Shipped From

Contact Us

Email support@xeostech.com
Phone (902) 444 7650
Fax (902) 444 7651
Website www.xeostech.com

Version History

<table>
<thead>
<tr>
<th>Version No.</th>
<th>Firmware Version</th>
<th>Date</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.7</td>
<td>1.1.4310</td>
<td>Apr 2018</td>
<td>Rough Manual Release</td>
</tr>
<tr>
<td>0.8</td>
<td>1.1.4310</td>
<td>July 2018</td>
<td>Combined Firmware/Hardware Manuals</td>
</tr>
<tr>
<td>1.0</td>
<td>1.1.4483</td>
<td>July 2018</td>
<td>Document Overhaul</td>
</tr>
<tr>
<td>1.1</td>
<td>1.1.4761</td>
<td>August 2018</td>
<td>New Serial Commands Added</td>
</tr>
</tbody>
</table>

This manual is written with respect to the Firmware aspects of the Brizo and is meant to be used in combination with the Brizo Hardware User Manual.
# Table of Contents

Shipped From.......................................................................................................................... i
Contact Us.............................................................................................................................. i
Version History...................................................................................................................... i
Device Theory of Operation.................................................................................................... 3
  GNSS Receiver Theory of Operation .................................................................................. 3
  SD Cards Theory of Operation........................................................................................... 3
  Tunnel Theory of Operation............................................................................................... 4
  SBD Theory of Operation................................................................................................. 5
Device Configuration.................................................................................................................. 7
USB Diagnostic Port................................................................................................................ 7
Using STerm and Sending Commands ................................................................................... 8
Configuration Commands ....................................................................................................... 9
  Timer Methodology............................................................................................................. 9
  Wave Height Measurement Configuration ...................................................................... 9
  Wave Height Measurement Output Interface .............................................................. 10
  Tunnel Configuration....................................................................................................... 10
  SBD and SMS Configuration........................................................................................... 11
Connecting and Operating over the Tunnel .......................................................................... 11
  Automatic File Retrieval (Auto-Downloader)................................................................. 11
  Sending SBDs Using XeosOnline..................................................................................... 12
XeosOnline.............................................................................................................................. 12
Retrieving Data ....................................................................................................................... 13
  SBD Transmissions......................................................................................................... 13
  Serial (RS-232)............................................................................................................. 13
Upgrading Firmware................................................................................................................ 14
Appendix A: Command Index .............................................................................................. 16
Appendix B: Serial Message Formats................................................................................... 18
Appendix C: RxTools Download ............................................................................................ 18
Warranty, Support and Limited Liability................................................................................ 19
Device Theory of Operation

The Brizo at its heart is an embedded peripheral control unit that utilizes Global Navigation Satellite System (GNSS) to measure wave statistics. All enabled peripherals, including the GNSS are controlled by a scheduler. The scheduler will enable and disable the GNSS at the commanded interval. After collecting enough data for the measurement period, the device will calculate the wave coefficients and output them over the Serial-2 connection or enabled telemetry.

GNSS Receiver Theory of Operation

The GNSS receiver schedule is controlled by the configured wave height sample rate. The GNSS outputs configurable Septentrio Binary Format (SBF) blocks, which the processor uses to calculate the wave coefficients. Raw GNSS measurements can be recorded to the SD Cards if desired. The GNSS will power on and off to save power if an ongoing measurement period is not occurring.

After power up, the GNSS receiver waits to receive GNSS signal before starting calculations. GNSS signals are not available indoors. This means the device will not attempt to transmit messages over Iridium, Cell, Radio or Serial until it has GNSS signal.

SD Cards Theory of Operation

All collected data is initially written to an SD Card. Diagnostics are also optionally written to the second SD Card. While the file is being written to the file that is ‘open’ on the device, this makes it important to notify the device when you are shutting it down.

