Product Manual

# SBE 25plus Sealogger CTD

Conductivity, Temperature, and Pressure Recorder
with RS-232 Interface

| Release Date | 05/24/17 |
|---|---|
| Version | 006 |
| Firmware | 1.0 & later |
| Software | Seaterm V2 2.6.1 & later |
| | Seasave V7 7.26.0 & later |
| | SBE Data Processing 7.26.0 & later |
| | UCI 1.2.1 & later |

CE

service@seabird.com | support@seabird.com | Tel: +1 425 643 9866

# Limited Liability Statement

Extreme care should be exercised when using or servicing this equipment. It should be used or serviced only by personnel with knowledge of and training in the use and maintenance of oceanographic electronic equipment.

SEA-BIRD SCIENTIFIC disclaims all product liability risks arising from the use or servicing of this system. SEA-BIRD SCIENTIFIC has no way of controlling the use of this equipment or of choosing the personnel to operate it, and therefore cannot take steps to comply with laws pertaining to product liability, including laws which impose a duty to warn the user of any dangers involved in operating this equipment. Therefore, acceptance of this system by the customer shall be conclusively deemed to include a covenant by the customer to defend, indemnify, and hold SEA-BIRD SCIENTIFIC harmless from all product liability claims arising from the use or servicing of this system.

# Declaration of Conformity

**Sea-Bird Electronics, Inc.**
13431 NE 20th Street Bellevue, WA 98005

## DECLARATION OF CONFORMITY

**Manufacturer's Name:**    Sea-Bird Electronics

**Manufacturer's Address:**    13431 NE 20th Street
Bellevue, WA 98005, USA

**Device Description:**    Various Data Acquisition Devices and Sensors

**Model Numbers:**

| 3S | 3F | 3plus | 4C | 4M | 5M | 5T |
|---|---|---|---|---|---|---|
| 5P | 7 | 8 | 9plus | 11plusV2 | 14 | 16plusV2 |
| 16plusIM-V2 | 17plusV2 | 18 | 19plusV2 | 21 | 25plus | 26plus |
| 27 | 29 | 32 | 32C | 32SC | 33 | 35 |
| 35RT | 36/PDIM | 37-SI | 37-SIP | 37-IM | 37-IMP | 37-SM |
| 37-SMP | 38 | 39 | 39-IM | 39plus | 41 | 41CP |
| 43 | 43F | 44 | 45 | 48 | 49 | 50 |
| 52-MP | 53 | 54 | 55 | 56 | 63 | PN 90204 |
| PN 90158.1 | PN 90488 | PN 90545 | PN 90402 | AFM | SIM | ICC |
| IMM/UIMM | Glider Payload CTD | NiMH Battery Charger & Battery Pack (PN 90504) | | | | |

**Applicable EU Directives:**    Machinery Directive 2006/42/EC
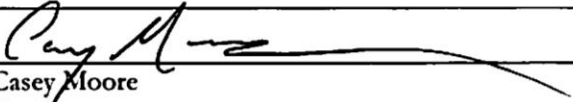
EMC Directive 2004/108/EC

Low Voltage Directive (2006/95/EC)

**Applicable Harmonized Standards:**

EN 61326-1:2013 Class A Electrical Equipment for Measurement, Control and Laboratory Use, EMC Requirement - Part 1: General Requirements (EN 55011:2009 Group 1, Class A)

EN 61010-1:2010, Safety Requirements for Electrical Equipments for Measurement, Control, and Laboratory Use - Part 1: General Requirements

I, the undersigned, hereby declare that the equipment specified above conforms to the above European Union Directives and Standards.

| Authorized Signature: | |
|---|---|
| Name: | Casey Moore |
| Title of Signatory: | President |
| Date: | June 6, 2014 |
| Place: | Belleview, WA |

# Table of Contents

# Section 1: Introduction

This section includes contact information, Quick Start procedure, and photos of a typical SBE 25*plus* shipment.

## About this Manual

This manual is to be used with the SBE 25*plus* Sealogger Conductivity, Temperature, and Pressure Recorder.

It is organized to guide the user from installation through operation and data collection. We have included detailed specifications, command descriptions, maintenance and calibration information, and helpful notes throughout the manual.

Sea-Bird welcomes suggestions for new features and enhancements of our products and/or documentation. Please contact us with any comments or suggestions (seabird@seabird.com or 425-643-9866). Our business hours are Monday through Friday, 0800 to 1700 Pacific Standard Time (1600 to 0100 Universal Time) in winter and 0800 to 1700 Pacific Daylight Time (1500 to 0000 Universal Time) the rest of the year.

## Quick Start

Follow these steps to get a Quick Start using the SBE 25*plus*.
The manual provides step-by-step details for performing each task:

1. Install batteries and test power and communications (see *Section 3: Power and Communications Test*).
2. Deploy the 25*plus* (see *Section 4: Deploying and Operating SBE 25plus*):
   A. Install new batteries if necessary.
   B. Ensure all data has been uploaded, and then send **DeleteAll** to make entire memory available for recording if desired.
   C. Set date and time and establish setup and logging parameters. Send **QS** to put 25*plus* in quiescent state.
   D. Install dummy plugs and/or cable connectors, and locking sleeves.
   E. Remove Tygon tubing that was looped end-to-end around conductivity cell for storage. Reconnect cell to 25*plus* plumbing.
   F. Push plunger switch in to On position.
   G. Deploy 25*plus*.

## Unpacking SBE 25*plus*

Shown below is a typical SBE 25*plus* shipment.



SBE 25*plus* Sealogger



Data I/O cable
(connects computer to Y-cable
from J7 connector on main housing
end cap)



Conductivity cell
cleaning solution
(Triton-X)



Conductivity cell filling
and storage kit



Spares - small hardware
and o-ring kit



Software, and Electronic Copies of
Software Manuals and User Manual

# Section 2: Description of SBE 25*plus*

This section describes the functions and features of the SBE 25*plus*, including specifications, dimensions, end cap connectors, plunger switch, data I/O, memory endurance, batteries and battery endurance, external power, and plumbing.

## System Description
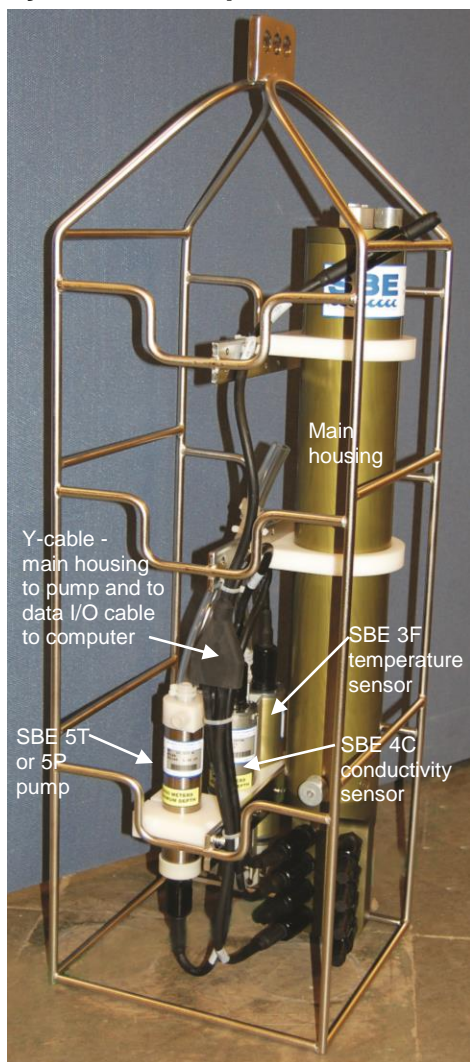


The SBE 25*plus* Sealogger is the ideal research-quality CTD profiling system for coastal, estuarine, and budget-minded deep-water deployments. The battery-powered 25*plus* records data in memory, eliminating the need for a large vessel, electro-mechanical sea cable, and on-board computer. The 25*plus* can also operate in real-time, transmitting data via an RS-232 serial port or Sea-Bird deck unit.

Compared to the previous SBE 25, the 25*plus* incorporates an electronics upgrade, mechanical redesign, and additional features, with 16 Hz sampling, 8 differentially amplified A/D input channels, 16-bit A/D resolution for auxiliary sensor channels, more power for auxiliary sensors such as nitrate and $CO_2$ sensors, 2 RS-232 data input channels, and 2 GB memory. Data in memory is uploaded via the external bulkhead connector or the internal USB connector (for fast upload of large data sets). The unique end cap design provides easy access to bulkhead connectors, simplifying the addition and removal of sensors from the package.

## Sensors

Temperature and conductivity are measured by the modular Sea-Bird SBE 3F and SBE 4C sensors. The SBE 5 Pump and TC Duct provide pump-controlled, TC-ducted flow, to ensure coordinated measurement of the same parcel of water. This significantly reduces salinity spiking caused by ship heave, and in calm waters allows slower descent for improved resolution of water column features. The integrated, temperature-compensated strain-gauge pressure sensor is available in eight depth ranges to suit the operating depth requirement. The 25*plus* includes interface electronics for up to 8 voltage-output sensors (dissolved oxygen, pH, fluorescence, PAR, light transmission, optical backscatter, etc.) and 2 RS-232 output sensors.

## Operation

The 16 Hz scan rate provides very high spatial resolution of oceanographic structures and gradients, and recording endurance of up to 55 hours (without auxiliary sensors) with alkaline batteries. Recorded data are transferred via RS-232 or USB to a PC for processing.

Simultaneous with recording, real-time data can be transmitted directly to a PC serial port **(serial output sensor data cannot be transmitted in real-time)**. Real-time data output rate varies from 0.5 to 8 Hz, depending on the baud rate and the number of auxiliary voltage sensors enabled for real-time output. Transmission distance is dependent on number of auxiliary voltage sensors enabled for real-time output, baud rate, and cable properties. External power and two-way real-time communication over 10,000 meters of single-core, armored cable can be provided with the SBE 36 CTD Deck Unit and Power/Data Interface Module (PDIM), or with the SBE 33 Deck Unit and SBE 32 or 55 Water Sampler.

The SBE 25*plus* is easily integrated with an SBE 32 Carousel Water Sampler or SBE 55 ECO Water Sampler. Both real-time and autonomous *auto-fire* operations are possible with any Sea-Bird CTD / Water Sampler system.

Future upgrades and enhancements to the SBE 25*plus* firmware can be easily installed in the field through a computer serial port and the *Data I/O, Pump, and External Power* bulkhead connector on the 25*plus*, without the need to return the 25*plus* to Sea-Bird.

## Configuration, Options, and Accessories

An SBE 25*plus* is supplied with:

- Plastic main housing for depths to 600 meters (1950 feet) or aluminum housing for depths to 6800 meters (22,300 feet)
- Modular temperature (SBE 3F) and conductivity (SBE 4C) sensors
- Integrated strain-gauge pressure sensor (choice of 8 ranges), in main housing
- T-C Duct, which ensures that Temperature and Conductivity measurements are made on the same parcel of water
- SBE 5P plastic (depths to 600 meters) or SBE 5T titanium (depths to 10,500 meters) submersible pump for pumped conductivity; by fixing the flow to a constant rate, the pump ensures a constant conductivity time response. *The T-C duct and pump combination results in dramatically lower salinity spiking*.
- Eight 16-bit differentially amplified A/D input channels for optional auxiliary sensors
- Two RS-232 input channels for optional auxiliary sensors (serial output sensor data cannot be transmitted in real-time)
- 2 GByte memory
- 12 D-size alkaline batteries (Duracell MN1300, LR20) in a battery pack
- Bulkhead connectors (XSG / AG or wet-pluggable MCBH)
- Protective stainless-steel cage

SBE 25*plus* options include:

- Auxiliary sensors for dissolved oxygen, pH, fluorescence, radiance (PAR), light transmission, optical backscatter (turbidity), etc.
- Plastic shipping case instead of wood crate
- Load-bearing underwater cables for hand-hauled, real-time profiling
- SBE 36 CTD Deck Unit and PDIM or SBE 33 Deck Unit and Sea-Bird water sampler for real-time operation on single-core armored cable (cable length to 10,000 m)

## Integration with Other Sea-Bird Systems

**Notes:**
- The SBE 32 Carousel is a 12-, 24-, or 36-bottle water sampler.
  The SBE 55 ECO is a 3- or 6-bottle water sampler.
- Seasave also supports acquisition of data from a NMEA device connected directly to the computer (instead of the deck unit).
- SBE 33's NMEA interface was optional prior to 2010.
- SBE 33's Surface PAR interface was optional prior to 2012.

The SBE 25*plus* can be used with the following Sea-Bird equipment:

- **SBE 32 Carousel Water Sampler and SBE 33 Carousel Deck Unit OR SBE 55 ECO Water Sampler and SBE 33 Carousel Deck Unit** -
  The SBE 32 or SBE 55 provides +15 VDC power to the 25*plus* and has ample power available for auxiliary sensors not normally supported by battery-powered CTDs. CTD data from the 25*plus* is converted into single-wire telemetry for transmission over long (up to 10,000 meter [32,800 feet]) sea cables. Bottles may be closed at any depth without interrupting CTD data via software control using the Seasave program or from the front panel of the SBE 33 Deck Unit. See the SBE 33 manual for operating details.
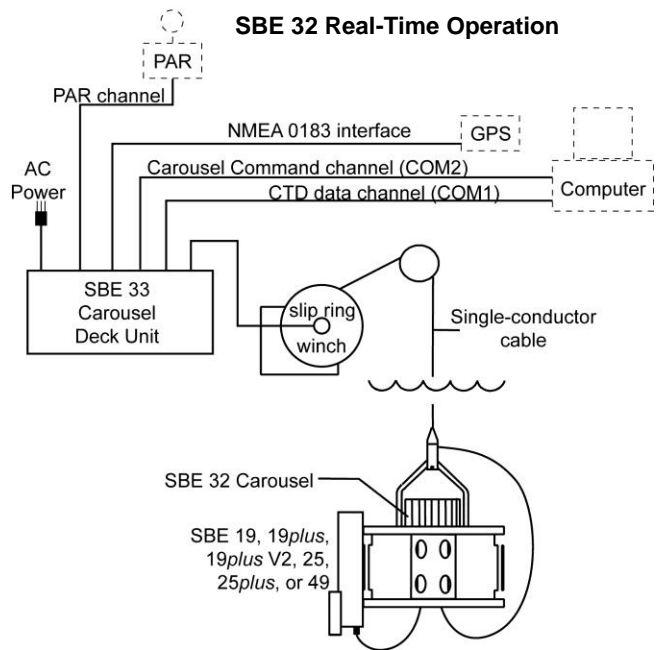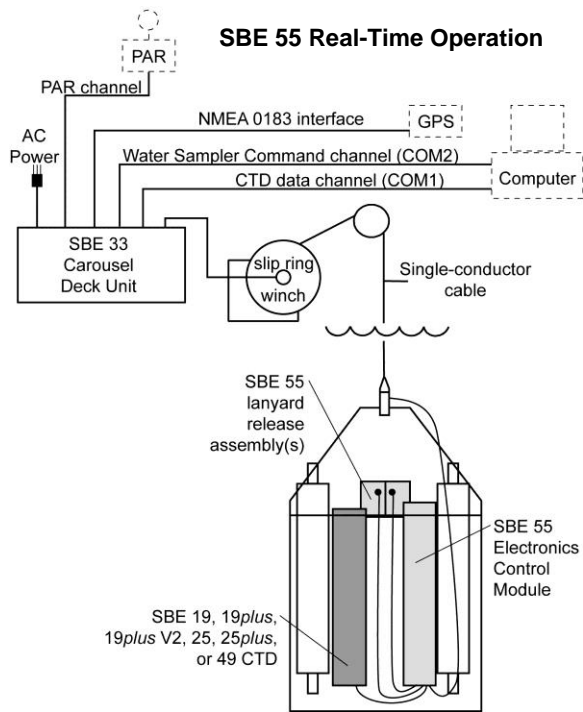
- **SBE 36 CTD Deck Unit and Power Data Interface Module (PDIM)** - These items provide power and real-time data handling capabilities over single-conductor sea cables using the same method as employed in the SBE 32/SBE 33. The PDIM is a small pressure housing that is mounted on or near the 25*plus*. It provides +15 VDC power to the 25*plus* and interfaces two-way RS-232 communications from the 25*plus* to the telemetry used on the sea cable. See the SBE 36/PDIM manual for operating details.

**Notes:**
- Seasave also supports acquisition of data from a NMEA device connected directly to the computer (instead of the deck unit).
- SBE 36's Surface PAR interface was optional prior to 2012.
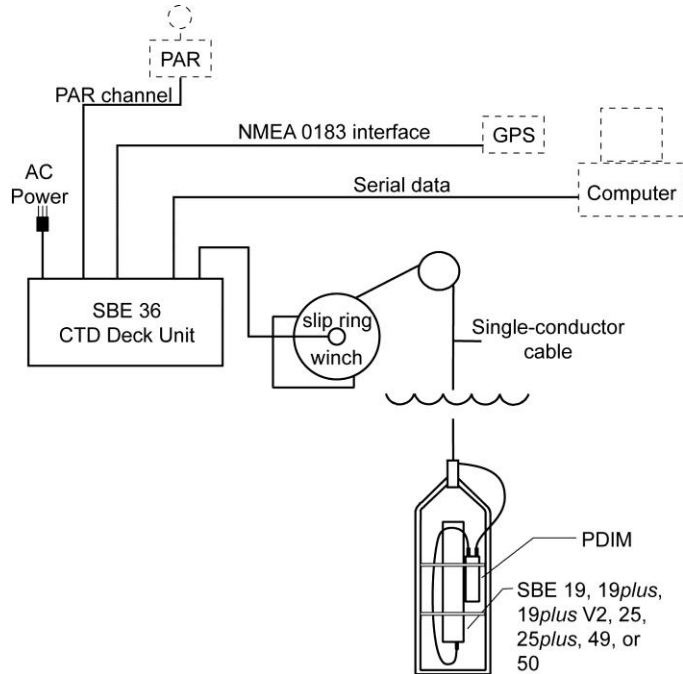


- **SBE 32 Carousel Water Sampler and Auto Fire Module (AFM)** - The AFM, mounted on or near the 25*plus*, allows the SBE 32 to operate autonomously on non-conducting cables. The AFM supplies the operating voltage, logic, and control commands necessary to operate the SBE 32. The AFM monitors the pressure data recorded by the 25*plus* in real-time, closing water sampler bottles at predefined pressures (depths) or whenever the system is stationary for a specified period of time. Bottle number, firing confirmation, and five scans of CTD data are recorded in the AFM memory for each bottle fired. See the AFM manual for operating details.

**SBE 32 Autonomous Operation**



11
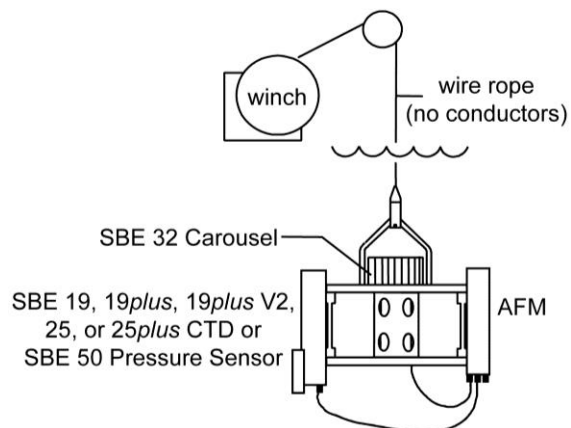
> **Note:**
> The SBE 32 Carousel is a 12-, 24-, or 36-bottle water sampler. The SBE 55 ECO is a 3- or 6-bottle water sampler.

- **SBE 55 ECO Water Sampler** - The SBE 55 has built-in capabilities similar to that of an SBE 32 Carousel Water Sampler integrated with an Auto Fire Module, allowing the SBE 55 to operate autonomously on non-conducting cables. The SBE 55 monitors the pressure data recorded by the 25*plus* in real-time, closing water sampler bottles at predefined pressures (depths) or whenever the system is stationary for a specified period of time. Bottle number, firing confirmation, and five scans of CTD data are recorded in the SBE 55 memory for each bottle fired. See the SBE 55 manual for operating details.



**SBE 55 Autonomous Operation**

## Software

> **Notes:**
> - Help files provide detailed information on the software.
> - Separate software manuals on CD-ROM contain detailed information on Seasave V7 and SBE Data Processing.
> - The older version of Seasave, Seasave-Win32, is not compatible with the 25*plus*.
> - Sea-Bird supplies the current version of our software when you purchase an instrument. As software revisions occur, we post the revised software on our website. See our website for the latest software version number, a description of the software changes, and instructions for downloading the software.

The SBE 25*plus* is supplied with a powerful Windows software package, Seasoft© V2, which includes:

- **SeatermV2** – terminal program for easy communication and data retrieval. SeatermV2 is a *launcher*. Depending on the instrument selected it launches **Seaterm232** (RS-232 instruments, such as the 25*plus*), Seaterm485 (RS-485 instruments), or SeatermIM (inductive modem instruments).

- **SeatermAF** – terminal program for easy communication and data retrieval when the SBE 25*plus* is used with a water sampler operated autonomously (SBE 32 Carousel Water Sampler with Auto Fire Module **or** SBE 55 ECO Water Sampler).

- **Seasave V7** – program for acquiring, converting, and displaying real-time or archived raw data.

- **SBE Data Processing** - program for calculation and plotting of conductivity, temperature, pressure, auxiliary sensor data, and derived variables such as salinity and sound velocity.

