration utility must contain an entry for the logical name. The logical name entry refers to a virtual instrument section in the IVI Configuration file. The virtual instrument section specifies a physical device and initial user options.

BU6100_ATTR_QUERY_INSTRUMENT_STATUS
Data Type: ViBoolean
Description: Specifies whether the instrument driver queries the instrument status after each operation. Querying the instrument status is very useful for debugging. After you validate your program, you can set this attribute to VI_FALSE to disable status checking and maximize performance
The instrument driver can choose to ignore status checking for particular attributes regardless of the setting of this attribute.
The default value is VI_FALSE. Use the bu6100_InitWithOptions function to override this value.

BU6100_ATTR_RANGE_CHECK
Data Type: ViBoolean
Description: Specifies whether to validate attribute values and function parameters. If enabled, the instrument driver validates the parameter values that you pass to driver functions. Range checking parameters is very useful for debugging. After you validate your program, you can set this attribute to VI_FALSE to disable range checking and maximize performance.
The default value is VI_TRUE. Use the bu6100_InitWithOptions function to override this value.

BU6100_ATTR_RECORD_COERCIONS
Data Type: ViBoolean
Description: Specifies whether the IVI engine keeps a list of the value coercions it makes for integer and real type attributes. You call bu6100 Get Next Coercion Record to extract and delete the oldest coercion record from the

```
list.
    The default value is VI_FALSE.  Use the bu6100_InitWithOptions function to
override this value.


BU6100_ATTR_SIMULATE
Data Type:     ViBoolean
Description:   Specifies whether or not to simulate instrument driver I/O
operations.  If simulation is enabled, instrument driver functions perform
range checking and call Ivi_GetAttribute and Ivi_SetAttribute functions, but
they do not perform instrument I/O.  For output parameters that represent
instrument data, the instrument driver functions return calculated values.
    The default value is VI_FALSE.  Use the bu6100_InitWithOptions function to
override this value.


BU6100_ATTR_SPECIFIC_DRIVER_CLASS_SPEC_MAJOR_VERSION
Data Type:     ViInt32
Restrictions:  Not settable.
Description:   The major version number of the class specification with which
this driver is compliant.


BU6100_ATTR_SPECIFIC_DRIVER_CLASS_SPEC_MINOR_VERSION
Data Type:     ViInt32
Restrictions:  Not settable.
Description:   The minor version number of the class specification with which
this driver is compliant.


BU6100_ATTR_SPECIFIC_DRIVER_DESCRIPTION
Data Type:     ViString
Restrictions:  Not settable.
Description:   A string that contains a brief description of the specific
driver.


BU6100_ATTR_SPECIFIC_DRIVER_PREFIX
Data Type:     ViString
Restrictions:  Not settable.
Description:   A string that contains the prefix for the instrument driver.
The name of each user-callable function in this driver starts with this prefix.


BU6100_ATTR_SPECIFIC_DRIVER_REVISION
Data Type:     ViString
Restrictions:  Not settable.
Description:   A string that contains additional version information about this
instrument driver.


BU6100_ATTR_SPECIFIC_DRIVER_VENDOR
Data Type:     ViString
Restrictions:  Not settable.
Description:   A string that contains the name of the vendor that supplies this
driver.
```

bustec