Proper shut down etiquette is to either send the device a $resetnow command or to hold the test button or 3+ seconds. Failure to do so will cause the file to be left open. Open files are partially recoverable on a computer provided they are not written over. Files written to the SD card use a naming convention with an ASCII representation of BASE-36 characters with the following name format:

\[ SSFYYMDDH.sbf \]

<table>
<thead>
<tr>
<th>SS</th>
<th>Two-character representation of device serial number</th>
</tr>
</thead>
<tbody>
<tr>
<td>F</td>
<td>Stream Source (1-4), D (Diagnostics))</td>
</tr>
<tr>
<td>YY</td>
<td>Two-character representation of year</td>
</tr>
<tr>
<td>M</td>
<td>Month of year</td>
</tr>
<tr>
<td>D</td>
<td>Day of Month</td>
</tr>
<tr>
<td>H</td>
<td>Hour of Day</td>
</tr>
</tbody>
</table>

Example: 2T10FATC.sbf

<table>
<thead>
<tr>
<th>Serial #</th>
<th>2T =101</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stream</td>
<td>1 =1</td>
</tr>
</tbody>
</table>
The GNSS SBF files follow the format outlined in the Septentrio SBF Reference Guide. SBF files can be converted to Rinex or ASCII using the SBF Converter from Septentrio included in the free RxTools software suite linked below:

[https://manuals.alertgeo.com/RxTools/](https://manuals.alertgeo.com/RxTools/)

**Tunnel Theory of Operation**

If your unit is equipped with a Cellular modem or Iridium modem and a data or RUDICS plan, your unit is capable of remotely connecting to the Xeos Tunnel. The Tunnel is useful for configuring your unit remotely along with retrieving data and diagnostics from the SD card remotely. It is also useful for diagnosing potential issues on units deployed in the field.

Upon reaching its start time, the tunnel will power on the modem configured and attempt to open a two way connection to XeosOnline (XO). The tunnel confirms two way operation to XO via a heartbeat message known as a “keep-alive”. A keep-alive is a small data packet sent from the unit to the XO server every 15 seconds and acknowledged by the server. This “keep-alive” manages the connection between the unit and XO irrespective of any sockets being opened by
the user or a script. If two packets are unacknowledged in a row, the unit closes the connection and tries to reestablish it.

Keep-alive data packets include Status of Health (SOH) information about the receiver, including voltage and temperature conditions. Keep alive data packets are also the best way to verify the tunnel is connected when your unit is in a remote location. If keep-alives are coming in to the Message Log of the unit every 15 seconds, the tunnel is connected.

Keep-alive data packets include Status of Health (SOH) information about the receiver, including voltage and temperature conditions. Keep alive data packets are also the best way to verify the tunnel is connected when your unit is in a remote location. If keep-alives are coming in to the Message Log of the unit every 15 seconds, the tunnel is connected.

Tunnel connections over cellular are normally stable (in good cellular coverage). Tunnel connections over Iridium RUDICS will have interruptions under normal operation. The maximum data transfer rate of Iridium RUDICS is 17 kb/min.

The user of the remote device can interact with the device through a number of interface (sockets) into the tunnel. These include an embedded HTTP (Web) Server, automatic download scripts and Telnet Server. It is possible to open only one socket of the tunnel at a time. Multiple open connections to the tunnel can result in the tunnel disconnecting. The tunnel will reconnect after disconnecting providing the tunnel end time has not been reached. See the Brizo Hardware Manual for information on setting up Iridium Service.

**Iridium RUDICS will interfere with your GNSS measurements if the antenna placement is not properly managed. See hardware manual for details.**

**SBD Theory of Operation**

SBDs are a low cost, low power method of communicating remotely with your unit if you have an Iridium modem installed and are the Iridium equivalent of a text message (SMS). The Brizo schedules SBD message checks using the scheduler.

If Iridium is the enabled telemetry of choice, an SBD message will be automatically scheduled and sent to XeosOnline with the calculated wave coefficients. This is timed to avoid GNSS interference. At this point the modem will also check for messages waiting at the Iridium gateway. Additional SBD message checks can be scheduled, but may interfere with the GNSS if the antenna placement is not properly managed.
SBDs cannot be sent at the same time as RUDICS tunnels are scheduled. If an SBD message check is scheduled during a tunnel connection it will be executed upon the tunnel reaching its end time.