## Specifications

|  | Temperature (°C) | Conductivity (S/m) | Pressure |
|---|---|---|---|
| **Measurement Range** | -5 to +35 | 0 to 7 | 0 to full scale range (expressed in meters of deployment depth capability): 20 / 100 / 350 / 600 / 1000 / 2000 / 3500 / 7000 meters |
| **Initial Accuracy** | ±0.001 | ±0.0003 | ±0.1% of full scale range |
| **Resolution** | 0.0003 | 0.00004 | 0.002% of full scale range |
| **Sensor Calibration** (measurement outside these ranges may be at slightly reduced accuracy due to extrapolation errors) | -1.4 to +32.5 | 2.6 to 6 S/m, plus zero conductivity (air) | Ambient pressure to full scale range in 5 steps |
| **Sampling Speed** | 16 Hz (16 samples/sec) | | |
| **Memory** | 2 GByte<br>Holds approximately 1320 hours of data (T, C, P, and 8 auxiliary voltage sensors; does not include auxiliary serial sensor data) | | |
| **Data Storage** | **Recorded Parameter**                 **Bytes/sample**<br>temperature + conductivity + pressure      10<br>each external voltage                             2<br>each auxiliary RS-232 sensor      sensor dependent | | |
| **Real-Time Clock** | 32,768 Hz TCXO accurate to ±1 minute/year | | |
| **Internal Batteries (nominal capacity)** | **Main batteries:**<br>Twelve (12) alkaline D-cells (Duracell MN1300, LR20) in battery pack - 55 hours profiling with no auxiliary sensors.<br>**Back-up batteries:**<br>*Main back-up* – two (2) ½AA (LS14250) batteries; usable range 7.2 to 3.8V<br>*Real-time clock back-up* – one (1) coin cell, CR2330/1HF, usable range 3.6 to 2.8V | | |
| **External Power Supply** (if not relying on internal batteries) | 14 - 20 VDC; consult factory for required current | | |
| **Power Requirements** | Sampling (no auxiliary sensors or pump)             95 mA<br>Pump (SBE 5P or 5T)                              150 mA<br>Communications                                   70 mA<br>Quiescent (when powered by internal batteries)     70 μA<br>Quiescent (when powered externally)              175 μA | | |
| **Auxiliary Sensors** | Auxiliary power out: 12 VD, up to 1.2 A across all channels (1 A maximum per channel).<br>Voltage sensor: 0 - 5 VDC, 16 bits.<br>*Note: serial output sensor data cannot be transmitted in real-time.* | | |
| **Main Housing Material, System Weight** (including C & T sensors, pump, and cage) | **600 meter (1950 ft)** - acetal copolymer (plastic), 20.0 kg (44 lbs) in air, 11.5 kg (25 lbs) in water<br>**6800 meter (22,300 ft)** - 7075 aluminum, 22.5 kg (50 lbs) in air, 13.5 kg (30 lbs) in water | | |
| **Cage** | 6.7 kg (15 lbs) in air, 5.9 kg (13 lbs) in water | | |

**Notes:**
- See *Battery Endurance* for example calculations.
- Auxiliary sensors that draw more than 1 Amp even momentarily must be cabled to J3 (voltage output sensors) or J6 (serial output sensors). See *Appendix V: Integration of Auxiliary Sensors with High Current Draw* for jumper details.

## Dimensions



SBE 25*plus*
main housing

SBE 3F
Temperature Sensor
& SBE 4
Conductivity Sensor

SBE 5T
Pump

96.5 cm
(38 in.)

30.5 cm
(12 in.)

27.9 cm
(11 in.)
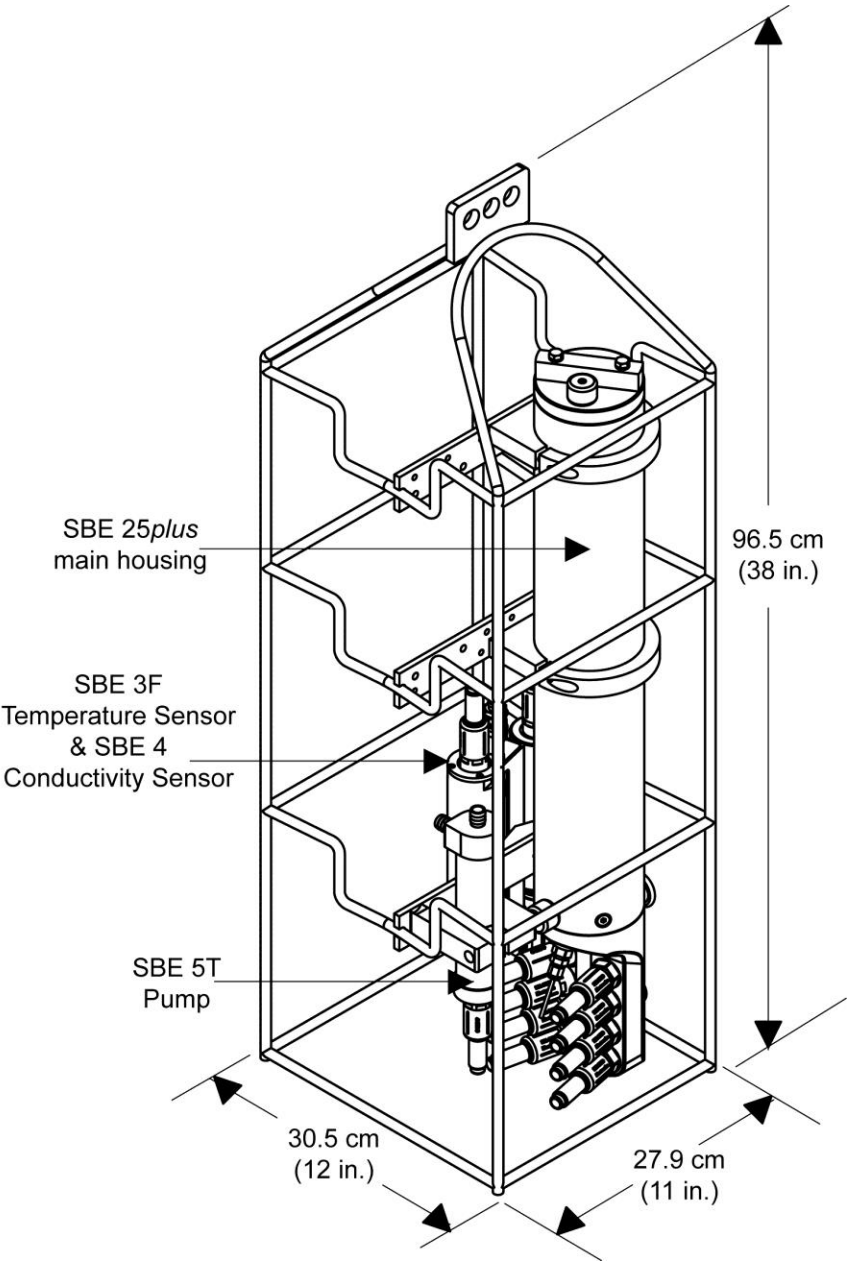
# Main Housing End Cap Connectors

**Notes:**
- A y-cable from J7 connects to the pump and to a data I/O-power cable (for connection to the computer for setup and uploading of data, and for externally powering 25*plus*).
- **Auxiliary sensors that draw more than 1 Amp even momentarily must be cabled to J3 (voltage output sensors) or J6 (serial output sensors). See** *Appendix V: Integration of Auxiliary Sensors with High Current Draw* **for jumper details.**



= XSG / AG connector

= MCBH connector (all with 3/8" length base, 1/2-20 thread)

**J4, J5**

XSG-3-BCL-HP

MCBH-3MP (WB), AL

**J0, J1, J2, J3, J6, J7**

AG-306-HP-SS

MCBH-6MP (WB), AL

SBE 25*plus* End Cap
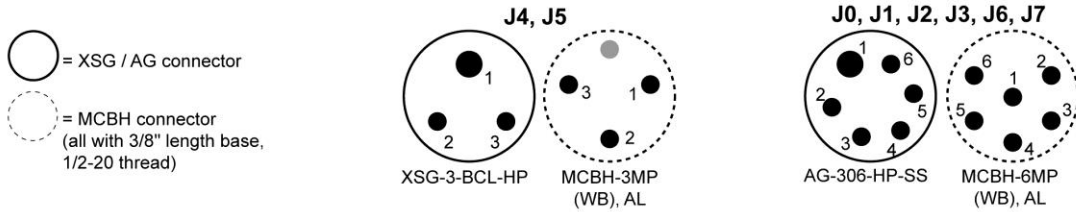
**J3 Auxiliary Differential Input 6,7**

| Pin | Signal | Pin | Signal |
|---|---|---|---|
| 1 | Common | 4 | Voltage 7 Signal |
| 2 | Voltage 6 Signal | 5 | Voltage 7 Common |
| 3 | Voltage 6 Common | 6 | Auxiliary Power Out |

**J2 Auxiliary Differential Input 4,5**

| Pin | Signal | Pin | Signal |
|---|---|---|---|
| 1 | Common | 4 | Voltage 5 Signal |
| 2 | Voltage 4 Signal | 5 | Voltage 5 Common |
| 3 | Voltage 4 Common | 6 | Auxiliary Power Out |

**J1 Auxiliary Differential Input 2,3**

| Pin | Signal | Pin | Signal |
|---|---|---|---|
| 1 | Common | 4 | Voltage 3 Signal |
| 2 | Voltage 2 Signal | 5 | Voltage 3 Common |
| 3 | Voltage 2 Common | 6 | Auxiliary Power Out |

**J0 Auxiliary Differential Input 0,1**

| Pin | Signal | Pin | Signal |
|---|---|---|---|
| 1 | Common | 4 | Voltage 1 Signal |
| 2 | Voltage 0 Signal | 5 | Voltage 1 Common |
| 3 | Voltage 0 Common | 6 | Auxiliary Power Out |
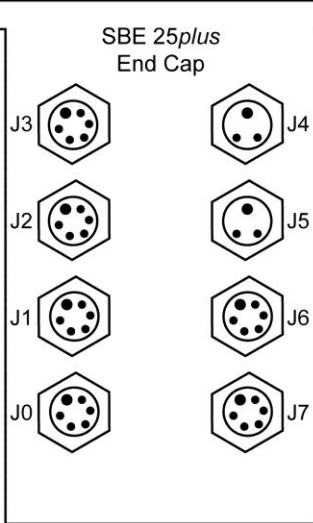
**J4 Conductivity**

| Pin | Signal |
|---|---|
| 1 | Common |
| 2 | Conductivity frequency |
| 3 | Auxiliary Power Out |

**J5 Temperature**

| Pin | Signal |
|---|---|
| 1 | Common |
| 2 | Temperature frequency |
| 3 | Auxiliary Power Out |

**J6 Auxiliary Serial Input 1,2**

| Pin | Signal | Pin | Signal |
|---|---|---|---|
| 1 | Common | 4 | Serial 1 data receive |
| 2 | Serial 2 data receive | 5 | Serial 1 data transmit |
| 3 | Serial 2 data transmit | 6 | Auxiliary power out |

**J7 Data I/O, Pump, and External Power**

| Pin | Signal | Pin | Signal |
|---|---|---|---|
| 1 | Common | 4 | Pump power common |
| 2 | RS-232 data receive | 5 | Pump power |
| 3 | RS-232 data transmit | 6 | Auxiliary power in (14-20 VDC) |

# Cables



**Y-cable** –
6-pin connects to J7 on 25*plus* bulkhead connector; 2-pin connects to pump; 4-pin connects to data I/O cable
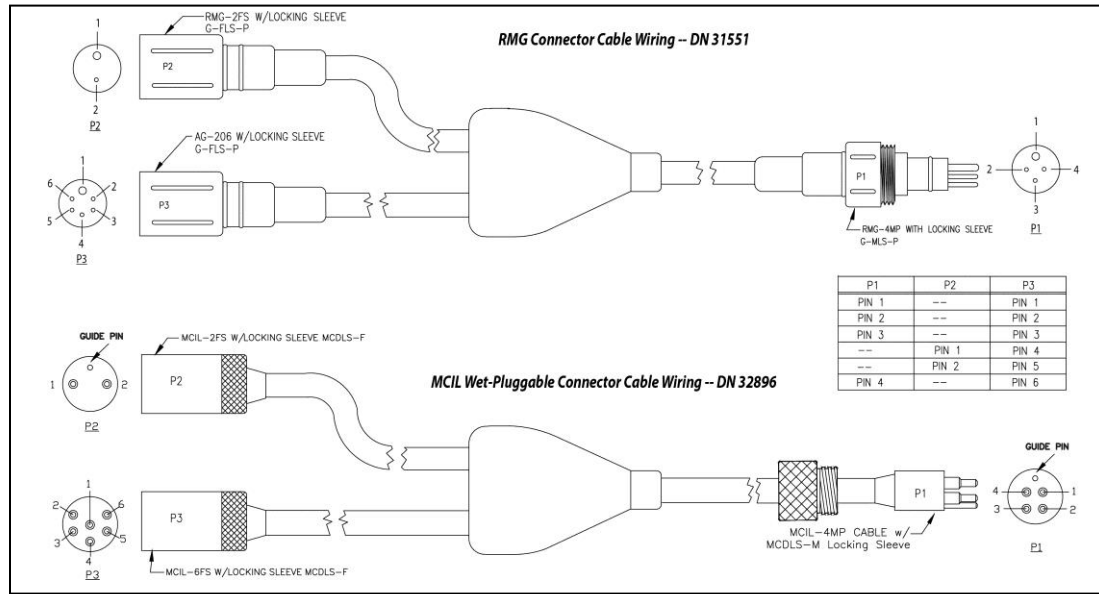
**Data I/O cable** – connects to 4-pin end of Y-cable
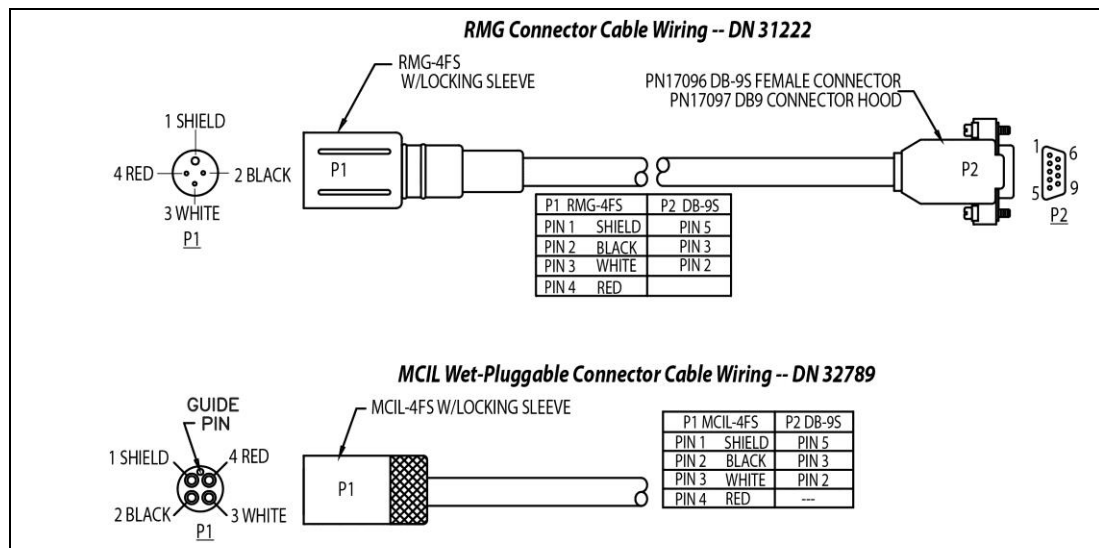
**Sensor cable –** connects SBE 3 to J5 bulkhead connector or SBE 4 to J4 bulkhead connector on 25*plus*

## Plunger Switch



Plunger      Connector
switch       end cap
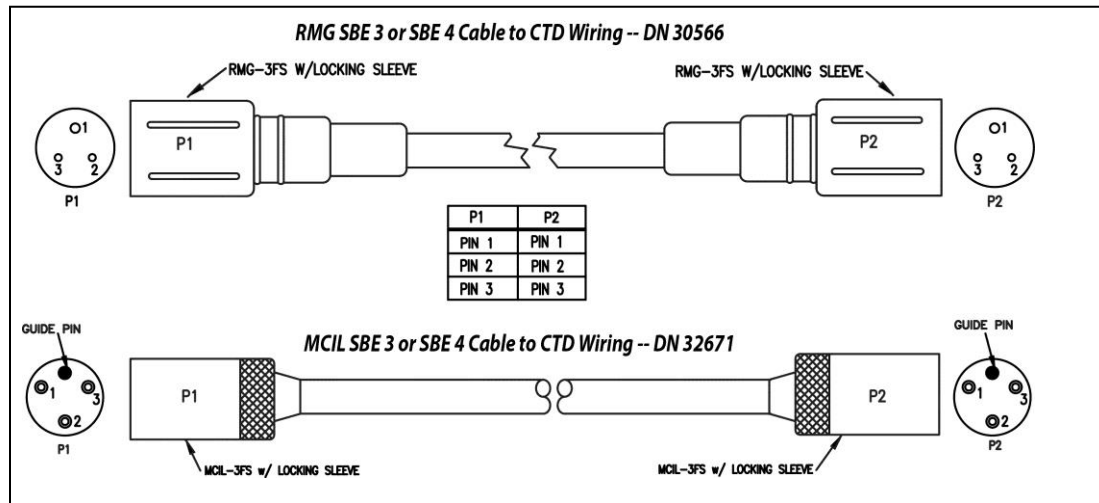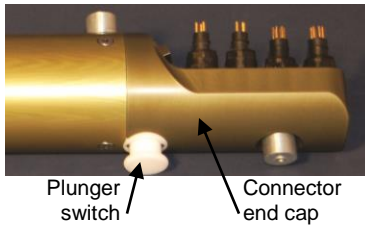
A plunger with a magnet, mounted on the end cap on the main housing, is used to start and stop logging. Pushing the switch in wakes up the SBE 25*plus* and starts logging. Pulling the switch out stops logging; if no commands are then sent, the 25*plus* enters quiescent (sleep) state after two minutes.

**The plunger must be pulled out when the SBE 25*plus* is not logging data; i.e., during setup, diagnostics, and data upload.**

**Note:**
See *Section 4: Deploying and Operating SBE 25plus* for details on commands and alternate methods for starting and stopping logging.

## Data I/O

The SBE 25*plus* receives setup instructions and outputs diagnostic information or previously recorded data via a three-wire RS-232C link, and is factory-configured for 9600 baud, 8 data bits, 1 stop bit, and no parity. SBE 25*plus* RS-232 levels are directly compatible with standard serial interface cards (IBM Asynchronous Communications Adapter or equal). The communications baud rate can be changed using **SetBaudConsole=** (see *Section 4: Deploying and Operating SBE 25plus*).

## Data Storage

**Note:**
See *Specifications* for data storage required for each parameter.

The SBE 25*plus* has a 2 GB memory. Shown below are calculations of available data storage for several configurations.

*Example 1:* 2 GB memory, no auxiliary sensors
T, C, and P = 10 bytes/sample
Storage space $\approx$ 2,000,000,000 / 10 $\approx$ 200,000,000 samples

*Example 2:* 2 GB memory, 4 auxiliary voltage sensors
T, C, and P = 10 bytes/sample
External voltages = 2 bytes/sample x 4 sensors = 8 bytes/sample
Storage space $\approx$ 2,000,000,000 / (10 + 8) $\approx$ 111,111,111 samples

# Batteries



> **Note:**
> See *Replacing Batteries* in *Section 5: Routine Maintenance and Calibration*.

For the main battery, the SBE 25*plus* uses twelve D-cell alkaline batteries (Duracell MN 1300, LR20) in a battery pack.

On-board lithium batteries (non-hazardous units that are unrestricted for shipping purposes) are provided to back-up the buffer and the real-time clock in the event of main battery failure or exhaustion. The main batteries may be replaced without affecting either the real-time clock or memory.

# Battery Endurance

> **Notes:**
> - See *Specifications* for power requirements.
> - See *Data Storage* and *Specifications* for data storage limitations.
> - If the SBE 25*plus* is logging data and the internal battery voltage is less than 9.5 volts for 20 seconds **or** the external power is less than 14 volts for 20 seconds, it halts logging and displays <Powerfail/>.
> - Auxiliary sensors that draw more than 1 Amp even momentarily must be cabled to J3 (voltage output sensors) or J6 (serial output sensors). See *Appendix V: Integration of Auxiliary Sensors with High Current Draw* for jumper details.

The 12-cell alkaline battery pack has a nominal capacity of 18.6 Amp-hours. Current consumption varies, depending on inclusion of auxiliary sensors. For planning purposes, Sea-Bird recommends using a conservative battery capacity value of:

- 14 Amp-hours for a 25*plus* with current draw of less than 500 mA from auxiliary sensors.
- 5 Amp-hours for a 25*plus* with current draw of more than 500 mA from auxiliary sensors (the faster you take the current out, the lower the capacity of the batteries).

Examples are shown below for three sampling schemes.

---

*Example 1:* no auxiliary sensors
Sampling current (T, C, and P) = 95 mA = 0.095 A
Pump current = 150 mA = 0.15 A
Maximum sampling time ≈ 14 Amp-hours / (0.095 A + 0.15 A) ≈ 55 hours

*Example 2:* auxiliary sensors drawing 100 mA
Sampling current (T, C, and P) = 95 mA = 0.095 A
Pump current = 150 mA = 0.15 A
Auxiliary sensor current = 100 mA = 0.10 A
Maximum sampling time ≈ 14 Amp-hours / (0.095 A + 0.15 A + 0.10 A)
                 ≈ 40 hours

*Example 3:* auxiliary sensors drawing 1000 mA (high current draw; use 5Amp-hour capacity for battery pack)
Sampling current (T, C, and P) = 95 mA = 0.095 A
Pump current = 150 mA = 0.15 A
Auxiliary sensor current = 1000 mA = 1.0 A
Maximum sampling time ≈ 5 Amp-hours / (0.095 A + 0.15 A + 1.0 A)
                 ≈ 4 hours

---

# External Power

The SBE 25*plus* can be powered from an external source (14 - 20 volts DC) through the Y-cable connected to the Data I/O, Pump, and External Power bulkhead connector (J7). If external power of 14 volts or higher is applied, the 25*plus* runs off of the external power, even if the main battery voltage is higher. The 25*plus* can also be operated from the external supply without having the internal battery pack installed. Electrical isolation of conductivity is retained in units powered externally, preventing ground loop noise contamination in the conductivity measurement.

On the ship, cables longer than 3 meters should be installed inside an earthed metal conduit by a qualified electrician. This minimizes the potential for external signals to disrupt communication and ensures that high voltage lines (such as the sea cable) are sufficiently protected. Cables shorter than 3 meters can be used without shielding when installing or bench testing the instrument.

## Cable Length and External Power without Deck Unit

**Note:**
See *Real-Time Setup* in *Section 4: Deploying and Operating SBE 25plus* for baud rate limitations on cable length if transmitting real-time data.

There are two issues to consider if powering the SBE 25*plus* externally:

- Limiting the communication IR loss to 1 volt **if transmitting real-time data**; higher IR loss will cause the instrument to transmit data that does not meet the RS-232 communication standard.
- Supplying enough power at the power source so that sufficient power is available at the instrument after considering IR loss.

Each issue is discussed below.

*Limiting Communication IR Loss to 1 Volt if Transmitting Real-Time Data*

**Note:**
Common wire resistances:

| Gauge | Resistance (ohms/foot) |
|-------|------------------------|
| 12 | 0.0016 |
| 14 | 0.0025 |
| 16 | 0.0040 |
| 18 | 0.0064 |
| 19 | 0.0081 |
| 20 | 0.0107 |
| 22 | 0.0162 |
| 24 | 0.0257 |
| 26 | 0.0410 |
| 28 | 0.0653 |

The limit to cable length is typically reached when the maximum current during *communication* times the power common wire resistance is more than 1 volt.

$$V_{limit} = 1 \text{ volt} = IR_{limit}$$

Maximum cable length = $R_{limit}$ / wire resistance per foot

*where* I = current required by 25*plus* during communication. The 25*plus* samples and transmits data continuously; use the total current required for sampling plus communication (sampling, pump, auxiliary sensor current, and communication current) in the calculation.