Any messages sent by the user between then last message check and the next message check will wait at the Iridium gateway for the unit to check in.

The unit requires an unlock code in order for any command sent to the device to be acted upon. This is to prevent undesired command and control. Unlock codes are sent out to the provisioned addresses of the modem on startup and each SBD after. Unlock codes are automatically included in SBDs appearing in XeosOnline’s message log.
Device Configuration

All of the settings on the Brizo can be configured either by connecting one of the diagnostic ports to a serial port terminal, a USB port, or by sending configuration commands from a remote location over the Iridium Satellite Network or Cellular SMS (telemetry options needed). If configuring the unit locally, USB is recommended.

USB Diagnostic Port

All the settings on the Brizo can be configured by connecting the USB diagnostic port to a USB port on your computer.

If using Windows to connect to the USB diagnostic port, you may need to download a driver for the STM32 USB chip contained in the unit. The driver can be found here:


Modern Linux and Mac operating systems already support the driver.

After downloading and installing the driver, connect the provided USB mini cable between the Brizo and your computer. Your computer should assign the Brizo a COM port number (Windows only), that will show up under ports in the Windows Device Manager. If the device is not recognized, check the driver the device is implementing.

To communicate with the Brizo you will need a serial port terminal program. Xeos Technologies recommends STerm, a Windows-only product. A free alternative for Linux and Mac is CoolTerm. STerm and related configuration files can be downloaded here:

https://manuals.alertgeo.com/STerm

The port settings for the Brizo USB are:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baud Rate</td>
<td>Any</td>
</tr>
<tr>
<td>Data Bits</td>
<td>8</td>
</tr>
<tr>
<td>Parity</td>
<td>None</td>
</tr>
<tr>
<td>Stop Bits</td>
<td>1</td>
</tr>
<tr>
<td>Flow Control</td>
<td>Disabled</td>
</tr>
</tbody>
</table>
Using STerm and Sending Commands

STerm is the officially supported serial port terminal program of Xeos Technologies. STerm can load configuration files to allow the commands for Xeos Technologies products to appear as buttons on the right side of the window. This can be done through the FILE>OPEN menu and opening the Brizo Config File.stm file. This is available to download here:

https://manuals.alertgeo.com/STerm

To open a connection to your device, power it on and wait for the computer to recognize the device driver. Select the COM port of your device on the Serial Port Settings menu in the bottom right corner of the screen. Click the Open Port button. You should now be connected to the unit. The buttons that appear on the right side of the window can be clicked and will send commands to your unit.

![Figure 3: An STerm Window](image)

Figure 4: STerm Serial Port Settings Menu

The buttons on the right side of the window can be edited to change the input variable you are sending to the device. This is done by hovering over the right side of the button then clicking the green edit button. Type your changes into the button then move your mouse away to end the edit. Remember to save your changes by navigating to File > Save.
Configuration Commands

While there are many commands outlined in this section, it is likely that your unit is mostly configured from the factory.

Timer Methodology

The timers on the device are powered by an onboard battery backed up by a real time clock (RTC). The RTC is updated from GNSS time to compensate for drift. The RTC continues to tick while the device is powered off and operates in the UTC timeframe.

All peripherals other than the wave height calculations are controlled by three separate timer settings: duration, interval and offset.

‘Duration’ is the length of time the peripheral is on.

‘Interval’ is the how often the device comes on.

‘Offset’ offsets the interval by the specified amount of time.

When adjusting these settings, the time input can take the form of seconds (default if no units are entered), minutes, hours or days (For example, 3 hours would be entered as “3h”. This command accepts “m” as minutes, “h” as hours and “d” as days). The timer input can only be input with one unit. If you wish the duration to be infinite (always on), simply set the duration time to “0”.