---

*Example - for 25plus with no auxiliary sensors*
For 20 gauge wire, what is maximum distance to transmit power to 25*plus* if transmitting real-time data?
Current = 95 mA (sampling) + 150 mA (pump) + 70 mA (communication) = 315 mA
$R_{limit} = V_{limit}$ / I = 1 volt / 0.315 amps = 3.2 ohms
For 20 gauge wire, resistance is 0.0107 ohms/foot.
Maximum cable length = 3.2 ohms / 0.0107 ohms/foot = 299 feet = 91 meters

---

### *Supplying Enough Power to SBE 25plus*

Another consideration in determining maximum cable length is supplying enough power at the power source so that sufficient voltage is available, after IR loss in the cable (***from the turn-on transient, two-way resistance***), to power the 25*plus*. The table summarizes the maximum 2-way resistance for various input supplies:

| Power Supply Input and Pump Configuration | $R_{limit}$ = Maximum 2-way Resistance (ohms) |
|---|---|
| 3.0 amps at 14V input, SBE 5T or 5P pump | 2 |
| 1.5 amps at 19V input, SBE 5T or 5P pump | 7 |

Calculate maximum cable length as:

Maximum cable length = $R_{limit}$ / 2 * wire resistance per foot

---

*Example* – For 20 gauge wire, what is maximum distance to transmit power to 25*plus* if using 12 volt power source?
Maximum cable length = $R_{limit}$ / 2 * wire resistance per foot = 2 ohms / 2 * 0.0107 ohms/foot = 93 ft = 28 meters
Note that 28 meters < 173 meters and 474 meters (maximum distance if transmitting real-time data in examples above), so IR drop in power is controlling factor for this example. Using a higher voltage power supply or a different wire gauge would increase allowable cable length.

---

# Plumbing

An SBE 25*plus* is typically deployed in a vertical position. However, when used with an SBE 32 Carousel Water Sampler, the 25*plus* may be deployed in a horizontal position. Pump placement and plumbing for a horizontal mount is different than that for a vertical mount.

Sea-Bird provides the system with pump placement and plumbing for the desired orientation at the time of purchase. However, you may reorient the system later as needed. **Failure to place the pump and plumbing properly can trap air, preventing the pump from working properly.**

Shown below are schematics of the system configuration for vertical and horizontal deployment.

## Vertical Mount

- The main plumbing is 13 mm ID x 19 mm OD (1/2 inch x 3/4 inch) Tygon tubing.
- Place the exhaust as far from the intake as possible, preferably in diagonally opposing corners, so that exhaust water is not pulled into the intake. Failure to place the exhaust away from the intake can lead to errors in temperature data, because the pump transfers heat to the exhaust water.
- Place a 13 mm (0.5 inch) piece of 9.5 mm ID Tygon tubing at the dissolved oxygen (DO) sensor intake and exhaust. Slide the larger diameter Tygon tubing (13 mm ID) over the smaller diameter tubing to provide tight seals.
- If the system does not include a DO sensor, connect the tubing from the conductivity cell directly to the Y-fitting.



SBE 43

Exhaust

SBE 43 plenum

Intake

Place DO sensor with plenum intake closer than exhaust to SBE 43 housing, as shown. Note: SBE 43 housing orientation (connector end up or down) does not affect operation. Plenum can be reversed on housing by removing and replacing 2 screws.

Secure each tubing connection with 2 cable ties (not shown)

Approximately 100 mm (4 inch) length of Tygon tubing

Y-fitting and air release valve

Air release valve detail

0.5 mm (0.02 in.)    UP

**Note**: Periodically clean air release valve to ensure proper functioning.

SBE 43 DO sensor

Tygon tubing (typical)

DO sensor intake

Quick release coupling

Pump intake

Exhaust

SBE 5T or 5P pump

DO sensor intake and exhaust detail

9.5 mm inner diameter Tygon tubing (13 mm long)

DO plenum hose

13 mm inner diameter Tygon tubing

Temperature sensor

Conductivity sensor

TC Duct

Intake

Place intake and exhaust far enough away from each other to avoid recirculation of water

21

## Horizontal Mount

- The main plumbing is 13 mm ID x 19 mm OD (1/2 inch x 3/4 inch) Tygon tubing.
- Place the DO sensor intake above the conductivity sensor exhaust.
- Place the pump intake above the DO sensor exhaust.
- Orient the pump with the exhaust outlet *corner* up.
- Place a 13 mm (0.5 inch) piece of the 9.5 mm ID Tygon tubing at the DO sensor intake and exhaust. Slide the larger diameter tubing (13 mm ID) over the smaller diameter tubing to provide tight seals.
- If the system does not include a DO sensor, connect the tubing from the conductivity cell directly to the pump intake.

Place pump exhaust above intake and DO sensor, with exhaust outlet 'corner' up

Exhaust

Secure each tubing connection with 2 cable ties (not shown)

Tygon tubing

Pump intake

SBE 5T or 5P pump

Conductivity sensor

TC Duct

SBE 43 DO sensor

Temperature sensor

Intake

SBE 43 plenum

Quick release coupling

Exhaust          Intake

SBE 43 DO sensor

1. Place DO sensor with exhaust above intake, oriented as shown (± 45°).
2. Place DO sensor with plenum intake closer than exhaust to SBE 43 housing, as shown.
Note: SBE 43 housing orientation (connector end to left or to right) does not affect operation. Plenum can be reversed on housing by removing and replacing 2 screws.

DO sensor intake and exhaust detail

DO plenum hose

9.5 mm inner diameter Tygon tubing (13 mm long)

13 mm inner diameter Tygon tubing

# Section 3: Power and Communications Test

This section describes software installation and the pre-check procedure for preparing the SBE 25*plus* for deployment. The power and communications test will verify that the system works, prior to deployment.

## Software Installation

**Notes:**
- Help files provide detailed information on the software. Separate software manuals on the CD-ROM contain detailed information on Seasave V7 and SBE Data Processing.
- The older version of Seasave, Seasave-Win32, is not compatible with the 25*plus*.
- It is possible to use the 25*plus* without the SeatermV2 terminal program by sending direct commands from a dumb terminal or terminal emulator, such as Windows HyperTerminal.
- Sea-Bird supplies the current version of our software when you purchase an instrument. As software revisions occur, we post the revised software on our website. See our website for the latest software version number, a description of the software changes, and instructions for downloading the software.

Seasoft V2 was designed to work with a PC running Windows XP service pack 2 or later, Windows Vista, or Windows 7 (32-bit or 64-bit).

If not already installed, install Sea-Bird software programs on your computer using the supplied software CD:

1. Insert the CD in your CD drive.

2. Install software: Double click on **SeasoftV2.exe**. Follow the dialog box directions to install the software. The installation program allows you to install the desired components. Install all the components, or just install SeatermV2 (terminal program *launcher* for 25*plus*), SeatermAF (terminal program for autonomous water sampler setup), Seasave V7 (real-time data acquisition), and SBE Data Processing (data processing).

The default location for the software is c:\Program Files\Sea-Bird. Within that folder is a sub-directory for each program.

## Test Setup



J7 Data I/O, pump, & external power (connects to Y-cable)

1. Remove the dummy plug from the Y-cable and install the I/O cable:
   A. Pulling the plug firmly away from the connector, remove the dummy plug from the 25*plus*' Y-cable connector strapped to the Main housing near the top of the cage. Note that the Y-cable is connected to the data I/O, pump, and external power connector, and provides power to the pump as well as communications with the computer.
   B. **XSG / AG Connector** - Install the I/O cable connector, aligning the raised bump on the side of the connector with the large pin (pin 1 - ground) on the Y-cable.    **OR**
   **MCBH Connector** – Install the I/O cable connector, aligning the pins.

2. Connect the I/O cable connector to your computer's serial port.

## Test

**Note:**
See SeatermV2's help files.

1.  Double click on **SeatermV2.exe**. The main screen looks like this
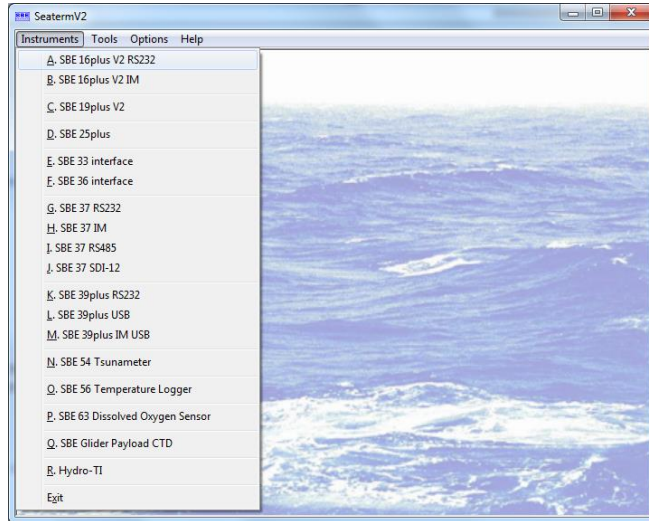
    

    SeatermV2 is a *launcher*, and launches the appropriate terminal program for the selected instrument.

**Note:**
See Seaterm232's Help files.

2.  In the Instruments menu, select *SBE 25plus*.
    **Seaterm232** opens; the main screen looks like this:
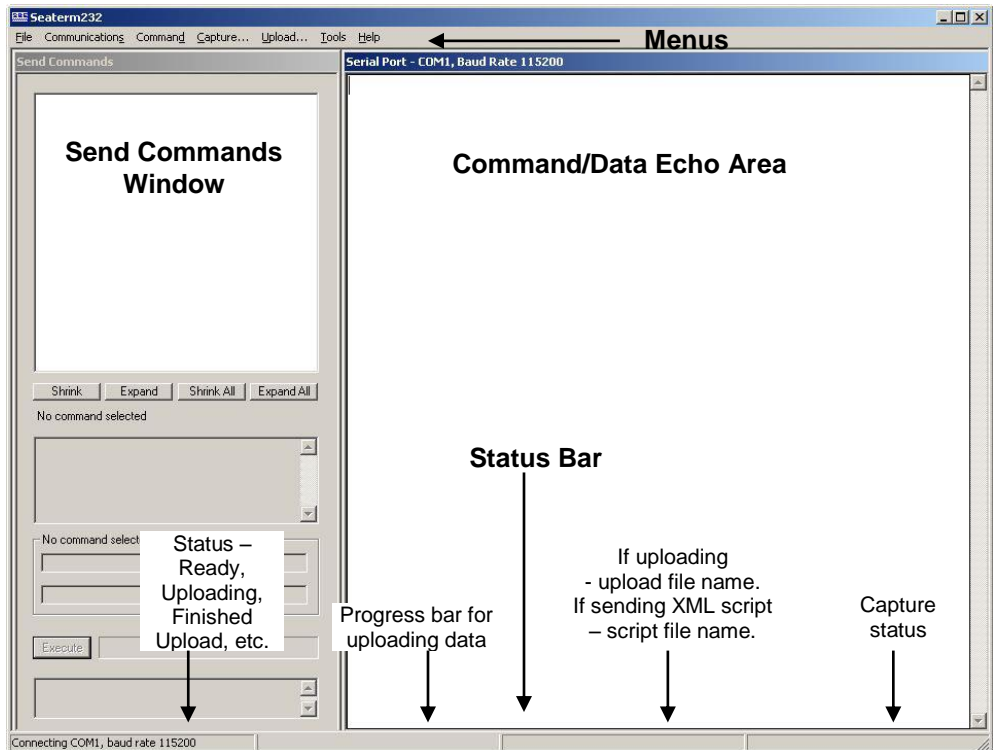
    

*   Menus – For tasks and frequently executed instrument commands.
*   Send Commands window – Contains commands applicable to your 25*plus*. The list appears after you connect to the 25*plus*.
*   Command/Data Echo Area – Title bar of this window shows Seaterm232's current comm port and baud rate. Commands and the 25*plus* responses are echoed here. Additionally, a command can be manually typed or pasted (ctrl + V) here. Note that the 25*plus* must be *connected* and *awake* for it to respond to a command.
*   Status bar – Provides connection, upload, script, and capture status information.

24

Following is a description of the menus:

| Menu | Description | Equivalent Command* |
|---|---|---|
| File | • Load command file – open selected .XML command file, and fill Send Commands window with commands<br>• Unload command file – close command file, and remove commands from Send Commands window<br>• Exit - Exit program. | - |
| Communications | • Connect – connect to comm port<br>• Disconnect – disconnect from comm port<br>• Configure – Establish communication parameters (comm port and baud rate).<br>• Disconnect and reconnect – may be useful if 25*plus* has stopped responding | - |
| Command | • Abort – interrupt and stop 25*plus* response<br>• Send 5 second break (not applicable to 25*plus*)<br>• Send stop command<br>• Set local time and Set UTC Time (Greenwich Mean Time) – Both of these selections set date and time to **UTC** time sent by timekeeping software on your computer; accuracy ± 25 msec of time provided by computer. | • (press Esc key several times for Abort)<br><br>• **Stop**<br>• **SetDateTime=** |
| Capture | Capture 25*plus* responses on screen to file, to save real-time data or use for diagnostics. File has .cap extension.<br>Click Capture again to turn off capture. Capture status displays in Status bar. | — |
| Upload | Upload data stored in memory, in format Sea-Bird's data processing software can use (raw hex). Uploaded data has .xml extension, and is then automatically converted to a file with .hex extension. Before using Upload: **stop logging** by sending **Stop**. | Several status commands and appropriate data upload command as applicable to user selection of range of data to upload (use Upload menu if you will be processing data with SBE Data Processing or viewing archived data with Seasave) |
| Tools | • Diagnostics log - Keep a diagnostics log.<br>• Convert .XML data file (not applicable to 25*plus*. Data Conversion module in SBE Data Processing uses .xml file uploaded from 25*plus* memory.)<br>• Send script – Send XML script to 25*plus*. May be useful if you have a number of 25*plus* to program with same setup. | - |

**Note:**
*Set local time* and *Set UTC time* are disabled if the baud rate in Seaterm232 is set to 115200, because the software cannot reliably set the time at that baud.

*See *Command Descriptions* in *Section 4: Deploying and Operating SBE 25plus*.

3. If this is the first time Seaterm232 is being used, the Serial Port Configuration dialog box displays:

Computer COM port and baud rate for communication between computer and 25*plus*. Seaterm232 tries to connect at this baud rate, but if unsuccessful will cycle through all available baud rates.

**Serial Port Configuration**

Port: COM1

Baud: 9600

OK      Cancel      Help

Update COM Port pulldown to include connected USB ports.

Discover Serial Ports
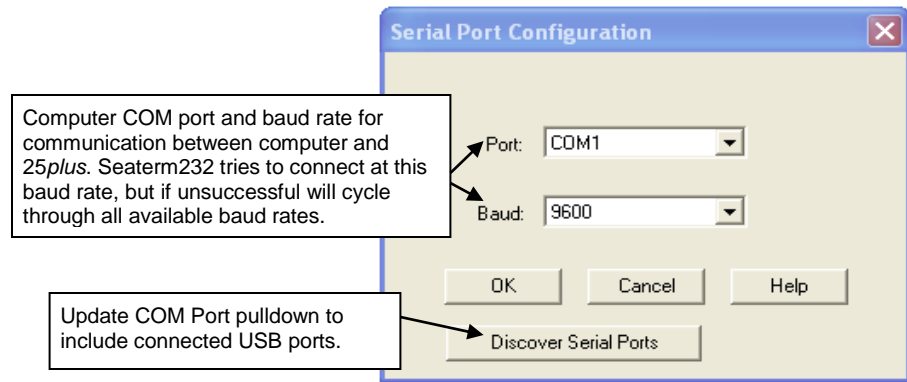
Make the desired selections, and click OK.

**Note:**
Seaterm232's baud rate must be the same as the 25*plus* baud rate (set with **SetBaudConsole=**). Baud is factory-set to 9600, but can be changed by the user (see *Command Descriptions* in *Section 4: Deploying and Operating SBE 25plus*). Other communication parameters –
8 data bits, 1 stop bit, and no parity – cannot be changed.

4. Seaterm232 tries to automatically connect to the 25*plus*. As it connects, it sends **GetHD** and displays the response, which provides factory-set data such as instrument type, serial number, and firmware version. Seaterm232 also fills the Send Commands window with the correct list of commands for your 25*plus*.
**If there is no communication:**

A. In the Communications menu, select *Configure*. The Serial Port Configuration dialog box appears. Select the Comm port and baud rate for communication, and click OK. Note that the factory-set baud rate is documented on the Configuration Sheet.
B. In the Communications menu, select *Connect* (if *Connect* is grayed out, select *Disconnect and reconnect*). Seaterm232 will attempt to connect at the baud specified in Step A, but if unsuccessful will then cycle through all other available baud rates.
C. If there is still no communication, check cabling between the computer and 25*plus*, and try to connect again.
D. If there is still no communication, repeat Step A with a different comm port, and try to connect again.

After Seaterm232 displays the **GetHD** response, it provides an S> prompt to indicate it is ready for the next command.

Taking a look at the Send Commands window:



Click on desired command description in list.

Help box describes selected command in more detail.

Enter any command arguments (such as starting and ending character number) for upload) in these boxes.

Click Execute when ready to send selected command.

This box shows selected command.

You can use the Send Commands window to send commands, or simply type the commands in the Command/Data Echo area if desired.

**Note:**

The 25*plus* automatically enters quiescent (sleep) state after 2 minutes without receiving a command. This timeout is designed to conserve battery energy if the user does not send **QS** to put the 25*plus* to sleep. If the 25*plus* does not appear to respond, click Connect in the Communications menu to reestablish communications.

5.   Display 25*plus* setup information by typing **GetCD** and pressing the Enter key. The display looks like this:

```
S>getcd
<ConfigurationData DeviceType='SBE25plus' SerialNumber='0250003'>
   <Serial>
      <SerialPort0>
         <baudconsole>9600</baudconsole>
         <echoconsole>1</echoconsole>
         </SerialPort0>
      <SerialPort1>
         <enableser1>0</enableser1>
         <nameser1>serial1</nameser1>
         <baudser1>9600</baudser1>
         <inlineser1>1</inlineser1>
         <promptser1>S</promptser1>
         <measstringser1>start</measstringser1>
         <measintervalser1>0</measintervalser1>
         <startstringser1>ts</startstringser1>
         <execcharser1>13</execcharser1>
         <termcharser1>10</termcharser1>
         <suppressser1>1</suppressser1>
         <timeoutser1>30</timeoutser1>
         <failoutser1>60</failoutser1>
         <echoser1>0</echoser1>
         </SerialPort1>
      <SerialPort2>
         <enableser2>0</enableser2>
         <nameser2>serial2</nameser2>
         <baudser2>9600</baudser2>
         <inlineser2>1</inlineser2>
         <promptser2>S</promptser2>
         <measstringser2>meas</measstringser2>
         <measintervalser2>16</measintervalser2>
         <startstringser2>gethd</startstringser2>
         <execcharser2>13</execcharser2>
         <termcharser2>10</termcharser2>
         <suppressser2>1</suppressser2>
         <timeoutser2>30</timeoutser2>
         <failoutser2>60</failoutser2>
         <echoser2>0</echoser2>
         </SerialPort2>
      </Serial>
   <Settings>
      <vauxdelay0>0</vauxdelay0>
      <vauxdelay1>0</vauxdelay1>
      <vauxdelay2>0</vauxdelay2>
      <vauxdelay3>0</vauxdelay3>
      <mincondfreq>3000</mincondfreq>
      <pumpdelay>60</pumpdelay>
      <executedtag>1</executedtag>
      </Settings>
   <RealTimeOutput>
      <outputformat>0</outputformat>
      <historicrate>1</historicrate>
      <vout0>0</vout0>
      <vout1>0</vout1>
      <vout2>0</vout2>
      <vout3>0</vout3>
      <vout4>0</vout4>
      <vout5>0</vout5>
      <vout6>0</vout6>
      <vout7>0</vout7>
      <outputrate>8</outputrate>
      </RealTimeOutput>
   </ConfigurationData>
<Executed/>
```

**Note:**

In response to **TS**, the 25*plus:*

- Always outputs all 8 voltage channels, regardless of which channels are enabled for real-time output with the **SetVOut#=** commands.
- Never outputs serial sensor data**.**

**Note:**

The response time to **TS** is dependent on the power-on delay for the auxiliary sensors [**SetVAuxDelay#=** (#=0, 1, 2, or 3)]. The response can be aborted by pressing the Esc key.

6. Command the 25*plus* to take a sample by typing **TS** and pressing the Enter key. The display looks like this:

0000000000040007000500000005000300060006007599B0008053B34597F32B45E135FE

with 72 Hex characters displayed:
(iiiiiiiivvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv00pppppp00ppppppcccccccctttttttt)

iiiiiiii = diagnostic information
vvvv = external voltage sensor 7 voltage
vvvv = external voltage sensor 6 voltage
vvvv = external voltage sensor 5 voltage
vvvv = external voltage sensor 4 voltage
vvvv = external voltage sensor 3 voltage
vvvv = external voltage sensor 2 voltage
vvvv = external voltage sensor 1 voltage
vvvv = external voltage sensor 0 voltage
00pppppp = pressure temperature counts
00pppppp = pressure sensor counts
cccccccc = conductivity (Hz)
tttttttt = temperature (Hz)

7. Command the 25*plus* to go to sleep (quiescent state) by typing **QS** and pressing the Enter key.

The SBE 25*plus* is ready for programming and deployment.

# Section 4: Deploying and Operating SBE 25*plus*

**Note:**
Separate software manuals and Help files contain detailed information on installation, setup, and use of Sea-Bird's real-time data acquisition software (Seasave) and data processing software (SBE Data Processing).

This section includes discussions of:
- Logging operation
- Pump operation
- Real-time setup
- Timeout description
- Command descriptions
- Data format
- Optimizing data quality
- Setup for deployment
- Deployment
- Acquiring real-time data with Seasave
- Multiple deployments
- Recovery - physical handling and uploading data
- Processing data

## Logging

**Note:**
The 25*plus* cannot transmit real-time data from a serial output sensor.

When logging, the SBE 25*plus* samples data at 16 Hz (16 samples/second), stores the data in its memory, and transmits the data in real-time at 0.5 to 8 Hz (real-time output rate dependent on baud rate and number of auxiliary voltage sensors enabled for real-time output; see *Real-Time Setup*). Each time the 25*plus* begins logging, it writes header information to the beginning of the data file, containing the date and time that logging was started. When the memory is full, the 25*plus* will continue to operate and transmit real-time data. However, the data will not be recorded in memory and will not overwrite previously recorded data.

Logging is started by:
- Pushing the plunger switch in. Data is stored in memory.
  **OR**
- Sending **StartNow**. Data is stored in memory.

**Note:**
The 25*plus* automatically enters quiescent (sleep) state after 2 minutes without receiving a command.

Power is applied as follows:
- Temperature, conductivity, and pressure sensors - Power is applied when the 25*plus* is woken up (by switch or by command).
- Pump - Power is applied *pump delay* seconds (**SetPumpDelay=**) after the *minimum conductivity frequency* (**SetMindCondFreq=)** is reached (see *Pump Operation*).
- Auxiliary voltage sensors and Auxiliary serial sensors - Power is applied **SetVAuxDelay#=** seconds (#= 0, 1, 2, 3) after logging begins. This enables you to stagger powering up of the auxiliary sensors, so that the total inrush current does not exceed the maximum current rating (1.2 Amps). The numbering scheme for the command corresponds to the numbering on the end cap: **SetVAuxDelay0=** is for J0 (voltage channels 0 and 1), **SetVAuxDelay1=** is for J1 (voltage channels 2 and 3), etc. **SetVAuxDelay3=** controls the power application to J3 (voltage channels 6 and 7) and also to J6 (serial sensors 1 and 2).

***As soon as logging is started***, the first data point is stored in memory and the cast header date and time is written. Therefore, the memory will include *bad* data at the beginning of the cast, because:

- sensors depending on the pump (conductivity, optional dissolved oxygen, etc.) will not provide *good* data until the pump turns on, and
- auxiliary sensors will not provide *good* data until power is applied.

When processing data after the cast, you can eliminate these initial scans in memory by setting up the post-processing software to *skip* the scans.

> **Note:**
> See SBE Data Processing's manual or Help files.

Logging is stopped by one of the following methods:

- Pull the plunger switch out to the Off position - stops logging and puts 25*plus* into quiescent (sleep) state.
  **OR**
- Send **Stop** in a terminal program – stops logging. The 25*plus* automatically enters quiescent (sleep) state after 2 minutes without receiving a command.

After recovery, data can be uploaded with one of the following methods:

> **Note:**
> Some USB cable connectors do not seat properly on the USB port in the 25*plus* battery pack, preventing communications. Because the power and ground pins are longer, the indicator light may come on even if all of the pins are not securely connected. If you cannot establish communications via the USB port, try again with a different USB cable.

- With the computer connected to the Y-cable from the 25*plus* J7 connector, use Seaterm232's Upload menu to upload one or more files via RS-232.
  **OR**
- (**Faster, recommended for large amounts of data**) With the 25*plus* battery end cap removed and a USB cable connected to the USB port in the battery pack, copy the files in memory and paste them to your computer.

Review the operation of the commands described in *Command Descriptions* before setting up your system.

---

*Example:*

Wake up 25*plus*. Set date and time to UTC October 6, 2012 at 9:05 am. Delete all existing data in memory. 25*plus* has voltage sensors on channels 0, 1, 2, and 3, and you want to view data from all of them in real time. Output data in format compatible with Seasave. Set up with a minimum conductivity frequency of 3000 Hz and 60-second pump turn-on delay after 25*plus* enters water, to ensure pump is primed before turning on. After all parameters are entered, verify setup with status command. Send power-off command.

(Click Connect in Seaterm232's Communications menu to connect and wake up.)
```
S>SETDATETIME=2012-10-06T09:05:00
S>DELETEALL
S>SETVOUT0=Y
S>SETVOUT1=Y
S>SETVOUT2=Y
S>SETVOUT3=Y
S>SETOUTPUTFORMAT=0
S>SETMINCONDFREQ=3000
S>SETPUMPDELAY=60
S>GETCD          (to verify setup)
S>QS
```

Start logging by pushing plunger switch in to On position.
Put 25*plus* in water, and allow to soak for at least time required for pump turn-on (**SetPumpDelay=60**) before beginning downcast. If desired, use Seasave to view real-time data. When cast is complete, stop logging by pulling switch out to Off position.

Upload data in memory, in format SBE Data Processing and Seasave can use. Send power-off command.

(Click Connect in Seaterm232's Communications menu to connect and wake up.)
(Click Upload – Seaterm232 leads you through screens to define data to be uploaded and where to store it.)
```
S>QS
```

# Pump Operation
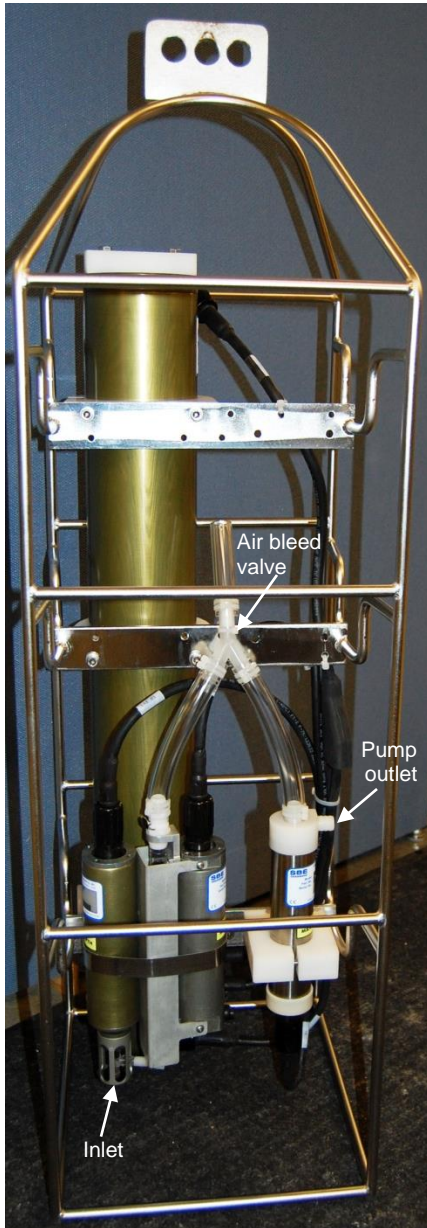


Plumbing for Vertical Orientation
(see *Configuration Options and Plumbing* in
*Section 2: Description of SBE 25plus* for
plumbing for horizontal orientation)

After the conductivity cell enters the water, there is a user-programmable delay before pump turn-on so that all the air in the pump tubing can escape. If the pump motor turns on when there is air in the impeller housing, priming is uncertain and a proper flow rate cannot be ensured. When the SBE 25*plus* is deployed in a vertical orientation, the tubing extending above the air-bleed hole will contain a small reserve of water. This maintains the pump prime (for up to 1 minute, depending on the length of tubing above the air-bleed), even if the 25*plus* is lifted up so that the cell inlet and pump outlet are **just below** the water surface. This allows beginning the actual profile very near the top of the water. **The cell inlet and pump outlet must not come above the water surface or the prime will be lost.**

- If prime is lost, pull the plunger switch out to the Off position. Wait at least 3 seconds, then push the switch in to the On position, submerge the 25*plus* completely, and wait for the pump delay time before beginning the profile.

Pump turn-on occurs when two user-programmable conditions have been met:

- **Raw conductivity frequency exceeds the minimum conductivity frequency** (**SetMindCondFreq=**)
  Set the minimum conductivity frequency for pump turn-on above the instrument's *zero conductivity raw frequency* (shown on the 25*plus* Configuration Sheet), to prevent the pump from turning on when the 25*plus* is in air.
  - For salt water and estuarine applications - typical value = *zero conductivity raw frequency* + 500 Hz
  - For fresh/nearly fresh water - typical value = *zero conductivity raw frequency* + 5 Hz

  If the minimum conductivity frequency is too close to the *zero conductivity raw frequency*, the pump may turn on when the 25*plus* is in air as result of small drifts in the electronics. Some experimentation may be required, and in some cases it may be necessary to rely only on the pump turn-on delay time to control the pump. If so, set a minimum conductivity frequency lower than the *zero conductivity raw frequency*.

- **Pump turn-on delay time has elapsed** (**SetPumpDelay=**)
  Set the pump turn-on delay time to allow time for the Tygon tubing and pump to fill with water after the 25*plus* is submerged. Determine the turn-on delay by immersing the 25*plus* (switch out, not running) just below the air-bleed hole at the top of the Tygon tubing (vertical orientation) or just below the pump exhaust (horizontal orientation). Measure the time needed to completely fill the tubing (by watching for when air bubbles stop coming of the air bleed valve or pump exhaust); 30 seconds is typical. Set the delay to at least 1.5 times longer. When actually using the 25*plus*, be sure to *soak* the instrument just under the surface for at least the time required for pump turn-on.

# Real-Time Setup

On the ship, cables longer than 3 meters should be installed inside an earthed metal conduit by a qualified electrician. This minimizes the potential for external signals to disrupt communication and ensures that high voltage lines (such as the sea cable) are sufficiently protected. Cables shorter than 3 meters can be used without shielding when installing or bench testing the instrument.

## Baud Rate, Cable Length, Voltage Channels, and Real-Time Output Rate

**Notes:**

- **The 25*plus* cannot transmit real-time data from a serial output sensor.**
- The 25*plus* calculates the real-time data output rate based on **SetBaudConsole=** and **SetVOut#=** (#=0-7). The calculated rate is output at the bottom of the **GetCD** response, with an <outputrate> tag.

The SBE 25*plus* real-time data output rate varies from 0.5 to 8 Hz, depending on the baud rate and the number of auxiliary voltage sensors enabled for real-time output. Transmission distance is dependent on number of auxiliary voltage sensors enabled for real-time output, baud rate, and cable properties.

The SBE 25*plus* always samples and stores data in memory at 16 Hz. For real-time data, it averages the appropriate number of scans and outputs the average. To illustrate the averaging and output scheme, the table below provides an example for a **4 Hz** real-time output rate:

| Internal Storage (16 Hz) | Real-Time Output (4 Hz for this example) |
|---|---|
| Scan 0 | - |
| Scan 1 | - |
| Scan 2 | - |
| Scan 3 | - |
| Scan 4 | (calculate average scans 0 to 3) |
| Scan 5 | Output average scans 0 to 3 |
| Scan 6 | - |
| Scan 7 | - |
| Scan 8 | (calculate average scans 4 to 7) |
| Scan 9 | Output average scans 4 to 7 |

### *Without a Sea-Bird Deck Unit*

When used without a Sea-Bird deck unit, the rate that real-time data can be transmitted from the SBE 25*plus* is dependent on the number of auxiliary voltage sensors being transmitted in real-time and the serial data baud rate. The length of cable that the 25*plus* can drive is also dependent on the baud rate. The allowable combinations are:

| Baud Rate | Voltage Channels Enabled for Real-Time Output | Real-Time Output (scans/second) | Maximum Cable Length (meters) |
|---|---|---|---|
| 600 | 0 | 1 | 1600 |
| | 1 - 8 | 0.5 | |
| 1200 | 0 | 2 | 800 |
| | 1 - 8 | 1 | |
| 2400 | 0 | 2 | 400 |
| | 1 - 8 | 1 | |
| 4800 | 0 | 8 | 200 |
| | 1 - 8 | 4 | |
| 9600 | 0 - 8 | 8 | 100 |
| 14400 | 0 – 8 | 8 | 66 |
| 19200 | 0 – 8 | 8 | 50 |
| 38400 | 0 – 8 | 8 | 25 |
| 57600 | 0 – 8 | 8 | 4 |
| 115200 | 0 – 8 | 8 | 8 |

The SBE 25*plus* **automatically** adjusts the real-time output rate to the value shown in the table, based on the baud rate (**SetBaudConsole=**) and the number of voltage channels enabled for real-time output (**SetVOut0=** through **SetVOut7=**). Note that the maximum real-time output rate is 8 Hz, while the 25*plus* always takes measurements and stores data in memory at 16 Hz.

### *With a Sea-Bird Deck Unit*

Set the SBE 25*plus* baud rate to 4800 (**SetBaudConsole=4800**) if using the 25*plus* with either of the following real-time data acquisition systems:
- SBE 36 CTD Deck Unit and Power and Data Interface Module (PDIM)
- SBE 33 Carousel Deck Unit and SBE 32 Carousel Water Sampler **OR** SBE 33 Carousel Deck Unit and SBE 55 ECO Water Sampler

The data telemetry link can drive 10,000 meters of cable while accepting 4800 baud serial data. The relationship between the number of auxiliary voltage sensor channels being transmitted in real-time, whether NMEA and Surface PAR are appended in the Deck Unit, and the real-time output data rate is shown below:

| Voltage Channels Enabled for Real-Time Output | Real-Time Output (scans/sec) through SBE 33 or 36, no NMEA & PAR (SetHistoricRate=1) | Real-Time Output (scans/sec) through SBE 33 or 36, with NMEA & PAR appended (SetHistoricRate=0) |
|---|---|---|
| 0 | 8 | 4 |
| 1 - 8 | 4 | 2 |

*Note that you must set **SetHistoricRate=0** in the SBE 25plus to provide a slower transmission rate if appending NMEA and Surface PAR in the deck unit, because of deck unit limitations.*

## Real-Time Data Acquisition

> **Note:**
> If the FLASH memory is filled to capacity, data sampling and transmission of real-time data (if programmed) continue, but excess data is not saved in memory.

Real-time data can be acquired in either of the following ways:

- With Seasave (**typical method**) – Data can be viewed in Seasave in tabular form or as plots, as raw data or as converted (engineering units) data. Data acquired with Seasave can be processed with SBE Data Processing. See Seasave's manual / Help files for setting up the program displays, baud rates, etc., and beginning data acquisition.
- With Seaterm232 – Click the Capture menu; enter the desired file name in the dialog box, and click Save. Begin logging. The data displayed in Seaterm232 will be saved to the designated file. Process the data as desired. Note that this file **cannot be processed by SBE Data Processing or displayed in Seasave, as it does not have the required headers and format for Sea-Bird's processing software.**

# Timeout Description

The SBE 25*plus* has a timeout algorithm. If the 25*plus* does not receive a command or sample data for 2 minutes, it powers down its main digital circuits. This places the 25*plus* in quiescent (sleep) state, drawing minimal current. **To re-establish control (wake up), click Connect in Seaterm232's Communications menu or press the Enter key.**

# Command Descriptions

This section describes commands and provides sample outputs.
See *Appendix III: Command Summary* for a summarized command list.

When entering commands:

- Input commands to the 25*plus* in upper or lower case letters and register commands by pressing the Enter key.
- Commands to enable a parameter or output (such as enabling real-time output from a voltage channel) can be entered with the *argument* as Y or 1 for yes, and N or 0 for no (for example, **SetVOut0=y** and **SetVOut0=1** are equivalent; both enable real-time output from voltage channel 0).
- The 25*plus* sends an error message if an invalid command is entered.
- If the 25*plus* does not return an S> prompt after executing a command, press the Enter key to get the S> prompt.
- If a new command is not received within 2 minutes after the completion of a command, the 25*plus* returns to the quiescent (sleep) state and the display shows *2 min inactivity time out, returning to sleep*.
- If in quiescent state, re-establish communications by selecting *Connect* in Seaterm232's Communications menu or pressing the Enter key.
- The 25*plus* does not respond to any commands while logging.
- If the 25*plus* is uploading data and you want to stop it, press the Esc key; then press the Enter key. Alternatively, select *Abort* in Seaterm232's Command menu.

Entries made with the commands are permanently stored in the SBE 25*plus* and remain in effect until you change them.

**Status** Commands

---

> **Note:**
> **GetCD** output does not include calibration coefficients. To display calibration coefficients, use the **GetCC** command.

**GetCD**

Get and display configuration data, which includes all parameters related to setup of 25*plus*, including communication settings and sampling settings. Most of these parameters can be user-input/modified. List below includes, where applicable, command used to modify parameter:

- Device type, Serial number
- Serial port setup for communication with computer
  - Baud rate [**SetBaudConsole=**]
  - Echo commands as typed [**SetEchoConsole=**]?
- Serial sensor 1 (Ser1) setup
  - Enable sensor [**SetEnableSer1=**]?
  - Sensor name [**SetNameSer1=**]
  - Sensor baud [**SetBaudSer1=**]
  - Store sensor data in file with CTD and voltage data [**SetInlineSer1=**]?
  - Sensor prompt [**SetPromptSer1=**]
  - Sensor measurement string [**SetMeasStringSer1=**]
  - Sensor measurement string interval [**SetMeasIntervalSer1=**]
  - Sensor start string [**SetStartStringSer1=**]
  - Sensor execute character [**SetExecCharSer1=**]
  - Sensor termination character [**SetTermCharSer1=**]
  - Suppress termination character and prompt [**SetSupressSer1=**]?
  - Sensor timeout for command response [**SetTimeoutSer1=**]
  - Sensor timeout for data [**SetFailoutSer1=**]
  - factory set echo setup
- Serial sensor 2 (Ser2) setup – see description above for sensor 1
- Miscellaneous settings
  - Power supply delay to J0 (voltage channels 0, 1) [**SetVauxDelay0=**]
  - Power supply delay to J1 (voltage channels 2, 3) [**SetVauxDelay1=**]
  - Power supply delay to J2 (voltage channels 4, 5) [**SetVauxDelay2=**]
  - Power supply delay to J3 (voltage channels 6, 7) and J6 (serial channels 1,2) [**SetVauxDelay3=**]
  - Minimum conductivity frequency for pump turn-on [**SetMinCondFreq=**]
  - Pump turn-on delay [**SetPumpDelay=**]
  - Output Executing and Executed tags? [**OutputExecutedTag=**]
- Real-time output setup
  - Real-time output format when logging [**SetOutputFormat=**]
  - Output at 8 Hz if baud is 4800 [**SetHistoricRate=**]?
  - Output external voltage channel data (0, 1, 2, 3, 4, 5, 6, 7) in real-time? [**SetVOut0=** through **SetVOut7=**]
  - Real-time data output rate (scans/sec) [calculated internally based on **SetBaudConsole=** and **SetVOut0=** through **SetVOut7=**]

**Status** Commands (*continued*)

*Example:* (user input in bold, command used to modify parameter in parentheses at right)

```
s>getcd
<ConfigurationData DeviceType='SBE25plus' SerialNumber='0250003'>
   <Serial>
      <SerialPort0>
         <baudconsole>9600</baudconsole>                            [SetBaudConsole=]
         <echoconsole>1</echoconsole>                               [SetEchoConsole=]
         </SerialPort0>
      <SerialPort1>
         <enableser1>1</enableser1>                                 [SetEnableSer1=]
         <nameser1>serial1</nameser1>                               [SetNameSer1=]
         <baudser1>38400</baudser1>                                 [SetBaudSer1=]
         <inlineser1>1</inlineser1>                                 [SetInlineSer1=]
         <promptser1>S</promptser1>                                 [SetPromptSer1=]
         <measstringser1>@</measstringser1>                         [SetMeasStringSer1=]
         <measintervalser1>0</measintervalser1>                     [SetMeasIntervalSer1=]
         <startstringser1>fastmode</startstringser1>                [SetStartStringSer1=]
         <execcharser1>13</execcharser1>                            [SetExecCharSer1=]
         <termcharser1>10</termcharser1>                            [SetTermCharSer1=]
         <suppressser1>1</suppressser1>                             [SetSuppressSer1=]
         <timeoutser1>4</timeoutser1>                               [SetTimeoutSer1=]
         <failoutser1>30</failoutser1>                              [SetFailoutSer1=]
         <echoser1>0</echoser1>
         </SerialPort1>
      <SerialPort2>
         <enableser2>1</enableser2>                                 [SetEnableSer2=]
         <nameser2>serial1</nameser2>                               [SetNameSer2=]
         <baudser2>38400</baudser2>                                 [SetBaudSer2=]
         <inlineser2>1</inlineser2>                                 [SetInlineSer2=]
         <promptser2>S</promptser2>                                 [SetPromptSer2=]
         <measstringser2>@</measstringser2>                         [SetMeasStringSer2=]
         <measintervalser2>0</measintervalser2>                     [SetMeasIntervalSer2=]
         <startstringser2>fastmode</startstringser2>                [SetStartStringSer2=]
         <execcharser2>13</execcharser2>                            [SetExecCharSer2=]
         <termcharser2>10</termcharser2>                            [SetTermCharSer2=]
         <suppressser2>1</suppressser2>                             [SetSuppressSer2=]
         <timeoutser2>4</timeoutser2>                               [SetTimeoutSer2=]
         <failoutser2>30</failoutser2>                              [SetFailoutSer2=]
         <echoser2>0</echoser2>
         </SerialPort2>
      </Serial>
   <Settings>
      <vaux0delay>0</vaux0delay>                                    [SetVAuxDelay0=]
      <vaux1delay>0</vaux1delay>                                    [SetVAuxDelay1=]
      <vaux2delay>0</vaux2delay>                                    [SetVAuxDelay2=]
      <vaux3delay>0</vaux3delay>                                    [SetVAuxDelay3=]
      <mincondfreq>0</mincondfreq>                                  [SetMinCondFreq=]
      <pumpdelay>60</pumpdelay>                                     [SetPumpDelay=]
      <executedtag>0</executedtag>                                  [SetExecutedTag=]
      </Settings>
   <RealTimeOutput>
      <outputformat>0</outputformat>                                [SetOutputFormat=]
      <historicrate>1</historicrate>                                [SetHistoricRate=]
      <vout0>0</vout0>                                              [SetVOut0=]
      <vout1>0</vout1>                                              [SetVOut1=]
      <vout2>0</vout2>                                              [SetVOut2=]
      <vout3>0</vout3>                                              [SetVOut3=]
      <vout4>0</vout4>                                              [SetVOut4=]
      <vout5>0</vout5>                                              [SetVOut5=]
      <vout6>0</vout6>                                              [SetVOut6=]
      <vout7>0</vout7>                                              [SetVOut7=]
      <outputrate>8</outputrate>       [calculated internally, based on SetBaudConsole=, SetVOut0= through SetVOut7=]
      </RealTimeOutput>
   </ConfigurationData>
```

**InitCD**        Reset all user-programmable parameters that appear in **GetCD** response to factory defaults. **Requires confirmation.**

**Notes:**

- Sending **GetSD** causes the pump to turn on for a moment, so that the 25*plus* can measure and output the pump current. Because the pump is designed to be water lubricated, you will hear a noise when the impeller spins in air. Running the pump *dry* for such a **short** time will not harm the pump.
- If there are a large number of files in memory, the 25*plus* can take up to approximately 45 seconds to complete the **GetSD** response, because it must go through the files in memory sequentially to output the MemorySummary portion of the response. Click the Esc key to abort the response.