All peripherals and calculations can be disabled using “$(function)SetEnabled F”. For example:

\[ $\text{WaveheightSetEnabled F} \]

Wave Height Measurement Configuration

Wave height calculations and measurements do not rely on the timers as other functions and peripherals do. The wave height sample interval can be configured with the following command:

\[ $\text{WaveheightSetSampleRate [number]} \]

This command sets the number of measurement periods per hour, not the sample rate of the measurement during the measuring period. For instance to configure the measurement rate to be 3 times per hour the following command would be used:

\[ $\text{WaveheightSetSampleRate 3} \]

The only valid options for this command are 1, 2 or 3. The number or periods per hour can be confirmed with the following command:

\[ $\text{WaveheightGetSampleRate} \]
Wave Height Measurement Output Interface

The output interface can currently be configured using

\$WaveHeightSetResultInterface [bitmap]

The current options for the bitmap are:

[1] Iridium SBD
[2] Output over serial

Other telemetry options available upon request. To set the unit to output results over Iridium for example the following command would be sent.

\$WaveheightSetResultInterface 1

The output interface can be confirmed with the \$WaveHeightGetResultInterface command without a number.

Tunnel Configuration

The tunnel timers are useful if you have either a Cellular or Iridium modem with a RUDICS connection. The unit should be set from factory to know which modem to prioritize when attempting tunnel connections. The priority list can be changed using:

\$TunnelSetModem [priority] [modem]

Priority 0 is the highest priority while 3 is the lowest. Modems have the following corresponding numbers for this command:

[2] Cellular
[3] Iridium

To prioritize the cell modem before the Iridium modem the following commands would be entered:

\$TunnelSetModem 0 2
\$TunnelSetModem 1 3

Tunnel timers operate the same as the GNSS timers, with duration, interval and offset. The commands are as follows:

\$TunnelSetDuration [time]
\$TunnelSetInterval [time]
\$TunnelSetOffset [time]
SBD and SMS Configuration

If your unit is equipped with cellular and/or Iridium modems, then it is possible to use SBD (Iridium) and/or SMS (cellular) to interrogate or configure the unit.

Additional SBD and SMS message checks can be configured individually using the interval and offset timer commands. Messages with the calculated data will be automatically configured based on the set sample rate. **It is not recommended to change the SBD duration timer from default.** The commands to configure the timers are below and operate on the same principals outlined in detail in the GNSS section:

```plaintext
$SmsSetInterval [time]  
$SmsSetOffset [time]     
$SbdSetInterval [time]   
$SbdSetOffset [time]     
```

SBD and SMS capable units will send a power-up message when the unit first turns on. This is useful for confirming when deploying in the field, and notifying users of remote stations of unexpected power loss.

SBD and SMS cannot occur simultaneously to their respective tunnels. If an SBD or SMS is scheduled during a tunnel session, a flag is set and the SBD or SMS will occur at the conclusion of the tunnel.

Connecting and Operating over the Tunnel

It is possible to connect to only one interface (socket) of the tunnel at a time. Multiple connections to the tunnel can result in the tunnel disconnecting. The tunnel will reconnect after disconnecting providing the tunnel end time has not been reached. See the Hardware Manual for information on setting up Iridium Service.

Automatic File Retrieval (Auto-Downloader)

If you wish to retrieve files from the embedded HTTP server on a consistent basis, it is possible to run a script to download all new files on the unit for retrieval via ftp client. The script will automatically start downloading files for the duration of the tunnel session. **This is especially recommended for retrieving files over Iridium.** This script acts as your one socket connection. Xeos Technologies must enable and disable the script at this time.

Contact Xeos Technologies for further details.
Sending SBDs Using XeosOnline

After logging into XeosOnline, click on “FILE>SEND”. A new window will pop up in your browser.

The left side of the window will have a list of all the units in your XeosOnline organization in the form of the Serial Number or other identification name followed by an (IMEI). IMEI’s are identifying numbers of Iridium modems.

Select which units you want to send SBDs to (you can send SBDs to multiple units at once) by moving them into the right hand selected column by using the arrow keys, or dragging the name of the device to the right side of the window.

Type out the commands you wish to send to your unit in the commands column at the bottom of the window.

Include the $ ahead of the commands, leave white space or return carriages after commands.

Click Send to distribute the SBDs to the Iridium gateway. When the unit receives the command(s) it will respond to the command(s) via SBD.