**Status** Commands (*continued*)

**GetSD**   Get and display status data, which contains data that changes while deployed. List below includes, where applicable, command used to modify parameter:

- Device type, Serial number
- Date and time [**SetDateTime**=] in ISO8601-2000 extended format (yyyy – mm-ddThh:mm:ss)
- Number of recorded events in event counter [see details with **GetEC**; reset with **ResetEC**]
- Voltages and currents -
  - Main battery pack voltage
  - External power input voltage
  - Back-up lithium battery voltage
  - Real-time clock battery voltage
  - Sum of currents for all auxiliary (voltage and serial) sensors
  - Sum of currents for C, T, and P sensors and pump
  - Sum of currents for processors
- Memory - [reset with **InitLogging**]
  - Number of bytes in memory
  - Number of additional bytes that can be placed in memory
  - Approximate number of samples in memory
  - Approximate number of additional samples that can be placed in memory
  - Total number of files in memory (there may be multiple files for each cast, if using serial sensors and sensor data is saved in separate files (see **SetInlineSer#=**)
  - Number of casts (profiles) in memory

---

*Example:* (user input in bold, command used to modify parameter in parentheses at right)

```
s>getsd
<StatusData DeviceType='SBE25plus' SerialNumber='0250003'>
   <DateTime>2012-01-20T10:25:41</DateTime>                              [SetDateTime=]
   <EventSummary numEvents='5'/>                                         [can clear with ResetEC]
   <Power>
      <vBattery>14.8</vBattery>                                          (main battery pack voltage)
      <vExternal>0.7</vExternal>                                         (external power input voltage)
      <vLithium>0.5</vLithium>                                           (back-up lithium battery voltage)
      <vRTC>0.7</vRTC>                                                   (real-time clock battery voltage)
      <iExt>0.0</iExt>                                                   (auxiliary sensor current)
      <iSys>15.0</iSys>                                                  (T + C + P + pump current)
      <iCPU>17.8</iCPU>                                                  (processor current)
      </Power>
   <MemorySummary>
      <Bytes> 262144 </Bytes>                                           [can clear with InitLogging]
      <BytesFree> 1966538752 </BytesFree>                               [can clear with InitLogging]
      <Samples>3640</Samples>                                           [can clear with InitLogging]
      <SamplesFree>27313038</SamplesFree>                               [can clear with InitLogging]
      <Files>1</Files>                                                  [can clear with InitLogging]
      <CastFiles>1</CastFiles>                                          [can clear with InitLogging]
      </MemorySummary>
   </StatusData>
```

**Status** Commands (*continued*)

| | |
|---|---|
| **GetCC** | Get and display pressure sensor calibration coefficients, which are initially factory-set and should agree with Calibration Certificates shipped with 25*plus*. |

```
Example: (user input in bold, command used to modify coefficient in parentheses at right)
s>getcc
<CalibrationCoefficients DeviceType='SBE25plus' SerialNumber='0250003'>
   <Calibration format='STRAIN0' id='Main Pressure'>
      <SerialNum>00000007</SerialNum>
      <Pcaldate>2000-01-01</Pcaldate>                          [SetPCalDate=]
      <PA0>4.032810e-02</PA0>                                       [SetPA0=]
      <PA1>1.336885e-04</PA1>                                       [SetPA1=]
      <PA2>-3.442034e-12</PA2>                                      [SetPA2=]
      <PTEMPA0>-6.530079e-01</PTEMPA0>                         [SetPTempA0=]
      <PTEMPA1>5.252739e+01</PTEMPA1>                             [PTempA1=]
      <PTEMPA2>-5.198816e-01</PTEMPA2>                            [PTempA2=]
      <PTCA0>5.251322e+05</PTCA0>                                [SetPTCA0=]
      <PTCA1>3.474745e+01</PTCA1>                                [SetPTCA1=]
      <PTCA2>-9.448004e-01</PTCA2>                               [SetPTCA2=]
      <PTCB0>2.499975e+01</PTCB0>                                [SetPTCB0=]
      <PTCB1>-6.500000e-04</PTCB1>                               [SetPTCB1=]
      <PTCB2>0.000000e+00</PTCB2>                                [SetPTCB2=]
      <Poffset>0.000000e+00</Poffset>                  [SetPOffset= (decibars)]
      <Pfullscale>0.000000e+00</Pfullscale>     [SetPFullScale= (psia); factory set]
      </Calibration>
   </CalibrationCoefficients>
```

**Status** Commands (*continued*)

**GetEC**                    Get and display event counter data, which can help to identify root cause of a malfunction. Event counter records number of occurrences of common timeouts, power-on resets, etc. Clear with **ResetEC**. Possible events include:

- Power On Reset - 25*plus* successfully powered up and completed self-checks.
- Console Buffer Overflow - Receive buffer was full and more text was attempted to be added; these characters were lost.
- Ser1 Buffer Overflow - Receive buffer was full and more text was attempted to be added; these characters were lost. Could indicate corruption (loss) of recorded serial data.
- Ser2 Buffer Overflow – See Ser1 Buffer Overflow above.
- Console Suppress - a command has been interpreted (termination character/prompt found), and before it could be executed more characters arrived; these characters are lost. Commands sent too quickly will likely be corrupted or ignored.
- Ser1 Suppress - Termination character and/or prompt was found and before string could be stored (once every 62.5mS) more characters arrived; these characters are lost. Could indicate corruption (loss) of recorded serial data.
- Ser2 Suppress – See Ser1 Suppress above.
- Ser1 Failout - While waiting for a response from serial instrument, Failout time was exceeded. Serial instrument could have become unresponsive during this time or termination character and/or prompt could have been set incorrectly, leading serial processor to misinterpret response.
- Ser2 Failout – See Ser1Failout above.
- 1 - Overflow has occurred on UART0 (console); these characters have been lost.
- 2 - Overflow has occurred on UART1 (Ser1); these characters have been lost
- 3 - Overflow has occurred on UART2 (Ser2); these characters have been lost.
- 4 - Overflow has occurred on UART3 (power monitor); these characters have been lost.

- 5 - Parity/framing/RX FIFO/break error on UART0 (console); these characters have been lost. Confirm attached serial device is set to 8 bits, 1 stop bit no parity.
- 6 - Parity/framing/RX FIFO/break error on UART1 (ser1); these characters have been lost. Confirm attached serial device is set to 8 bits, 1 stop bit no parity.
- 7 - Parity/framing/RX FIFO/break error on UART2 (Ser2); these characters have been lost. Confirm attached serial device is set to 8 bits, 1 stop bit no parity.
- 8 - Parity/framing/RX FIFO/break error on UART3 (power monitor) - Power supply supervisor is in an undefined state.
- 12 - Receive buffer is full, but more text was attempted to be added; these characters are lost. Power supply supervisor is in an undefined state
- 16 - Termination character/prompt has been found, but before it could be stored more characters arrived; these characters are lost. Power supply supervisor is in an undefined state.
- 17 - Real-time clock error. Real time clock is in undefined state when trying to set alarm interval.
- 18 - Real-time clock error. Real time clock is in undefined state when trying to set alarm interval.
- 22 - Event number error. Undefined event has been added to event counter.

---

*Example:* (user input in bold, command used to modify parameter in parentheses)

**getec**
```
<EventCounters DeviceType = 'SBE25plus' SerialNumber = '0250003'>
   <EventSummary numEvents = '1'/>                                  [can clear with ResetEC]
   <Event type='PowerOnReset' count='1'/>
</EventCounters>
```

---

**ResetEC**            Delete all events in event counter (number of events displays in **GetSD** response, and event details display in **GetEC** response).

**Status** Commands (*continued*)

**GetHD**

> **Note:**
> External sensor types and serial numbers can be changed in the field, to accommodate changes in auxiliary sensors cabled to the 25*plus*. Sensor types do not affect the 25*plus* operation, and are simply informational. Other hardware data is typically not changed by the user.

Get and display hardware data, which is fixed data describing 25*plus*:

- Device type, Serial number
- Manufacturer
- Firmware version
- Firmware date
- Command set version
- PCB serial numbers and assembly numbers
- Manufacture date
- T, C, and P sensor types and serial numbers
- External voltage sensor types (DO, fluorometer, etc.) and serial numbers
- External serial sensor types (SBE 38, WET Labs, etc.) and serial numbers

**Status** Commands (*continued*)

**Status** Commands (*continued*)

```
Example: (user input in bold, command used to modify parameter in parentheses)
s>gethd
<HardwareData DeviceType='SBE25plus' SerialNumber='0250003'>
   <Manufacturer>Sea-Bird Electronics, Inc</Manufacturer>
   <FirmwareVersion>1.0</FirmwareVersion>
   <FirmwareDate>Mar  6 2012 13:18:08</FirmwareDate>
   <CommandSetVersion>1.0</CommandSetVersion>
   <PCBAssembly PCBSerialNum='040086' AssemblyNum='41760A'/>
   <PCBAssembly PCBSerialNum='039835' AssemblyNum='41761A'/>
   <MfgDate>2012-02-28T10:00:00</MfgDate>
   <InternalSensors>
      <Sensor id = 'Main Temperature'>
         <SerialNum>56789</SerialNum>
      </Sensor>
      <Sensor id = 'Main Conductivity'>
         <SerialNum>12345</SerialNum>
      </Sensor>
      <Sensor id = 'Main Pressure'>
         <SerialNum>34567</SerialNum>
         <Type>strain0</Type>
      </Sensor>
   </InternalSensors>
   <ExternalSensors>
      <Sensor id = 'volt 0'>
         <type>not assigned</type>                          [SetVoltType0=]
         <SerialNum>not assigned</SerialNum>                [SetVoltSN0=]
      </Sensor>
      <Sensor id = 'volt 1'>
         <type>not assigned</type>                          [SetVoltType1=]
         <SerialNum>not assigned</SerialNum>                [SetVoltSN1=]
      </Sensor>
      <Sensor id = 'volt 2'>
         <type>not assigned</type>                          [SetVoltType2=]
         <SerialNum>not assigned</SerialNum>                [SetVoltSN2=]
      </Sensor>
      <Sensor id = 'volt 3'>
         <type>not assigned</type>                          [SetVoltType3=]
         <SerialNum>not assigned</SerialNum>                [SetVoltSN3=]
      </Sensor>
      <Sensor id = 'volt 4'>
         <type>not assigned</type>                          [SetVoltType4=]
         <SerialNum>not assigned</SerialNum>                [SetVoltSN4=]
      </Sensor>
      <Sensor id = 'volt 5'>
         <type>not assigned</type>                          [SetVoltType5=]
         <SerialNum>hi there</SerialNum>                    [SetVoltSN5=]
      </Sensor>
      <Sensor id = 'volt 6'>
         <type>not assigned</type>                          [SetVoltType6=]
         <SerialNum>hi there</SerialNum>                    [SetVoltSN6=]
      </Sensor>
      <Sensor id = 'volt 6'>
         <type>not assigned</type>                          [SetVoltType7=]
         <SerialNum>hi there</SerialNum>                    [SetVoltSN7=]
      </Sensor>
      <Sensor id = 'serial 1'>
         <type>not assigned</type>                          [SetSerialTypeSer1=]
         <SerialNum> not assigned </SerialNum>              [SetSerialSNSer1=]
      </Sensor>
      <Sensor id = 'serial 2'>
         <type>not assigned</type>                          [SetSerialTypeSer2=]
         <SerialNum> not assigned </SerialNum>              [SetSerialSNSer2=]
      </Sensor>
   </ExternalSensors>
</HardwareData>
```

**InitHD**                    Reset all user-programmable parameters that appear in **GetHD** response to factory defaults (*Not Set*). **Requires confirmation**.

---

**Status** Commands (*continued*)

> **Note:**
> If there are a large number of files in memory, the 25*plus* can take up to approximately 45 seconds to complete the **GetFiles** response. Click the Esc key to abort the response.

**GetFiles**

Get and display cast (.xml) and serial sensor (.txt) summary files names from 25*plus*, listing all files in memory. There can be up to two files types in memory:

- .xml files contain T, C, P, and auxiliary voltage sensor data; they also contain auxiliary serial sensor data if **SetInlineSer#=Y**.
- .txt files contain auxiliary serial sensor data if **SetInlineSer#=N**.

---

*Example:* (user input in bold, command used to modify parameter in parentheses)
```
s>getfiles
<FileData DeviceType='SBE25plus' SerialNumber='0250003'>
   <files>
      <casts date='2012-01-11'>
         <file index='0' name='2012-01-11T170131 SBE250250003.xml' size='81'/>
         </casts>
      <casts date='2012-01-17'>
         <file index='1' name='2012-01-17T133743 SBE250250003.xml' size='6463'/>
         </casts>
      <casts date='2012-01-19'>
         <file index='2' name='2012-01-19T114803 SBE250250003.xml' size='2528156'/>
         <file index='3' name='2012-01-19T120252 SBE250250003.xml' size='7094901'/>
         </casts>
      </files>
   </FileData>
```

---

> **Note:**
> A fault occurs when an attached sensor draws more than ~1.25A current; the fault cuts the channel off. To clear the fault, disable the channel using **one** of the following methods:
> - If logging, pull the plunger switch out to the Off position or send **Stop**.
>   **or**
> - Send **SetVAuxPower#=N** (where # = 0, 1, 2, 3).
>
> Fault checking restarts when the channel is re-enabled (send **SetVAuxPower#=Y**, push the plunger switch in to the On position, or send **Start**).
> Fault states are stored with the measurement in the status field.

**GetFault**

Get and display fault status of all auxiliary sensor channels from 25*plus*. If a channel is in fault, power cycle it to remove fault. If a channel is in fault at start of a measurement, that channel is automatically powered down for 0.625 sec or **SetVAuxDelay#=** (# = 0, 1, 2, or 3, as applicable), whichever is larger, in an attempt to clear fault.

---

*Example:* (user input in bold, command used to modify parameter in parentheses)
```
s>getfault
<VoltageChannelFaults>
     <vaux0>no</vaux0>                            (applies to voltage channels 0 and 1)
     <vaux1>no</vaux1>                            (applies to voltage channels 2 and 3)
     <vaux2>no</vaux2>                            (applies to voltage channels 4 and 5)
     <vaux3>yes</vaux3>          (applies to voltage channels 6 and 7; and serial channels 1 and 2)
      </VoltageChannelFaults>
</Executed>
```

**General Setup** Commands

**SetDateTime=**yyyy-mm-ddThh:mm:ss

Set 25*plus* real-time clock **UTC** date and time, *where*
yyyy = year,
mm = month (01, 02, etc.),
dd = day of month (01, 02, etc.),
hh = hour (0-24),
mm = minute, ss = second

---

*Example* (user input in bold): Set date and time to 05 March 2012 14:35:20.
```
S>setdatetime=2012-03-05T14:35:20
```

---

**SetBaudConsole=x**

x= baud for communication with 25*plus* (600, 1200, 2400, 4800, 9600, 14400, 19200, 38400, 57600, and 115200). *Default 9600*. Check capability of your computer and terminal program before increasing baud; high baud requires a short cable and good PC serial port with accurate clock. **Must turn off executed tags before sending command. Must be sent twice to change baud**.

---

*Example* (user input in bold): While communicating at 9600 baud, set baud for communications between 25*plus* and computer to 38400.
```
s>setexecutedtag=n
S>setbaudconsole=38400
<!—repeat command to execute-->        (still communicating at 9600 baud)
S>setbaudconsole=38400                 (now displaying garbage characters)
```
In Seaterm232's Communications menu, select Configure. In Configure dialog box, select new baud (38400) and click OK. In Communications menu, select Disconnect and reconnect. 25*plus* connects at new baud. If desired, turn executed tags back on (**SetExecutedTag=Y**).

---

**SetEchoConsole=x**

x=Y (*default*): Echo characters received from computer; monitor will show entered commands as you type.

x=N: Do not.

**SetExecutedTag=x**

x=Y (*default*): Display XML Executing and Executed tags. Executed tag displays at end of each command response; Executing tag displays 1 or more times if 25*plus* response to a command requires additional time.

x=N: Do not.

---

*Example:* Set 25*plus* to output Executed and Executing tags (user input in bold).
```
s>setexecutedtag=y
<Executed/>
s>getcd
. . .            (GetCD response)
<Executed/>
```

---

**QS**

Quit session and place 25*plus* in quiescent (sleep) state. Main power turned off. Memory retention not affected.

**Pump Setup** Commands

**Notes:**
- See *Pump Operation.*
- The conductivity sensor output frequency must exceed **SetMinCondFreq=** for 8 consecutive measurements to turn on the pump. Similarly, when the frequency falls below **SetMinCondFreq=** for 8 consecutive measurements, the pump turns off.

**SetMinCondFreq=x**          **x**= minimum conductivity sensor output frequency (Hz) to enable pump turn-on, to prevent pump from turning on before 25*plus* is in water. Pump stops when conductivity frequency drops below **SetMinCondFreq=**. 25*plus* Configuration Sheet lists uncorrected (raw) frequency output at 0 conductivity. Typical value for salt water and estuarine application is: (0 conductivity frequency + 500 Hz). Typical value for fresh water applications is: (0 conductivity frequency + 5 Hz). ***Default 3000.***

**SetPumpDelay=x**          **x**= time (sec) to wait after **SetMinCondFreq=** is reached before turning pump on. Pump starts **SetPumpDelay=** sec after conductivity cell's frequency output is greater than **SetMinCondFreq=**, allowing time for Tygon tubing and pump to fill with water after 25*plus* is submerged. Range 0 – 600 sec; ***default 60 sec.***

**Voltage Sensor Setup** Commands

These commands set the delay for supply of power to auxiliary sensor channels. The delay is from the start of logging (initiated with the plunger switch or **StartNow**), and allows the customer to stagger powering up of the auxiliary sensors, so that the total inrush current does not exceed the maximum current rating (1.2 Amps).

**Notes:**
- **Auxiliary sensors that draw more than 1 Amp even momentarily must be cabled to J3 (voltage output sensors) or J6 (serial output sensors). See *Appendix V: Integration of Auxiliary Sensors with High Current Draw* for jumper details.**
- The 25*plus* measures and stores the output from all auxiliary voltage sensor channels and all enabled auxiliary serial sensor channels, even if they are not powered up yet. The data from measurements made before power is applied is not valid.
- The power-on delay for any auxiliary **serial** sensors is also set by **SetVAuxDelay3=**.

**SetVAuxDelay0=x**          **x=** Delay (integer seconds) for supply of power to **voltage channels 0 and 1** (**J0** end cap connector). Range 0 – 600 sec; *default 0*.

**SetVAuxDelay1=x**          **x=** Delay (integer seconds) for supply of power to **voltage channels 2 and 3** (**J1** end cap connector). Range 0 – 600 sec; *default 0.*

**SetVAuxDelay2=x**          **x=** Delay (integer seconds) for supply of power to **voltage channels 4 and 5** (**J2** end cap connector). Range 0 – 600 sec; *default 0*.

**SetVAuxDelay3=x**          **x=** Delay (integer seconds) for supply of power to **voltage channels 6 and 7** (**J3** end cap connector) and to **serial channels 1 and 2** (**J6** end cap connector). Range 0 – 600 sec; *default 0*.

*Example:* Set voltage channels 2 and 3 (end cap connector J1) to power up 10 seconds after logging is started (user input in bold).
s>**SETVAUXDELAY1=10**

**Serial Sensor Setup** Commands

**Notes:**
- **The 25*plus* cannot transmit real-time data from a serial output sensor.**
- See *Appendix IV: Integration of Serial Sensors* for a detailed description of the interface and specific setup guidelines for use with a WET Labs Triplet, WET Labs SeaOWL UV-A, or Satlantic SUNA V2.
- For integration of auxiliary sensors that draw more than 1 Amp even momentarily, see *Appendix V: Integration of Auxiliary Sensors with High Current Draw* for jumper details.
- You cannot use the same name for both serial port channels.

There are two channels for auxiliary serial (RS-232) sensors: 1 and 2. These are designated as **#** in the commands below (for example, **SetEnableSer1=Y** enables acquisition of data from serial channel 1; **SetEnableSer2=Y** enables acquisition of data from serial channel 2).

| | |
|---|---|
| **InitSer#** | Reset all serial sensor setup parameters to factory defaults. **Requires confirmation**. |
| **SetEnableSer#=x** | **x=Y:** Enable measurement of serial sensor channel. |
| | **x=N (default):** Disable measurement of serial sensor channel. |
| **SetNameSer#=x** | **x=** string (0 - 10 characters) to describe sensor on serial channel, useful for helping to identify sensor to user and for naming output data file for serial sensor. *Default #* (i.e., default for serial channel 1 is **SetNameSer1=Ser1**; default for serial channel 2 is **SetNameSer2=Ser2**). Name becomes part of .txt file name (if **SetInlineSer#=N**), after date and time that cast started. |

*Example* (user input in bold): Set name for serial sensor 1 to sbe38, so that output data file will be named *<datetime> sbe38.txt*.
```
S>setnameser1=sbe38
```

| | |
|---|---|
| **SetBaudSer#=x** | **x=** baud for communication between 25*plus* and serial sensor (600, 1200, 2400, 4800, 9600, 14400, 19200, and 38400). *Default 9600*. |

*Example* (user input in bold): Set baud for communications between 25*plus* and serial sensor 1 to 38400.
```
S>setbaudser1=38400
<Executed/>
```

**Note:**
The 25*plus* configuration (.xmlcon) file must match this selection of whether the serial sensor data is stored in-line with other data or in a separate .txt file when processing data uploaded from memory in SBE Data Processing. View and edit the configuration file in SBE Data Processing.

| | |
|---|---|
| **SetInlineSer#=x** | **x=Y (*default*):** Store serial sensor data in memory in-line with all other data (T, C, P, and auxiliary sensor voltage channels) in .xml file. |
| | **x=N:** Store serial sensor data in memory in a separate file (.txt file) from all other data. |
| **SetPromptSer#=x** | **x=** prompt (0 - 10 characters) that **serial sensor sends to 25*plus*** to indicate that it is awake and ready to receive commands. *Default S>*. Leave blank (if no prompt **SetPromptSer#=**; hit Enter key after equal sign). |

**Serial Sensor Setup** Commands (*continued*)

| | | |
|---|---|---|
| | **SetMeasStringSer#=x** | **x**= command string (0-32 characters) that 25*plus* sends to serial sensor to command it to take **1** sample when 25*plus* is logging. 25*plus* sends **SetMeasStringSer#=** every **SetMeasIntervalSer#=** seconds. If blank (**SetMeasStringSer#=**; hit Enter key after equal sign), no command will be sent to serial sensor; this is appropriate for a serial sensor that can be commanded to sample continuously (see **SetStartStringSer#=** below) or that automatically takes samples and sends data when power is applied. *Default TS.* |
| | **SetMeasIntervalSer#=x** | **x**= interval (sec) between each time 25*plus* sends command string defined by **SetMeasStringSer#=** to serial sensor when 25*plus* is logging. If **SetMeasIntervalSer#=0**, 25*plus* sends command string defined by **SetMeasStringSer#=** as soon as it receives reply to previous request for data. Range 0-600 sec. *Default 0.* |
| | **SetStartStringSer#=x** | **x**= command string (0-32 characters) 25*plus* sends to serial sensor to command it to sample **autonomously** when 25*plus* is logging. If blank (**SetStartStringSer#=**; hit Enter key after equal sign), no command will be sent to serial sensor; this is appropriate for a serial sensor that must be commanded to sample individually for each sample (see **SetMeasStringSer#=** above) or that automatically takes samples and sends data when power is applied. *Default Start.* |
| **Note:**<br>See *Appendix VI: Character Map and Values* for the list of termination characters. | **SetExecCharSer#=x** | **x**= decimal value (0-255) of command execution character that **25*plus* adds to end of command string** defined by **SetMeasStringSer#=** and **SetStartStringSer#=**. 254 is carriage return line feed ('\r\n'); 255 is no execution character. *Default 254*. |

---

*Example 1* (user input in bold): Sensor on serial port channel 1 can be commanded to **take 1 sample** with the following command: TakeSample. You want it to sample every 2 seconds. Sensor uses a carriage return line feed command string execution character ('\r\n' = 254).
```
s>setmeasstringser1=takesample
s>setmeasintervalser1=2
s>setstartstringser1=
s>setexeccharser1=254
```

*Example 2* (user input in bold): Sensor on serial port channel 1 can be commanded to **sample autonomously** with the following command: Start. Sensor uses a carriage return line feed command string execution character ('\r\n' = 254).
```
s>setmeasstringser1=
s>setstartstringser1=start
s>setexeccharser1=254
```

*Example 3* (user input in bold): Sensor on serial port channel 1 can start sampling automatically when power is applied.
```
s>setmeasstringser1=
s>setstartstringser1=
```

---

**Serial Sensor Setup** Commands *(continued)*

| | | |
|---|---|---|
| **Note:**<br>See *Appendix VI: Character Map and Values* for the list of termination characters. | **SetTermCharSer#=x** | **x=** decimal value (0-255) of termination character that **serial sensor sends to 25***plus* at end of its reply to a command. Once 25*plus* receives a termination character, it ignores any additional characters received from sensor. 254 is carriage return line feed ('\r\n'); 255 is no termination character. If 25*plus* does not receive termination character, it stops waiting after **SetFailoutSer#=**. *Default 254*. |
| | **SetSuppressSer#=x** | **x=Y (default)**: Termination character (**SetTermCharSer#=**) and prompt (**SetPromptSer#=**) sent by serial sensor to 25*plus* are suppressed, so they do not appear in data file.<br><br>**x=N**: Termination character and prompt sent by serial sensor to 25*plus* are included in data file. |
| | **SetTimeoutSer#=x** | **x=** maximum time (seconds) that 25*plus* will wait for prompt (**SetPromptSer#=**) from serial sensor, indicating that serial sensor has completed processing a command that was sent. Range 1-600 sec; *default 3 sec*. |
| | **SetFailoutSer#=x** | **x=** maximum time (seconds) that 25*plus* will wait for **data** from serial sensor. Range 1 - 600 sec; *default 60 sec*. |
| | **ToSer#=x** | **x=** command string (1-64 characters) to send to serial sensor. Can be any command recognized by serial sensor. |

**Real-Time Output Setup** Commands

**Notes:**
- See *Data Formats*.
- **The 25*plus* cannot transmit real-time data from a serial output sensor.**
- If calculating elapsed time in Seasave (real-time data acquisition program), Seasave uses the real time data output rate in the 25*plus* configuration (.xmlcon) file. The configuration file must match the real-time data output rate in the 25*plus*. View and edit the configuration file in Seasave or SBE Data Processing.
- If calculating elapsed time in SBE Data Processing (post-processing software), SBE Data Processing uses the information in the file header to determine the sample rate:
  - ➢ .hex file saved by Seasave during real-time data acquisition - SBE Data Processing uses the real time data output rate in the header to calculate elapsed time.
  - ➢ File uploaded from 25*plus* memory - SBE Data Processing uses the internal data storage rate in the header (16 scans/sec) to calculate elapsed time.

The 25*plus* always takes measurements and stores data in memory at 16 Hz, but the real-time output rate varies. The rate at which data can be transmitted is dependent on the baud rate and the number of auxiliary voltage sensors being transmitted in real-time. **The maximum real-time output rate is 8 Hz (8 scans/sec).** See *Real-Time Setup* for baud rate, voltage sensor output, and real-time output rate combinations.

**SetOutputFormat=x**

**x=0 (default):** Format required for use **with Seasave real-time data acquisition software**. When logging data, output all T, C, P data and selected auxiliary voltage sensor data in real-time, at baud rate set by **SetBaudConsole**=. See **SetVOut#=** commands below for selection of which voltage sensor channels to output in real-time. 25*plus* varies real-time output rate automatically, depending on baud rate and number of voltage sensors enabled for real-time output. Maximum real-time output rate is 8 Hz (8 scans/sec).

**x=1:** Format required for use with **autonomous water sampling systems (SBE 32 Carousel Water Sampler with AFM, or SBE 55 ECO Water Sampler).** When logging data, **output converted pressure and scan number** at a baud rate of 9600, at 1 Hz, regardless of baud rate set by **SetBaudConsole**=.
When you stop logging (by sending **Stop** or Esc), 25*plus* returns to baud rate set by **SetBaudConsole**=.

**SetHistoricRate=x**

Applicable only if 25*plus* baud rate is 4800 (**SetBaudConsole=4800**) and **SetOutputFormat=0.** A baud of 4800 is required when 25*plus* used with SBE 33 or SBE 36 Deck Unit; this command sets up 25*plus* for operation with one of these Deck Units. A slower transmission rate is required if Deck Unit is appending NMEA and Surface PAR data, due to Deck Unit limitations.

**x=0:** Slower. Output real-time data at 4 Hz if no external voltage sensors are enabled for real-time output; output real-time data at 2 Hz if 1 or more external voltage sensors are enabled for real-time output. **Use this setting if SBE 33 or 36 Deck Unit is appending NMEA and Surface PAR data.**

**x=1:** Faster. Output real-time data at 8 Hz if no external voltage sensors are enabled for real-time output; output real-time data at 4 Hz if 1 or more external voltage sensors are enabled for real-time output.

---

**Real-Time Output Setup** Commands *(continued)*

---

| | |
|---|---|
| **Notes:** | |

**Notes:**
- External voltage numbers 0, 1, 2, 3, 4, 5, 6, and 7 correspond to wiring of sensors to a voltage channel on the 25*plus* end cap (see *Dimensions and End Cap Connectors* in *Section 2: Description of SBE 25plus*). The 25*plus* always **stores** data from all 8 voltage channels in memory, regardless of the selections for **SetVOut0=** through **SetVOut7=**, and regardless of whether a sensor is present on each channel.
- The 25*plus* configuration (.xmlcon) file must match this selection of external voltages to be included with real-time data when viewing real-time data in Seasave and processing real-time data in SBE Data Processing. View and edit the configuration file in Seasave or SBE Data Processing.

**SetVOut0=x**

**x=Y:** If **SetOutputFormat=0**, output data from voltage **channel 0** with real-time data.

**x=N:** Do not.

**SetVOut1=x**

**x=Y:** If **SetOutputFormat=0**, output data from voltage **channel 1** with real-time data.

**x=N:** Do not.

**SetVOut2=x**

**x=Y:** If **SetOutputFormat=0**, output data from voltage **channel 2** with real-time data.

**x=N:** Do not.

**SetVOut3=x**

**x=Y:** If **SetOutputFormat=0**, output data from voltage **channel 3** with real-time data.

**x=N:** Do not.

**SetVOut4=x**

**x=Y:** If **SetOutputFormat=0**, output data from voltage **channel 4** with real-time data.

**x=N:** Do not.

**SetVOut5=x**

**x=Y:** If **SetOutputFormat=0**, output data from voltage **channel 5** with real-time data.

**x=N:** Do not.

**SetVOut6=x**

**x=Y:** If **SetOutputFormat=0**, output data from voltage **channel 6** with real-time data.

**x=N:** Do not.

**SetVOut7=x**

**x=Y:** If **SetOutputFormat=0**, output data from voltage **channel 7** with real-time data.

**x=N:** Do not.

**Logging** Commands

Logging commands direct the SBE 25*plus* to sample data. When commanded to start logging, the 25*plus* takes samples, stores the data in its memory, and transmits real-time data. There are two ways to start logging:

- **Using the plunger switch to start and stop logging**: Logging is started by pushing the switch in to the On position.
- **Using commands to start logging**: Logging is started by sending **StartNow** through a terminal program.

The first time logging is started after logging was initialized (**InitLogging or DeleteAll**), data recording starts at the beginning of memory and any previously recorded data is written over. When the switch is pulled out to the Off position or the **Stop** command is sent, recording stops. Each time logging is restarted, recording continues, with new data stored in a new directory. The data file contains header information, indicating the incremented cast number, date, time, and sample numbers contained in the cast.

---

**Notes:**
- In Seaterm232, to save real-time data to a file, click the Capture menu before beginning logging.
- If the FLASH memory is filled to capacity, data sampling and transmission of real-time data continue, but excess data is not saved in memory. **The 25*plus* will not overwrite data stored in memory.**
- If the SBE 25*plus* is logging data and the internal battery voltage is less than 9.5 volts for 20 seconds **or** the external power is less than 14 volts for 20 seconds, it halts logging and displays <Powerfail/>.

---

**Notes:**
- To stop logging:
  - Pull plunger switch out to the Off position, or
  - Send **Stop**, or
  - Click Esc on the keyboard.
- You may need to send **Stop** several times to get the 25*plus* to respond.
- You must stop logging before uploading data.

---

**StartNow**          Start logging.


**Stop**          Stop logging.

**Notes:**

- **Use the Upload menu to upload data that will be processed by SBE Data Processing**. Manually entering a data upload command does not produce data with the required header information for processing by our software. These commands are included here for reference for users who are writing their own software.
- **If not using the Upload menu in Seaterm232** - To save data to a file, click the Capture menu before entering a data upload command.
- All file index numbers and file sizes in memory can be viewed with **GetFiles**.

**Data Upload and Memory Reset** Commands

**Stop logging before uploading data.**

| | |
|---|---|
| **SetFile=x** | **x**= file index number (0 – 400) for file that you want to upload or delete (see **UploadData=** and **DeleteFile**). **SetFile=** is cleared any time 1 file is deleted, all files are deleted, or a new cast is added to memory (logging is started and stopped). If **x** is a large number, 25*plus* can take up to approximately 45 seconds to process command, because it must go through files in memory sequentially to assign appropriate file. To abort, click Esc key. |
| **GetFile** | Get and display file index number and name for file selected with **SetFile=**. |
| **UploadData=x,y** | Upload file selected with **SetFile=**, starting at character **x** (characters in each file are numbered starting with 0) and uploading a total of **y** characters. Range for both **x** and **y** is 0 to $(2^{32} – 1)$. Interrupt and cancel request by clicking Esc. |

*Example:* 25*plus* contains files from 4 casts, with index numbers 0, 1, 2, and 3. You want to upload all of the data in cast 3. **GetFiles** output indicates that cast 3 is *size='81'*. Send **SetFile=3**, and then send **UploadData=0,81** (start uploading at character 0 and upload a total of 81 characters).

| | |
|---|---|
| **DeleteFile** | Delete file selected with **SetFile=**. **Requires confirmation.** 25*plus* updates all file index numbers larger than this one, after file is deleted. 25*plus* nulls out **SetFile=** selection after deleting file; you must reset **SetFile=** to perform any additional upload or delete operations. |

*Example:* 25*plus* contains files from 4 casts, with index numbers 0, 1, 2, and 3. You send **SetFile=2**, and then send **DeleteFile**. This deletes the file with index number 2. Files 2 and 3 are then automatically renumbered as 1 and 2, so the index numbers are now 0, 1, and 2. If you want to upload or delete the new file 2, you need to resend **SetFile=2** before sending **UploadData=** or **DeleteFile**.

| | |
|---|---|
| **DeleteAll** | Initialize logging - after all previous data has been uploaded, make entire memory available for recording. **DeleteAll** sets index number (cast number) to 0 (sampling will start with cast 0). **Do not send DeleteAll until all existing data has been uploaded. Requires confirmation.** |
| **InitLogging** | Equivalent to **DeleteAll. Requires confirmation.** |

**Testing** Commands

**TS**                    Apply power to all sensors and channels, take **1 measurement**, output data in Hex (not including serial sensor data), and remove power.

---

*Example (user input in bold):*

s>**ts**

00000000000400070005000000050003000600060075 99B0008053B34597F32B45E135FE

72 Hex characters displayed:

iiiiiiiivvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv00pppppp00ppppppcccccccctttttttt

iiiiiiii = diagnostic information
vvvv = external voltage sensor 7 voltage
vvvv = external voltage sensor 6 voltage
vvvv = external voltage sensor 5 voltage
vvvv = external voltage sensor 4 voltage
vvvv = external voltage sensor 3 voltage
vvvv = external voltage sensor 2 voltage
vvvv = external voltage sensor 1 voltage
vvvv = external voltage sensor 0 voltage
00pppppp = pressure temperature counts
00pppppp = pressure sensor counts
cccccccc = conductivity (Hz)
tttttttt = temperature (Hz)

---

**TSC**                    Continuously measure and output conductivity frequency (Hz). Click Esc key to stop and remove power.

**TST**                    Continuously measure and output temperature frequency (Hz). Click Esc key to stop and remove power.

**TSPR**                    Continuously measure and output raw pressure (counts) followed by raw pressure temperature (counts). Click Esc key to stop and remove power.

**TSP**                    Continuously measure and output pressure (decibars). Click Esc key to stop and remove power.

**TSVR**                    Apply power to all auxiliary voltage channels (0 – 7), and continuously measure and output raw counts. Click Esc key to stop and remove power.

**TSV**                    Apply power to all auxiliary voltage channels (0 – 7), and continuously measure and output voltages. Click Esc key to stop and remove power.

**Testing** Commands *(continued)*

| | |
|---|---|
| **GetVAux0** | Apply power to **voltage channels 0 and 1** (**J0** end cap connector), and continuously measure and output voltages. Click Esc key to stop and remove power. |
| **GetVAux1** | Apply power to **voltage channels 2 and 3** (**J1** end cap connector), and continuously measure and output voltages. Click Esc key to stop and remove power. |
| **GetVAux2** | Apply power to **voltage channels 4 and 5** (**J2** end cap connector), and continuously measure and output voltages. Click Esc key to stop and remove power. |
| **GetVAux3** | Apply power to **voltage channels 6 and 7** (**J3** end cap connector) and **serial channels 1 and 2** (**J6** end cap connector), and continuously measure and output voltages (does not output serial data). Click Esc key to stop and remove power. |

---

*Example:* Apply power to connector J1 (voltage channels 2 and 3) and measure its output (user input in bold).

```
S>GETVAUX1
1.0035,  0.0001                    (voltage outputs from channels 2 and 3)
1.0034,  0.0002
(Press Esc key to stop and remove power)
```

---

| | |
|---|---|
| **SetVAuxPower0=x** | **x=Y:** Apply power to **voltage channels 0 and 1** (**J0** end cap connector) for testing.<br><br>**x=N:** Remove power. |
| **SetVAuxPower1=x** | **x=Y:** Apply power to **voltage channels 2 and 3** (**J1** end cap connector) for testing.<br><br>**x=N:** Remove power. |
| **SetVAuxPower2=x** | **x=Y:** Apply power to **voltage channels 4 and 5** (**J2** end cap connector) for testing.<br><br>**x=N:** Remove power. |
| **SetVAuxPower3=x** | **x=Y:** Apply power to **voltage channels 6 and 7** (**J3** end cap connector) and to **serial channels 1 and 2** (**J6** end cap connector) for testing.<br><br>**x=N:** Remove power. |

**Note:**
Any auxiliary **serial** sensors are also powered on by **SetVAuxPower3=**.

## Calibration Coefficients Commands

**Notes:**
- F = floating point number
  S = string with no spaces
- Temperature and conductivity sensor calibration coefficients are not stored in the 25*plus* EEPROM, since these sensors are modular and can be switched out in the field. These calibration coefficients are stored in the 25*plus* configuration (.xmlcon) file. View and/or modify the calibration coefficients using the Configure menu in SBE Data Processing or the Configure Inputs menu in Seasave.
- Auxiliary sensor calibration coefficients are not stored in the 25*plus* EEPROM.
  - Calibration coefficients for sensors that output raw data are stored in the 25*plus* configuration (.xmlcon) file. View and/or modify the calibration coefficients using the Configure menu in SBE Data Processing or the Configure Inputs menu in Seasave.
  - Some RS-232 auxiliary sensors store their calibration coefficients internally, and output data in engineering units. For Sea-Bird auxiliary sensors such as the SBE 38, view and/or modify the instrument's calibration coefficients by connecting the instrument to the computer directly and using Seaterm, our older terminal program software (not SeatermV2).

Pressure sensor calibration coefficients are initially factory-set and should agree with Calibration Certificate shipped with the 25*plus*. These coefficients should match those stored in the 25*plus* configuration (.xmlcon) file. View and/or modify the coefficients in the .xmlcon file using the Configure menu in SBE Data Processing or the Configure Inputs menu in Seasave.

| Command | Description |
|---|---|
| **SetPCalDate=yyyy-mm-dd** | Set pressure sensor calibration UTC date and time, *where* <br> yyyy = year <br> mm = month (01, 02, etc.) <br> dd = day of month (01, 02, etc.) |
| **SetPA0=F** | **F**=pressure sensor A0. |
| **SetPA1=F** | **F**=pressure sensor A1. |
| **SetPA2=F** | **F**=pressure sensor A2. |
| **SetPTempA0=F** | **F**=pressure temperature A0. |
| **SetPTempA1=F** | **F**=pressure temperature A1. |
| **SetPTempA2=F** | **F**=pressure temperature A2. |
| **SetPTCA0=F** | **F**=pressure temperature compensation ptca0. |
| **SetPTCA1=F** | **F**=pressure temperature compensation ptca1. |
| **SetPTCA2=F** | **F**=pressure temperature compensation ptca2. |
| **SetPTCB0=F** | **F**=pressure temperature compensation ptcb0. |
| **SetPTCB1=F** | **F**=pressure temperature compensation ptcb1. |
| **SetPTCB2=F** | **F**=pressure temperature compensation ptcb2. |
| **SetPOffset=F** | **F**=pressure sensor offset correction (**decibars**). |
| **SetPFullScale=F** | **F**=pressure sensor full scale range (**psia**); factory set. |

**Hardware Configuration** Commands

Hardware Configuration commands are descriptive character strings, **up to 11 characters long**. The character string can include the following symbols:
- +
- -
- = (note: '=' cannot be the first character in the string).

*Temperature and Conductivity Sensor Settings - can be modified in the field to accommodate changes in T and C sensors cabled to the 25plus*
**SetTempSer=x**
**SetCondSer=x**

The remaining Hardware Configuration commands are used to set auxiliary channel sensor types (description such as *oxygen*, *fluorometer*, etc.) and serial number. Note that the sensor 'type' does not affect the SBE 25*plus* operation, and is simply informational.

*Auxiliary* **Voltage Sensor** *Settings – can be modified in the field to accommodate changes in auxiliary voltage sensors cabled to the 25plus*
**SetVoltType0=**
**SetVoltSN0=**
**SetVoltType1=**
**SetVoltSN1=**
**SetVoltType2=**
**SetVoltSN2=**
**SetVoltType3=**
**SetVoltSN3=**
**SetVoltType4=**
**SetVoltSN4=**
**SetVoltType5=**
**SetVoltSN5=**
**SetVoltType6=**
**SetVoltSN6=**
**SetVoltType7=**
**SetVoltSN7=**

*Auxiliary* **Serial Sensor** *Settings – can be modified in the field to accommodate changes in auxiliary serial sensors cabled to the 25plus*
**SetSerialTypeSer1=**
**SetSerialSNSer1=**
**SetSerialTypeSer2=**
**SetSerialSNSer2=**

# Data Format

## SetOutputFormat=0 (T, C, P, voltage sensors, serial sensors)

**Real-time data consists of temperature, conductivity, pressure, pressure temperature, and data from enabled auxiliary voltage sensors (SetVOut0= through SetVOut7=). The SBE 25*plus* cannot transmit data from serial output sensors or diagnostic data in real time. The serial sensor output and diagnostic data described below is obtained by uploading data from the 25*plus* memory.**

The data format is a combination of fixed byte size and markers. The fixed byte area corresponds to core measurements of the SBE 25*plus* and diagnostic output, while the variable area is allocated to the, possibly asynchronous, serial instrument(s). The areas are separated by a tab deliminator, and each scan is terminated by a carriage return line feed. An example is shown in hexadecimal format.
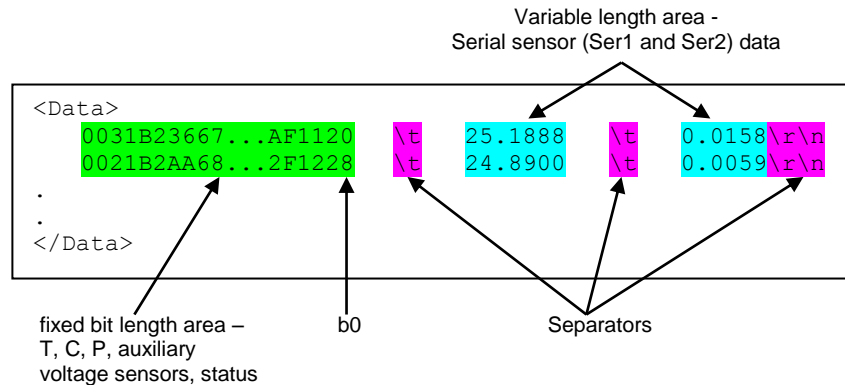


In the fixed length region, the SBE 25*plus* stores data from its core set of sensors: T, C, P, and eight voltage inputs, as well as some diagnostic information. 24b numbers are stored as 32b numbers with zero padding. The full string is represented by 36 bytes (72 hex characters). All auxiliary voltage channels are always recorded in memory, regardless of whether they are output in real-time.

| Location | Representation (hex) | Variable | Encoding |
|----------|---------------------|----------|----------|
| *b31-b0* | tttttttt | temperature (Hz) | IEEE 754 format floating point, 32 bit |
| *b63-b32* | cccccccc | conductivity (Hz) | IEEE 754 format floating point, 32 bit |
| *b95-b64* | 00pppppp | pressure | raw ADC counts * |
| *b127-b96* | 00pppppp | pressure temperature | raw ADC counts * |
| *b143-b128* | vvvv | voltage input 0 | $V_0 = (code / 2^{16}) * 5.000$ |
| *b159-b144* | vvvv | voltage input 1 | $V_1 = (code / 2^{16}) * 5.000$ |
| *b175-b160* | vvvv | voltage input 2 | $V_2 = (code / 2^{16}) * 5.000$ |
| *b191-b176* | vvvv | voltage input 3 | $V_3 = (code / 2^{16}) * 5.000$ |
| *b207-b192* | vvvv | voltage input 4 | $V_4 = (code / 2^{16}) * 5.000$ |
| *b223-b208* | vvvv | voltage input 5 | $V_5 = (code / 2^{16}) * 5.000$ |
| *b239-b224* | vvvv | voltage input 6 | $V_6 = (code / 2^{16}) * 5.000$ |
| *b255-b240* | vvvv | voltage input 7 | $V_7 = (code / 2^{16}) * 5.000$ |
| *b287-b256* | iiiiiiii | diagnostic | see below |

* For real-time output, pressure and pressure temperature are output without the zero padding (i.e., pppppp instead of **00**pppppp).