Only single line responses are sent via SBD. Any command that responds in a multiline list will not reply with the list. Both Outgoing and incoming messages can be viewed in the messages tab on XeosOnline.

![Send Message Window](image)

**FIGURE 6: XEOSONLINE SEND MESSAGE WINDOW**

XeosOnline

Contact Xeos Technologies to set up an organization in XeosOnline. XeosOnline can be used to view the state of the unit’s Tunnel Connection, Status of Health and for easily sending and
receiving SBD messages to units configured with the correct modems. Units with Iridium SBD can also view the wave parameters on Xeos Online under the device tab.

**Retrieving Data**

The Brizo includes multiple interfaces to retrieve data. This section outlines briefly how to access that data. It is important to note the GNSS requires a clear view of the sky to begin calculating wave parameters and no message will be transmitted if the unit antenna cannot view the sky or collect the GNSS signals (for example the unit is indoors).

**SBD Transmissions**

SBD Transmissions can be accessed on Xeos Online. Log in to Xeos Online and click on the Wave Height Tab to display wave parameters and status of health information. Tabular data can be downloaded as CSV or excel files or displayed graphically on Xeos Online.

**Serial (RS-232)**

Serial data can be collected on the ‘Serial 2’ RS-232 port. Data is transmitted in a **binary** form outlined in [Appendix B](#). Sample messages, parsed code, and matlab parsing code are also available to help with integration here.

Xeos supplies a serial cable with the Brizo when no other telemetry packages are selected. The Pin-out for the cable is pictured below.

Wiring of the serial 2 connection can be confirmed by sending a `\$serial2test` command over either USB or Serial 1. The device will output the ASCII text “Serial 2 Test” over both the diagnostic port, and serial 2 port. Note: the device will not output wave data until it has a valid GNSS fix with greater than 8 satellites (initial start condition only), this means it must be outdoors to send wave data.
Upgrading Firmware

Connect to the Brizo with a USB or Serial cable and start STerm. Open a connection at 115200 baud.

Send the following commands to turn off wave height calculation:

```
$waveheightsetenabled f
```

On the top menu bar, navigate to Send File and click on Send Bootload Image.

The Firmware Image Send window will open and you will need to select the new version of firmware by clicking the ... and opening the .bin file. Once the Sent number matches the Size number, the upload is complete.

```
Reset the unit by sending the following command and wait for the unit to restart:

```
$resetnow
```

Check the firmware version to confirm the installation was successful by entering the $ver command and comparing it to the number on the .bin file that was uploaded:
$ver

2018-02-28 16:20:34 - Ver: 1.1.4275  Boot: 1.1.4097

If the new version wasn’t properly uploaded, resend the image.

Delete the old firmware using the $imglst command to view all the firmware images on the Brizo.

<table>
<thead>
<tr>
<th>Date</th>
<th>Size</th>
<th>Type</th>
<th>Version</th>
<th>Pri/Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>2018-02-28</td>
<td>320512</td>
<td>x3000</td>
<td>1.01.04262</td>
<td>1/011</td>
</tr>
<tr>
<td>2018-02-28</td>
<td>321792</td>
<td>x3000</td>
<td>1.01.04275</td>
<td>1/015</td>
</tr>
</tbody>
</table>

In this example, firmware version 4275 was uploaded so 4262 must be deleted. Enter $imgdelete followed by the version number, omitting the zeroes and replacing periods with a space. For this example, the following command would be sent to delete the old firmware version 1.01.04262:

$imgdelete 1 1 4262

Be careful to not delete the current version of firmware by accident.

Finally, re-enable the wave height measurements by entering this command:

$waveheightsetenabled t
## Appendix A: Command Index

<table>
<thead>
<tr>
<th>Command</th>
<th>Variables</th>
<th>Description</th>
<th>Sample</th>
</tr>
</thead>
<tbody>
<tr>
<td>$resetnow</td>
<td>N/A</td>
<td>Resets the unit</td>
<td>$resetnow</td>
</tr>
<tr>
<td>$schedule</td>
<td>N/A</td>
<td>Shows schedule of all enabled peripherals</td>
<td>$schedule</td>
</tr>
<tr>
<td>$ver</td>
<td>N/A</td>
<td>Shows the firmware version loaded</td>
<td>$ver</td>
</tr>
<tr>
<td>$getserialnumber</td>
<td>N/A</td>
<td>Shows the device serial number</td>
<td>$getserialnumber</td>
</tr>
<tr>
<td>$waveheightsetenabled</td>
<td>T,F</td>
<td>T (true) enables the measurement, F (false) disables the measurement</td>
<td>$waveheightsetenabled t</td>
</tr>
<tr>
<td>$waveheightgetenabled</td>
<td>N/A</td>
<td>Returns the enabled state of the measurement</td>
<td>$waveheightgetenabled</td>
</tr>
<tr>
<td>$waveheightsetsamplerate</td>
<td>[1-3]</td>
<td>Set the number of measurement periods per hour</td>
<td>$waveheightsetsamplerate 1</td>
</tr>
<tr>
<td>$waveheightgetsamplerate</td>
<td>N/A</td>
<td>Gets the number of measurement periods per hour</td>
<td>$waveheightgetsamplerate</td>
</tr>
<tr>
<td>$tunnelsetenabled</td>
<td>T,F</td>
<td>T (true) enables the tunnel, F (false) disables the tunnel</td>
<td>$tunnelsetenabled t</td>
</tr>
<tr>
<td>$tunnelgetenabled</td>
<td>N/A</td>
<td>Returns the enabled state of the tunnel</td>
<td>$tunnelgetenabled</td>
</tr>
<tr>
<td>$tunnelsetmodem</td>
<td>[0-3],[2-3]</td>
<td>Sets the priority of multiple cell and iridium modems</td>
<td>$tunnelsetmodem 0 3</td>
</tr>
<tr>
<td>$tunnelsetduration</td>
<td>s,m,h,d</td>
<td>Set the duration of the tunnel connection</td>
<td>$tunnelsetduration 600s</td>
</tr>
<tr>
<td>$tunnelgetduration</td>
<td>N/A</td>
<td>Returns the duration of the tunnel connection</td>
<td>$tunnelgetduration</td>
</tr>
<tr>
<td>$tunnelsetinterval</td>
<td>s,m,h,d</td>
<td>Set the interval of the tunnel</td>
<td>$tunnelsetinterval 1h</td>
</tr>
<tr>
<td>$tunnelgetinterval</td>
<td>N/A</td>
<td>Returns interval of the tunnel</td>
<td>$tunnelgetinterval</td>
</tr>
<tr>
<td>$tunnelsetoffset</td>
<td>s,m,h,d</td>
<td>Set interval of the tunnel</td>
<td>$tunnelsetoffset 20m</td>
</tr>
<tr>
<td>$tunnelgetoffset</td>
<td>N/A</td>
<td>Returns the offset of the tunnel</td>
<td>$tunnelgetoffset</td>
</tr>
<tr>
<td>Command</td>
<td>Type</td>
<td>Description</td>
<td>Function</td>
</tr>
<tr>
<td>---------</td>
<td>------</td>
<td>-------------</td>
<td>----------</td>
</tr>
<tr>
<td>$sirddsetdod</td>
<td>T,F</td>
<td>T (true) enables DISA gateway, F (false) disables DISA gateway</td>
<td>$sirddsetdod t</td>
</tr>
<tr>
<td>$sirddgetdod</td>
<td>N/A</td>
<td>Returns enabled state of DISA gateway</td>
<td>$sirddgetdod</td>
</tr>
<tr>
<td>$sbdsetenabled</td>
<td>T,F</td>