The IEEE 754 floating point numbers are decoded as follows:

| 31 | 30 | | 23 | 22 | | 0 |
|---|---|---|---|---|---|---|
| S | Exponent (E) | | | Mantissa (M) | | |

*Num = (-1) $^S$ * 2 $^{(E-127)}$ * 1. M*

All engineering units are computed in this variable type and as such conversion to fixed arithmetic is apt to introduce error.

For a tool to convert the IEEE 754 Hex output temperature and conductivity to frequencies, see www.h-schmidt.net/FloatConverter/IEEE754.html.

---

*Example*: example real-time scan = 459A00FE452010CD808B00628E36

- Temperature tttttttt = 459A00FE Hex IEEE 754 = 4928.124 Hz
- Conductivity cccccccc = 452010CD Hex IEEE 754 = 2561.05 Hz
- Pressure pppppp = 808B00 Hex = 8424192 counts
- Pressure temperature pppppp = 628E36 Hex = 6458934 counts;
  converting pressure temperature counts to a voltage
  V = 4.096 * (raw counts / 16777216) = 4.096 * (6458934 / 16777216)
  = 1.57 V

Use the equations on the calibration sheets for temperature, conductivity, and pressure to convert these values to engineering units.

---

**Note:**
Diagnostic/status information is not included with the real-time output. It is stored in memory and retrieved when uploading data from memory.

Instrument diagnostic / status information is saved with each scan:

| Location | Representation | Variable | Encoding |
|---|---|---|---|
| b3-b0 | 0000b | Vout fault | 0 = no fault, 1 = fault. Bit position corresponds to channel. |
| b7-b4 | 0000b | Vout enable | 0 = no fault, 1 = fault. Bit position corresponds to channel. |
| b15-b8 | 0xii | auxiliary current | Raw ADC counts * |
| b23-b16 | 0xii | system current | Raw ADC counts * |
| b24 | 0b | memory full | File system always maintains a few 10s of kB at the end. This bit is set and the scan is stopped. |
| b25 | 0b | battery low | Voltage input too low during cast. This bit is set and scan is stopped. |
| b26 | 0b | serial overflow 1 | Serial overflow occurred on this scan on channel 1. |
| b27 | 0b | serial overflow 2 | Serial overflow occurred on this scan on channel 2. |
| b28 | 0b | pump enable | Pump turned on. |
| b29 | 0b | generic error #1 | Miscellaneous error. |
| b30 | 0b | generic error #2 | Miscellaneous error. |
| b31 | 0b | generic error #3 | Miscellaneous error. |

* To convert Raw ADC counts to current (auxiliary current or system current):
current (milliAmps) = 2.5 * Raw ADC counts / 1024

### SetOutputFormat=1 (P and scan number in Hex)

SBE 25*plus* outputs converted pressure and scan number in Hex. Data is output at 1 Hz, at a baud rate of 9600, regardless of the baud rate set by **SetBaudConsole**=. This is the format required for use with autonomous water sampling systems (SBE 32 Carousel Water Sampler with AFM, or SBE 55 ECO Water Sampler).

ppppssssss
*where*
pressure (decibars) = pppp – 100
scan number = ssssss

---

*Example*: example scan = ppppssssss = 00C80001F0

- pppp = 007B Hex = 123 decimal
- Pressure= pppp = 00C8 Hex (200 decimal);
  pressure (decibars) = 200 – 100 = 100 decibars
- Scan number = ssssss = 0001F0 (496 decimal);
  scan number = 496

---

## Optimizing Data Quality

Our deployment recommendations are only general guidelines, and are focused primarily on maximizing data quality. Actual sea state conditions, and winch and deck gear configurations, may require some adaptation in order to achieve best data quality and avoid mechanical problems.

The SBE 25*plus* is optimized for profiling rates of 0.5 to 2 meters/second, with 1 meter/second generally providing the best compromise between data quality and profile resolution. The 25*plus* pump maintains a constant and optimum flow rate, which ensures that the sensor response times are not dependent on the descent rate. Adjust the descent rate of the 25*plus* based on the amount of ship motion (i.e., sea state) imparted to the cable, and the size, weight, and drag of the underwater package at the end of the cable. Generally, use a faster lowering rate in rougher seas, allowing the 25*plus* to stay clear of turbulent wakes that are shed downward when the ship heaves up; *shed wakes* are an error source from which all CTDs suffer. When the sea is very calm, the 25*plus* can be lowered more slowly (for example, 10 to 20 cm/second) to give greater vertical resolution.

Like for other CTDs, the common configurations of the SBE 25*plus* are intended for obtaining downcast data, because the sensors are oriented so that the T-C Duct inlet passes through relatively undisturbed water as the CTD is lowered. As the CTD is raised (upcast), the sensors sample the turbulent wake of the package, resulting in lower quality data. If planning to sample on the upcast, consider the following:

- Vertical orientation - In the typical, sensors-down configuration, the sensors are in the wake of the housing on the upcast, providing poor data quality. If you must have good quality upcast data, invert the 25*plus* so that the T-C Duct is at the top (ignore the downcast data for this configuration).
- Horizontal orientation (for example, under SBE 32 Carousel Water Sampler) - Upcast data is improved somewhat because the sensors are mounted as close as possible to the outside edge of the package. Position other instruments, sample bottles, etc. so that they are not directly above the T & C sensors and do not thermally contaminate the water that flows to the sensors at the T-C Duct inlet.

When an SBE 25*plus* is installed on a water sampler, good conductivity and optional dissolved oxygen data can be collected, even when stopped to collect a water sample, because water continues to flow through the sensors at a fixed and optimal rate.

**Whether sampling on the upcast and/or downcast, position the T-C Duct inlet so that other instruments, sample bottles, etc. do not thermally contaminate the water that flows past the sensors.**

Where the water temperature is very different from the temperature at which the SBE 25*plus* has been stored, better results are obtained if the 25*plus* is allowed to equilibrate to the water temperature at the surface (soak) for several (3 -5) minutes before beginning the profile. The reason is not that the electronics are temperature sensitive - they are not - but that the thermal influence of the instrument housing on the water entering the cell will be reduced. If the difference between water and storage temperature is extreme, allow more *soak time*.

When very heavy seas cause severe ship motion, the SBE 25*plus* descent can actually stop or reverse direction if the ship heaves upward faster than the package can descend. These occurrences can often be seen as loops in the real-time temperature trace. If the winch payout rate is too high, a loop of wire can form under water when the package descent is slowed or reversed by ship heave and the winch is still paying out wire. Inevitably, the loop formation gets out of phase with the heave and the loop closes on itself when the ship heaves, causing a kink. If the package includes a Carousel Water Sampler as well as the CTD, the package creates much more drag than with only the CTD, further increasing the possibility that a loop will form in the cable. After 1000 to 2000 meters of cable are paid out, it can be difficult or impossible to detect a loop in the cable. Adding 100 to 200 kg (maybe more by experimentation) of lead weights to the Carousel frame will help overcome the effect of drag, and allow the package to descend faster and stay more in line directly below the overboarding sheave.

*Spiking* is sometimes seen in the derived values for salinity, density, or sound velocity. Spiking results largely from a response time mismatch of the conductivity and temperature sensors, especially when the profiling descent rate is non-uniform. Much of the spiking can be removed from the data set by aligning the data in time. This alignment can be performed when post-processing the data in SBE Data Processing.

**Note:**
See SBE Data Processing's Help files for information on data processing modules that can correct data for the influences of ship motion and minimize salinity spiking.

The amount of spiking depends on the temperature gradient, and is much worse when coupled surface motion causes the instrument to stop - or even reverse - its descent. In the event of heavy ship motion, it may be worth letting the instrument *free-fall*. When very heavy seas cause severe ship motion and result in periodic reversals of the instrument descent, the data set can be greatly improved by removing scans taken when the pressure is not increasing, using SBE Data Processing.

## Setup for Deployment

1. Install new batteries (see *Section 5: Routine Maintenance and Calibration*) or ensure the existing batteries have enough capacity to cover the intended deployment.

2. Program the 25*plus* for the intended deployment using Seaterm (see *Section 3: Power and Communications Test* for connection information; see this section for setup/configuration, pump operation, and commands):

   A. Set the date and time (**SetDateTime=yyyy-mm-ddThh:mm:ss**).

   B. Ensure all data has been uploaded, and then send **DeleteAll** to make the entire memory available for recording.

   C. Establish the setup and logging parameters.

   D. Send **GetCD** to verify the setup.

> **Notes:**
> - **The 25*plus* cannot transmit real-time data from a serial output sensor.**
> - A new or recalibrated CTD ships with a configuration file that reflects the current configuration *as we know it*. The file is named with the instrument serial number, followed by a .xmlcon extension. For example, for a CTD with serial number 2375, Sea-Bird names the file *2375.xmlcon*. You may rename the file (but not the extension) if desired; this will not affect the results.

3. If you will be using Seasave to acquire and view real-time data, verify that the configuration (.xmlcon) file matches the instrument configuration. Sea-Bird supplies a configuration file to match the factory configuration and calibrations. If the instrument is recalibrated or the configuration is changed (such as by adding external sensors), the configuration file must be updated to reflect the current condition. See *Verifying Contents of Configuration (.xmlcon) File*.

4. If you will be using Seaterm232 to view real-time data, click the Capture menu to save the data to a file. Enter the desired capture file name in the dialog box, and click Save.

# Deployment

1. Install a cable or dummy plug for each connector on the Main Housing end cap:
   A. Lightly lubricate the inside of the dummy plug/cable connector with silicone grease (DC-4 or equivalent).
   B. **XSG / AG Connector** - Install the plug/cable connector, aligning the raised bump on the side of the plug/cable connector with the large pin (pin 1 - ground) on the 25*plus*. Remove any trapped air by *burping* or gently squeezing the plug/connector near the top and moving your fingers toward the end cap.    **OR**
   **MCBH Connector** – Install the plug/cable connector, aligning the pins.
   C. Place the locking sleeve over the plug/cable connector. Tighten the locking sleeve finger tight only. **Do not overtighten the locking sleeve and do not use a wrench or pliers.**
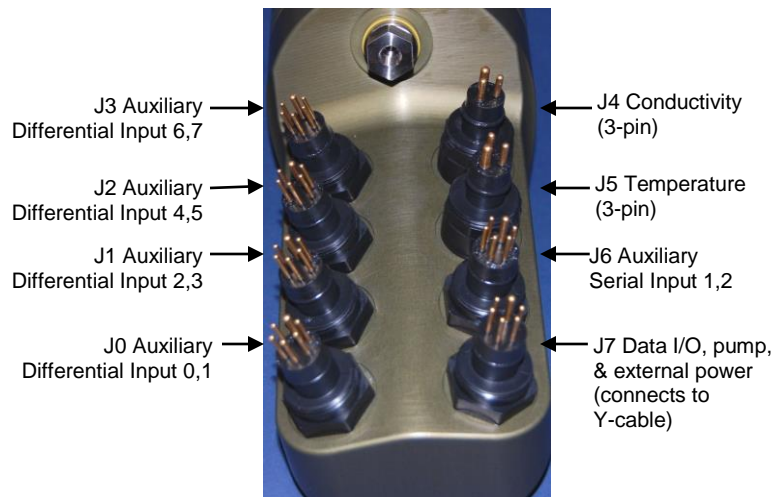
**CAUTIONS:**
- **Do not use WD-40** or other petroleum-based lubricants, as they will damage the connectors.
- For wet-pluggable MCBH connectors: **Silicone lubricants in a spray can** may contain ketones, esters, ethers, alcohols, or glycols in their propellant. **Do not use these sprays, as they will damage the connector.**



2. Connect the other end of the cables installed in Step 1 to the appropriate sensors.

3. Verify that the hardware and external fittings are secure.

4. If applicable, remove the Tygon tubing that was looped end-to-end around the conductivity cell for storage. Reconnect the system plumbing (see *Plumbing* in *Section 2: Description of SBE 25plus*).

5. Immediately prior to deployment:
   - Push the plunger switch in to the On position, **or**
   - Send **StartNow**.

The SBE 25*plus* is ready to go into the water.

## Acquiring Real-Time Data with Seasave

**Notes:**
- Seasave and SBE Data Processing use the same configuration file.
- A new or recalibrated CTD ships with a configuration file that reflects the current configuration *as we know it*. The file is named with the instrument serial number, followed by a .xmlcon extension. For example, for a CTD with serial number 2375, Sea-Bird names the file *2375.xmlcon*. You may rename the file (but not the extension) if desired; this will not affect the results.
- The 25*plus* automatically saves in memory 2 bytes of data for each auxiliary voltage channel, regardless of whether there is a sensor connected to that channel. The .xmlcon file defines which sensors are on each channel, and also defines which channels are to be included in the **real-time** output.

### Verifying Contents of Configuration (.xmlcon) File

Seasave, our real-time data acquisition and display program, requires a configuration file, which defines the CTD – integrated auxiliary sensors, and channels, serial numbers, and calibration dates and coefficients for all sensors (conductivity, temperature, and pressure as well as auxiliary sensors). Seasave (as well as our data processing software) uses the information in the configuration file to interpret and process the raw data. **If the configuration file does not match the actual instrument configuration, the software will not be able to interpret and process the data correctly.**

1. Double click on Seasave.exe.

2. Click Configure Inputs. On the Instrument Configuration tab, click *Open*. In the dialog box, select the .xmlcon and click Open.

3.  The configuration information appears on the Instrument Configuration tab. Verify that the sensors match those on your 25*plus*, and that auxiliary sensors are assigned to the correct channels. Click Modify to bring up a dialog box (shown below) to change the configuration and/or to view / modify calibration coefficients.

**Configuration for the SBE 25plus Sealogger CTD**                                                    ✕

File, T, C, P, Voltage Sensors | Serial Sensors | Real-Time Options |

| New | Open | Save | Save As |

- **New** to create new .xmlcon file for this CTD.
- **Open** to select different .xmlcon file.
- **Save** or **Save As** to save current .xmlcon file settings.

Configuration file opened: 25plustest.xmlcon

Configuration file usage:

Must select *Collect real-time data with Seasave.* Selections on Serial Sensors tab are grayed out.

○ Process .XML file uploaded from CTD memory

● Collect real-time data with Seasave and/or process real-time .HEX file

Click a (non-shaded) sensor and click **Select** to pick a different sensor for that channel. A dialog box with a list of sensors appears.

| Channel | Sensor | |
|---|---|---|
| Frequency | Temperature | |
| Frequency | Conductivity | |
| Count | Pressure, Strain Gauge | |
| A/D voltage 0 | Oxygen, SBE 43 | ☒ |
| A/D voltage 1 | pH | ☐ |
| A/D voltage 2 | Oxidation Reduction Potential | ☐ |
| A/D voltage 3 | Altimeter | ☒ |
| A/D voltage 4 | Fluorometer, WET Labs ECO-AFL/FL | ☐ |
| A/D voltage 5 | Fluorometer, WET Labs ECO CDOM | ☒ |
| A/D voltage 6 | Free | ☐ |
| A/D voltage 7 | Free | ☐ |

Select... | Modify...

Shaded sensors cannot be removed or changed to another type of sensor.

Click a sensor and click **Modify** to change calibration coefficients for that sensor.

Data from checked voltage channels transmitted in real-time, and included in .hex file.

Select voltage channels that you want to view in real-time. Must match **SetVOut#=** (#=0-7) settings in 25*plus*. See *Real-Time Setup* for baud and real-time output rate limitations.

| Report | Help | Exit | Cancel |

**Note:**
Some serial sensors, such as the SBE 38, transmit data in engineering units. Consequently, no calibration coefficients are entered for these sensors.

Verify that the sensors match those on your 25*plus*, and that auxiliary sensors are assigned to the correct serial channels. Verify that calibration coefficients for all sensors are up-to-date.

4.    Click the Real-Time Options tab.

**Configuration for the SBE 25plus Sealogger CTD**

File, T, C, P, Voltage Sensors | Serial Sensors | **Real-Time Options**

Select if deck unit used, and select baud rate at which CTD is set to communicate.
Note: If no deck unit is used, this baud rate must match CTD Serial Port baud on Serial Ports tab in Configure Inputs.

○ No deck unit
● SBE 33 or SBE 36 deck unit
CTD baud rate (SetBaudConsole=)    [4800 ▼]

Real-time data output rate is calculated by the 25plus, based on number of voltages exported, CTD baud rate, and (if CTD baud rate is 4800), historic rate.

Must agree with **SetHistoricRate=** in 25*plus*. See reply from **GetCD** and *Real-Time Setup*.

● Historic rate = 0; Real-time data output rate = 2 Hz or 4 Hz
○ Historic rate = 1; Real-time data output rate = 4 Hz or 8 Hz

**NMEA** - Select if NMEA navigation device used, **and** whether NMEA device is connected directly to Deck Unit **or** to computer. You can also append NMEA depth data (3 bytes) and NMEA time data (4 bytes) after Lat/Lon data. Seasave adds current latitude, longitude, and universal time code to data header; appends NMEA data to every scan; and writes NMEA data to .nav file every time Ctrl F7 is pressed or Add to .nav File is clicked. Note: Whether NMEA device was connected to a deck unit or directly to computer during data acquisition in Seasave has no effect on data file used by SBE Data Processing, and therefore has no effect on data processing.

☑ NMEA position data added            ☐ NMEA depth data added
● NMEA device connected to deck unit            ☐ NMEA time added
○ NMEA device connected to PC

☑ Surface PAR voltage added            ☐ Scan time added

Configure SPAR Sensor...

Enter/verify calibration coefficients for Surface PAR sensor. See Application Note 47.

Select if using with deck unit connected to Surface PAR sensor. Seasave appends Surface PAR data to every scan.

Select to have Seasave append time (seconds since January 1, 1970 GMT) to each data scan.

**Report**            **Help**            **Exit**            **Cancel**

Verify that the setup matches the setup of your 25*plus*.

5.    On the File, T, C, P, Voltage Sensors tab, click *Save* or *Save As* to save any changes to the .xmlcon file. Click Exit when done reviewing / modifying the configuration.

## Acquiring Real-Time Data

Instructions below are for an SBE 25*plus* with a conventional single-core armored cable, used without a Sea-Bird Deck Unit. **If using the SBE 25*plus* with the SBE 33 or 36 Deck Unit or the Seacat/Sealogger RS-232 and Navigation Interface Box, see the Deck Unit or Interface Box manual.**

**Notes:**
- **The 25*plus* cannot transmit real-time data from a serial output sensor.**
- The baud rate between the 25*plus* and computer (defined in Configure Inputs, on the Serial Ports tab) must match the baud rate set in the 25*plus* with **SetBaudConsole=**.

1. Wiring - Terminate the single-core armored cable with an RMG-4FS or MCIL-4FS connector, as applicable. Wire the cable armor to pin 1 (large pin for 25*plus* with XSG / AG connectors) and the inner conductor to pin 3 (opposite pin 1) on the 25*plus* data I/O-power connector (4-pin leg on Y-cable connected to J7). On deck, wire:

| Slip-ring lead | 25-pin serial port | 9-pin serial port |
| --- | --- | --- |
| from armor | Pin 7 | Pin 5 |
| from inner conductor | Pin 3 | Pin 2 |

2. Double click on **Seasave.exe**.

3. Perform any desired setup in the Configure Inputs, Configure Outputs, and Display menus..

4. In the Real-Time Data menu, select *Start*. The dialog box looks like this:

**Data Archiving Options:**
- **Begin archiving data immediately** to store raw (frequencies, A/D counts, and/or voltages) real-time data as soon as Start button is clicked and communication is established.
- **Begin archiving data when 'Start Archiving' command is sent** to control when data begins to be written to file. This allows you to eliminate scans associated with deploying CTD from ship deck and soaking instrument near surface (alternatively, remove these scans later in SBE Data Processing). If you make this selection, when you click Start button and communication is established, a dialog box with Start Archiving button appears. Click this button when ready to begin saving scans to file, or select Start Archiving in Real-Time Data menu.
- **Do not archive data for this cast** to not save data to a file. Real-time data will still appear in displays.

**Configuration Options**: Currently selected instrument configuration (.xmlcon) file is shown, containing information on number and type of sensors interfacing with 25*plus*, calibration coefficients, and inclusion of NMEA and/or Surface PAR data with output from CTD. To modify input configuration (.xmlcon file, serial ports, water sampler, TCP/IP ports, and/or miscellaneous), click Configure Inputs. To modify outputs (serial data output, serial ports, shared file output, mark variables, TCP/IP output, TCP/IP ports, SBE 14 remote display, PC alarms, header form, and/or diagnostics), click Configure Outputs.

- **Timeout at startup**: Time allowed before first data scan is received from 25*plus*. Seasave *times out* and stops attempting to acquire data if data is not received within this time period.
- **Timeout between scans**: Maximum **gap** allowed between scans after first scan is received. Seasave *times out* and stops attempting to acquire data if data is not received within this time period (for example, if a shark bites cable and interrupts data acquisition, Seasave stops attempting to acquire data after this gap).

(Appears only if .xmlcon file indicates that NMEA data is added through deck unit.) If selected, Seasave resets deck unit (SBE 33, SBE 36, or Seacat/Sealogger RS-232 and Navigation Interface Box) to State 1 when acquisition stops. State 1 is Echo only mode (no NMEA or Surface PAR data is sent), and is useful for setting up CTD.

**Start Real-Time Data Acquisition**

Data Archiving Options

- ● Begin archiving data immediately
- ○ Begin archiving data when 'Start Archiving' command is sent
- ○ Do not archive data for this cast

Output data [.HEX] file

C:\Documents and Settings\dbresko.SEABIRD\My Documents\25plustest.hex

Select Output Data File Name

Click **Select Output Data File Name**. Save Archived Data As dialog box appears; browse to desired file location, enter desired file name, and click Save.