<td>T (true) enables SBD, F (false) disables SBD</td>
<td>$sbdsetenabled t</td>
</tr>
<tr>
<td>$sbdgetenabled</td>
<td>N/A</td>
<td>Returns the enabled state of SBD message checks</td>
<td>$sbdgetenabled</td>
</tr>
<tr>
<td>$sbdsetinterval</td>
<td>s,m,h,d</td>
<td>Set the interval of SBD message checks</td>
<td>$sbdsetinterval 4h</td>
</tr>
<tr>
<td>$sbdgetinterval</td>
<td>N/A</td>
<td>Returns the interval of the SBD message checks</td>
<td>$sbdgetinterval</td>
</tr>
<tr>
<td>$sbdsetoffset</td>
<td>s,m,h,d</td>
<td>Set the interval of the SBD message checks</td>
<td>$sbdsetoffset 0</td>
</tr>
<tr>
<td>$sbdgetoffset</td>
<td>N/A</td>
<td>Returns the offset of the SBD message checks</td>
<td>$sbdgetoffset</td>
</tr>
<tr>
<td>$serial2test</td>
<td>N/A</td>
<td>Outputs “Serial 2 Test” in ASCII format over diagnostic and Serial 2 ports</td>
<td>$serial2test</td>
</tr>
</tbody>
</table>
Appendix B: Serial Message Formats

"First 5" Coefficients Message

<table>
<thead>
<tr>
<th>Data Type</th>
<th>Name</th>
<th>Value /Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>uint8_t</td>
<td>Start Byte</td>
<td>0x6f</td>
</tr>
<tr>
<td>uint8_t</td>
<td>Version</td>
<td>0x02</td>
</tr>
<tr>
<td>uint8_t</td>
<td>Battery Voltage</td>
<td>0.1 V</td>
</tr>
<tr>
<td>int8_t</td>
<td>Temperature</td>
<td>deg C</td>
</tr>
<tr>
<td>uint16_t</td>
<td>Significant Wave Height</td>
<td>meters</td>
</tr>
<tr>
<td>uint16_t</td>
<td>Maximum Wave Height</td>
<td>Meters</td>
</tr>
<tr>
<td>uint16_t</td>
<td>Peak Wave Period</td>
<td>Seconds</td>
</tr>
<tr>
<td>uint16_t</td>
<td>Peak Wave Direction</td>
<td>Degrees</td>
</tr>
<tr>
<td>uint16_t</td>
<td>Peak Wave Spread</td>
<td>Degrees</td>
</tr>
<tr>
<td>uint8_t</td>
<td>Quality</td>
<td>Integer</td>
</tr>
<tr>
<td>uint32_t</td>
<td>Unix Time</td>
<td>Unix Minutes</td>
</tr>
</tbody>
</table>

Appendix C: RxTools Download

Download the RxTools software and manual from https://manuals.alertgeo.com/RxTools/
RX Tools contains the SBF Converter to convert SBF blocks to ASCII or RINEX. SBF Block info can be found in the SBF Reference Guide also in the link above.
Warranty, Support and Limited Liability

Xeos Technologies Inc. warranties the Brizo to be free of defects in material or manufacturing for a period of one year following delivery. Liability is limited to repair or replacement of the defective part and will be done free of charge.

LIMITED WARRANTY: Xeos Technologies Inc. warrants that the product will perform substantially in accordance with the accompanying written materials for a period of one year from the date of receipt.

CUSTOMER REMEDIES: Xeos Technologies Inc. entire liability and your exclusive remedy shall be at Xeos Technologies Inc. option, either (a) return of the price paid or (b) repair or replacement of the product that does not meet Xeos Technologies Inc. Limited Warranty and that is returned to Xeos Technologies Inc. with a copy of your receipt. This Limited Warranty is void if failure of the product has resulted from accident, abuse, or misapplication. Any replacement product will be accepted for the remainder of the original warranty period or ninety (90) days, whichever is longer.

NO OTHER WARRANTIES: Xeos Technologies Inc. disclaims all other warranties, either expressed or implied, including but not limited to implied warranties of merchantability and fitness for a purpose, with respect to the product or the accompanying written materials. This limited warranty gives you specific legal rights. You may have others, which vary from state to state.

NO LIABILITY FOR CONSEQUENTIAL DAMAGES: In no event shall Xeos Technologies Inc. or its suppliers be liable for any damages whatsoever (including, without limitation, damages for loss of equipment, for loss of business profits, business interruption, loss of business information, or other pecuniary loss) arising out of the use of or inability to use this Xeos Technologies Inc. product, even if Xeos Technologies Inc. has been advised of the possibility of such damages.