Configuration Options

Instrument configuration [.xmlcon or .con] file: (to change select Configure Inputs)

C:\Documents and Settings\dbresko.SEABIRD\My Documents\25plustest.xmlco

Configure Inputs      Configure Outputs

Timeout in seconds at startup   60

Timeout in seconds between scans   20

☑ Reset deck unit to state 1 when acquisition stops

Report   Help   Start   Exit   Cancel

Make the desired selections.

5. In the Start Real-Time Data Acquisition dialog box, click *Start*.

   A. If you selected *Begin archiving data immediately* or *Begin archiving data when 'Start Archiving' command is sent* above, and selected *Prompt for Header Information* in the Header Form setup (Configure Outputs), the Header Information dialog box appears. Fill in the desired header and click OK.

   B. If you selected *NMEA position data added* in the .xmlcon file, Seasave initializes NMEA communications.

   C. If you selected *Check Scan Length* in the Options menu, Seasave checks the .xmlcon file to verify that the scan length defined by the configuration file matches the 25*plus* (i.e., number of sensors and inclusion of NMEA is as defined in the file). If a *Scan length error* appears, verify that:

      • You are using the correct .xmlcon configuration file.
      • The configuration file has been updated as necessary if you added or deleted sensors, added or deleted NMEA or Surface PAR, etc.

   D. Seasave sends a message: *Please turn on the SBE 25plus using the magnetic switch*.

      • If you have not already started logging, push the switch in to the On position.
      • If you have already started logging data, ignore the message. Note: You may not see the message if Seasave gets the first scans quickly.

      Seasave *times out* if data is not received within *Timeout in seconds at startup*.

   E. Real-time data then starts appearing in the screen displays.

6. To stop real-time data acquisition: In the Real-Time Data menu, select *Stop*.

7. Stop logging: Pull the 25*plus* switch out to the Off position.

## Recovery

> **WARNING!**
> **If the SBE 25*plus* stops working while underwater, is unresponsive to commands, or shows other signs of flooding or damage, carefully secure it away from people until you have determined that abnormal internal pressure does not exist or has been relieved.** Pressure housings may flood under pressure due to dirty or damaged o-rings, or other failed seals. When a sealed pressure housing floods at great depths and is subsequently raised to the surface, water may be trapped at the pressure at which it entered the housing, presenting a danger if the housing is opened before relieving the internal pressure. Instances of such flooding are rare. However, a housing that floods at 5000 meters depth holds an internal pressure of more than 7000 psia, and has the potential to eject the end cap with lethal force. A housing that floods at 50 meters holds an internal pressure of more than 85 psia; this force could still cause injury.
> If you suspect the SBE 25*plus* is flooded, point it in a safe direction away from people, and loosen 1 end cap bulkhead connector very slowly, at least 1 turn. This opens an o-ring seal under the connector. Look for signs of internal pressure (hissing or water leak). If internal pressure is detected, let it bleed off slowly past the connector o-ring. Then, you can safely remove the end cap.

### Physical Handling

1. Rinse the conductivity cell with fresh water. (See *Section 5: Routine Maintenance and Calibration* for cell cleaning and storage.)

2. If the batteries are exhausted, new batteries must be installed before the data can be extracted. Stored data will not be lost as a result of exhaustion or removal of batteries. (See *Section 5: Routine Maintenance and Calibration* for replacement of batteries.)

3. If immediate redeployment is not required, it is best to leave the SBE 25*plus* with batteries in place and in a quiescent state (**QS**). Because the quiescent current on required is only 70 microAmps, the batteries can be left in place without significant loss of capacity. If the 25*plus* is to be stored for long periods, **replace alkaline batteries yearly to prevent battery leakage** (which could damage the 25*plus*).

# Uploading Data

**Notes:**
- If your 25*plus* includes two auxiliary RS-232 serial sensors of the same type, with data saved inline (**SetInlineSer#=y**), the data **may** contain null characters that prevent uploading of the data via RS-232. Upload via the USB port in the battery pack.
- Some USB cable connectors do not seat properly on the USB port in the 25*plus* battery pack, preventing communications. Because the power and ground pins are longer, the indicator light may come on even if all of the pins are not securely connected. If you cannot establish communications via the USB port, try again with a different USB cable.

After recovery, data can be uploaded with one of the following methods:
- With the computer connected to the Y-cable from the 25*plus* J7 connector, use Seaterm232's Upload menu to upload one or more files via RS-232. **OR**
- (**much faster, recommended for large amounts of data**) With the 25*plus* battery end cap removed and a USB cable connected to the USB port in the battery pack, drag and drop (or copy and paste) the files in memory onto your computer.

Both methods are detailed below.

## Uploading Data via RS-232

1. Double click on **SeatermV2.exe**. The main screen appears.

2. In the Instruments menu, select *SBE 25plus*. **Seaterm232** opens.

**Note:**
Connect the I/O cable as described in *Section 3: Power and Communications Test.*

3. Seaterm232 will try to automatically connect to the 25*plus*. As it connects, it sends **GetHD** and displays the response. Seaterm232 also fills the Send Commands window with the correct list of commands for your 25*plus*. **If there is no communication**:

   A. In the Communications menu, select *Configure*. The Serial Port Configuration dialog box appears. Select the Comm port and baud rate for communication, and click OK.
   B. In the Communications menu, select *Connect* (if *Connect* is grayed out, select *Disconnect and reconnect*).
   C. If there is still no communication, check cabling between the computer and 25*plus*.
   D. If there is still no communication, repeat Step A with a different baud rate and/or comm port, and try to connect again. Note that the factory-set baud rate is documented on the Configuration Sheet.

**Note:**
You may need to send **Stop** several times to get the 25*plus* to respond.

4. Command the 25*plus* to stop data logging by one of the following methods:
   - Type **Stop** and press the Enter key, or
   - Pull the plunger switch out to the Off position

5. Click Upload to upload stored data. Seaterm232 responds as follows:
   A. Seaterm232 sends **SetOutputExecutedTag=Y**, to put the 25*plus* into the configuration required for data upload.
   B. Seaterm232 sends **GetSD** and displays the response. **GetSD** provides information on the instrument status, and number of samples in memory.
   C. Seaterm232 sends **GetFiles** and displays the response. **GetFiles** provides information on all of the files in memory.
   D. An Upload Data dialog box appears:

Select files to upload. Files are within folders that are organized by date. There are two file types:
- .xml – Data file from CTD, labeled with UTC date and time that cast started, and with 25*plus* serial number. Includes serial sensor data if **SetEnableSer1=Y** and/or **SetEnableSer2=Y**, and **SetInlineSer1=Y** and/or **SetInlineSer2=Y**.
- .txt – If your 25*plus* is set up with 1 or more serial sensors (**SetEnableSer1=Y** and/or **SetEnableSer2=Y**), and serial sensor data is not stored within CTD data file (**SetInlineSer1=N** and/or **SetInlineSer2=N**), a .txt file is created with serial sensor data for each cast. Beginning of file name is same as .XML file name (UTC date and time that cast started), but then says *serial1* or *serial2* instead of 25*plus* serial number.

Click Select Folder to navigate to desired upload file path. Upload files have .xml or .txt extensions. .xml files are compatible with Seasave and SBE Data Processing. You must provide your own software to process .txt files.

**Sub-folder options:**
- Place all files in destination folder – for this example, 3 files selected would be directly placed in Destination folder.
- Place files in sub-folders by cast start date - for this example, there would be 2 sub-folders (2012-03-10, and 2012-03-13), within Destination folder.
- Place files in sub-folders by cast start date and cast start time - for this example, there would be 2 sub-folders (2012-03-10T134108, and 2012-03-13T22726), within Destination folder.
**Name append:** If desired, add an identintifying name before extension, to all files to be uploaded. For example, if using a Name Append of HOT, file 2012-03-13T122726 SBE0250003.xml would be uploaded as 2012-03-13T122726 SBE0250003**HOT**.xml, etc.



Set number of bytes uploaded in each block. Seaterm232 uploads data in blocks, and calculates a checksum at end of each block. If block fails checksum verification, Seaterm232 tries to upload block of data again, cutting block size in half.
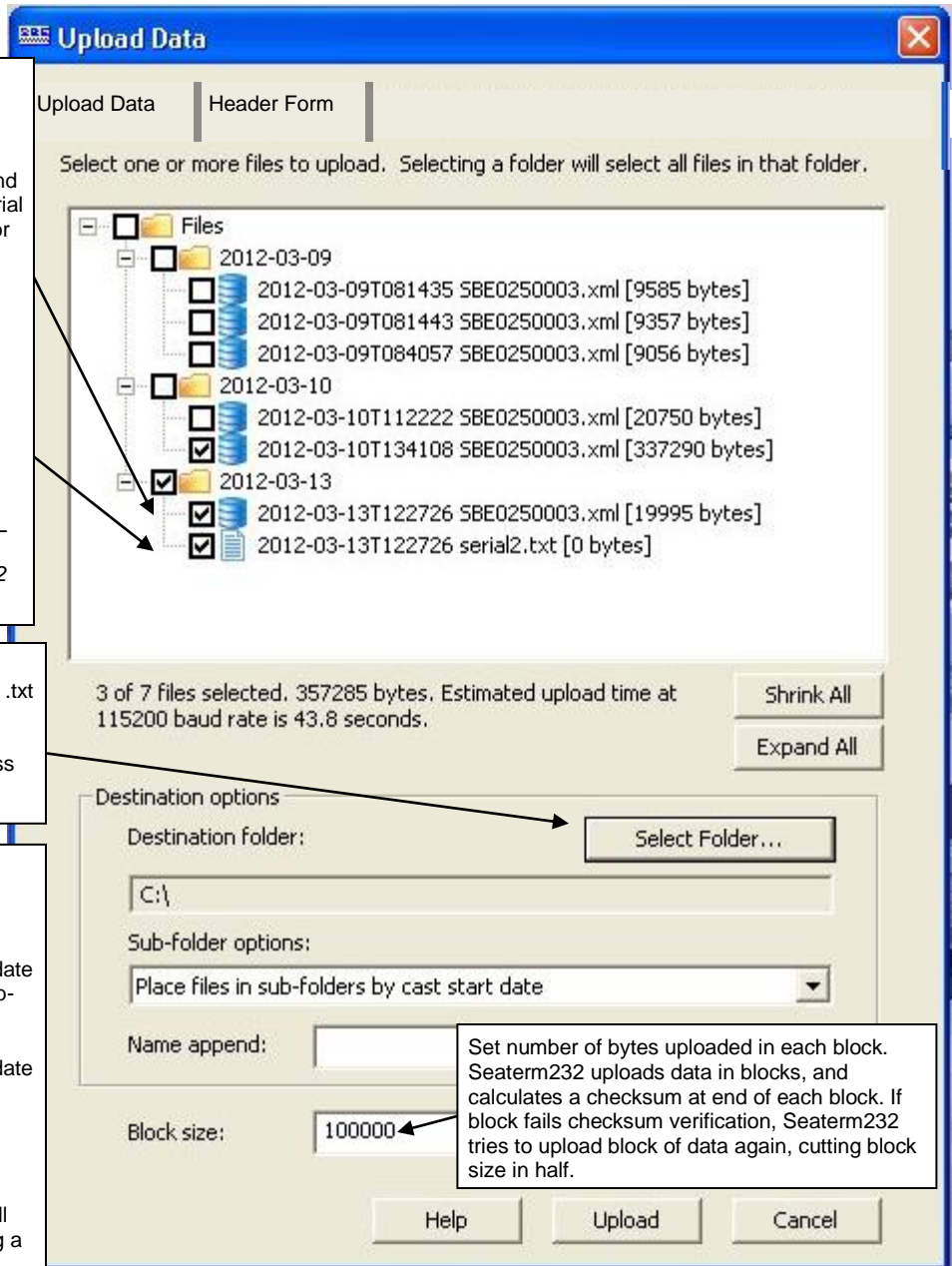
Make the desired selections.

6. Click the Header Form tab to customize the header:

<table><tr><td>Defines header information included with uploaded data:
• Prompt for header information – Each time data is uploaded, user is prompted to fill out user-defined header form.
• Include default header form in upload file – User-defined default header form included in upload file. User is not prompted to add any information when data is uploaded.
• Don't include default header form in upload file – Header information not included in upload file.</td><td>

**Upload Data**  ×

Upload Data | Header Form |

Header Choice   [Prompt for Header Information ▼]

Prompt for line # 01  [Ship:]
Prompt for line # 02  [Cruise:]
Prompt for line # 03  [Station:]
Prompt for line # 04  [Latitude:]
Prompt for line # 05  [Longitude:]
Prompt for line # 06  [ ]
Prompt for line # 07  [ ]
Prompt for line # 08  [ ]
Prompt for line # 09  [ ]
Prompt for line # 10  [ ]
Prompt for line # 11  [ ]
Prompt for line # 12  [ ]
</td></tr></table>

The entries are free form, 0 to 12 lines long. This dialog box establishes:
• the header prompts that appear for the user to fill in when uploading data, if *Prompt for header information* was selected
• the header included with the uploaded data, if *Include default header form in upload file* was selected

Enter the desired header/header prompts.

**Notes**:
• Sea-Bird software (SBE Data Processing and Seasave) cannot process serial sensor data in the uploaded .txt file.
• If your 25*plus* includes one or more auxiliary RS-232 serial sensors with data saved inline (**SetInlineSer#=y**), the data **may** contain null characters that cause the RS-232 upload to fail; upload via the internal USB connector (see *Uploading Data via USB* below).

7. Click Upload; the Status bar at the bottom of the window displays the upload progress:
   A. **If you selected *Prompt for header information* in the Upload Data dialog box** – a dialog box with the header form appears. Enter the desired header information, and click OK. Seaterm232 writes the header information to the upload file.
   B. Seaterm232 sends the data upload command, and writes the data to the upload .xml file or .txt file, as applicable. The .xml file contains the data in raw hexadecimal, for compatibility with Seasave and SBE Data Processing.
   C. **If you selected *multiple files*** – Seaterm232 repeats Steps A and B for each cast.
   D. When the data has been uploaded, Seaterm232 shows the S> prompt.

**Notes**:
To prepare for redeployment:
1. After all data has been uploaded, send **DeleteAll**. If this command is not sent, old data will not be overwritten, preventing use of the entire memory capacity.
2. Send **QS** to put the 25*plus* in quiescent (sleep) state until ready to redeploy. Quiescent current is only 70 microamps, so the batteries can be left in place without significant loss of capacity.

8. Ensure all data has been uploaded by reviewing and processing the data:
   A. Use **Seasave** to display the *raw* (unprocessed) hexadecimal data from the .xml file in engineering units (see *Verifying Contents of Configuration (.xmlcon) File* and Seasave's manual / Help files).
   B. Use **SBE Data Processing** to process and plot the data (see *Processing Data Using SBE Data Processing* and SBE Data Processing's manual / Help files).

## Uploading Data via USB



Unthread cap by rotating counter-clockwise



USB connector

1. Command the 25*plus* to stop data logging by pulling the plunger switch out to the Off position.

2. Remove the main housing battery end cap (end cap without connectors):
   A. Wipe the outside of the end cap and housing dry, being careful to remove any water at the seam between them.
   B. Unthread the end cap by rotating counter-clockwise (use a wrench on the white plastic bar if necessary).
   C. Remove any water from the O-ring mating surfaces inside the housing with a lint-free cloth or tissue.
   D. Put the end cap aside, being careful to protect the O-ring from damage or contamination.

3. With the battery pack installed in the 25*plus*, plug the provided USB cable into the USB port in the 25*plus* battery pack. Plug the other end of the USB cable into your computer USB port.
   **NOTE:** Some USB cable connectors do not seat properly on the USB port in the 25*plus* battery pack, preventing communications. Because the power and ground pins are longer, the indicator light may come on even if all of the pins are not securely connected. If you cannot establish communications via the USB port, try again with a different USB cable.

4. Your computer should open a window that displays the contents of the 25*plus* memory. The file structure looks like this:



Folders for each day's casts, automatically named with date (year-month-day) at **start** of cast.

Config folder only appears if *Show hidden files and folders* is enabled in Windows.
**Do not make changes to this folder.**

Select files to upload. Files are within folders that are organized by date. There are two file types:
- .xml – Data file from CTD, labeled with UTC date and time that cast started, and with 25*plus* serial number. Includes serial sensor data if **SetEnableSer1=Y** and/or **SetEnableSer2=Y**, and **SetInlineSer1=Y** and/or **SetInlineSer2=Y**.
- .txt – If your 25*plus* is set up with 1 or more serial sensors (**SetEnableSer1=Y** and/or **SetEnableSer2=Y**), and serial sensor data is not stored within CTD data file (**SetInlineSer1=N** and/or **SetInlineSer2=N**), a .txt file is created with serial sensor data for each cast. Beginning of file name is same as .XML file name (UTC date and time that cast started), but then says *serial1* or *serial2* instead of 25*plus* serial number.

5. Drag the desired file(s) from the window into the desired location on your computer; note that **this does not delete the file(s)** from the 25*plus* memory. Alternatively, copy the desired file(s) and paste them into the desired location on your computer.

6. To delete the files that you copied onto your computer: Select the desired file(s) and click Delete on your keyboard.

7. Ensure all data has been uploaded by reviewing and processing the data:
   A. Use **Seasave** to display the *raw* (unprocessed) hexadecimal data in engineering units (see *Verifying Contents of Configuration (.xmlcon) File* and Seasave's manual / Help files).
   B. Use **SBE Data Processing** to process and plot the data (see *Processing Data Using SBE Data Processing* and SBE Data Processing's manual / Help files).

**Notes**:
To prepare for redeployment:
1. After all data has been uploaded, send **DeleteAll**. If this command is not sent, old data will not be overwritten, preventing use of the entire memory capacity.
2. Send **QS** to put the 25*plus* in quiescent (sleep) state until ready to redeploy. Quiescent current is only 70 microamps, so the batteries can be left in place without significant loss of capacity.

# Processing Data Using SBE Data Processing

**Notes:**
- Seasave and SBE Data Processing use the same configuration file.
- A new or recalibrated CTD ships with a configuration file that reflects the current configuration *as we know it*. The file is named with the instrument serial number, followed by a .xmlcon extension. For example, for a CTD with serial number 2375, Sea-Bird names the file *2375.xmlcon*. You may rename the file (but not the extension) if desired; this will not affect the results.
- The 25*plus* automatically saves in memory 2 bytes of data for each auxiliary voltage channel, regardless of whether there is a sensor connected to that channel. The .xmlcon file defines which sensors are on each channel, and also defines which channels are to be included in the real-time output.

1. Convert the .xml file (data uploaded from 25*plus* memory) or .hex file (real-time data saved by Seasave) to a .cnv (engineering units) file in SBE Data Processing's Data Conversion module. Verify that the configuration (.xmlcon) file is correct before converting the data; see *Verifying Contents of Configuration (.xmlcon) File* below.

   - If your 25*plus* includes two auxiliary RS-232 serial sensors of the same type, with data saved inline (**SetInlineSer#=y**), the data **may** contain null characters that prevent proper processing of the serial sensor data. The symptom of this problem is that Data Conversion produces a file with *bad flags* for the second auxiliary serial sensor data. If this occurs, run **StripNullChars.exe**, which is installed with SBE Data Processing version 7.22.5 and later (in same directory as SBE Data Processing). This program removes null characters from an uploaded 25*plus* data file; the file can then be processed in Data Conversion.
     - ➢ Run StripNULLchars.exe from a DOS window, following instructions provided in the software.
     - ➢ Note: The null characters in the file also prevent uploading of the data from the 25*plus* via RS-232. You must open the 25*plus* and upload via the internal USB connector.

2. Once the data is converted: perform further processing (align, filter, remove bad data, etc.), calculate derived variables, and plot data using SBE Data Processing's other modules.

## Verifying Contents of Configuration (.xmlcon) File

To convert the.xml or .hex file, you need a .xmlcon configuration file, which defines the instrument – integrated sensors, and channels, serial numbers, and calibration dates and coefficients for all sensors (conductivity, temperature, and pressure as well as auxiliary sensors). SBE Data Processing (as well as our real-time data acquisition software) uses the configuration file information to interpret and process the raw data. **If the configuration file does not match the actual instrument configuration, the software will be unable to interpret and process the data correctly.**

To view or modify the .xmlcon file:

1.  Double click on SBEDataProc.exe.

2.  In the Configure menu, select *SBE 25plus Sealogger CTD*. The configuration dialog box appears; click Open. In the Open dialog box, select the .xmlcon file and click Open.

## Configuration for the SBE 25plus Sealogger CTD

**File, T, C, P, Voltage Sensors** | Serial Sensors | Real-Time Options

New   Open   Save   Save As

- **New** to create new .xmlcon file for this CTD.
- **Open** to select different .xmlcon file.
- **Save** or **Save As** to save current .xmlcon file settings.

Configuration file opened: 25plustest.xmlcon

- **.XML file** – if selected, selections on Real-Time Options tab are grayed out.
- **.HEX file** - if selected, selections on Serial Sensors tab are grayed out.

Configuration file usage:

(•) Process .XML file uploaded from CTD memory

( ) Collect real-time data with Seasave and/or process real-time .HEX file

Click a (non-shaded) sensor and click **Select** to pick a different sensor for that channel. A dialog box with a list of sensors appears.

Shaded sensors cannot be removed or changed to another type of sensor.

| Channel | Sensor | |
|---|---|---|
| Frequency | Temperature | |
| Frequency | Conductivity | |
| Count | Pressure, Strain Gauge | |
| A/D voltage 0 | Oxygen, SBE 43 | ☒ |
| A/D voltage 1 | pH | ☐ |
| A/D voltage 2 | Oxidation Reduction Potential | ☐ |
| A/D voltage 3 | Altimeter | ☒ |
| A/D voltage 4 | Fluorometer, WET Labs ECO-AFL/FL | ☐ |
| A/D voltage 5 | Fluorometer, WET Labs ECO CDOM | ☒ |
| A/D voltage 6 | Free | ☐ |
| A/D voltage 7 | Free | ☐ |

Select...

Modify...

Click a sensor and click **Modify** to change calibration coefficients for that sensor.

Data from checked voltage channels transmitted in real-time, and included in .hex file.

If processing .HEX file collected in Seasave, must match settings used in Seasave for real-time data acquisition. If processing .XML file uploaded from memory, selections of real-time data from voltage channels have no effect.

Report   Help   Exit   Cancel

Verify that the sensors match those on your 25*plus*, and that auxiliary sensors are assigned to the correct A/D voltage channels. Verify that calibration coefficients for all sensors are up-to-date.

**Note:**
This tab is grayed out if you selected *Collect real-time data with Seasave and/or process real-time .HEX file* on the first tab, because the 25*plus* does not transmit real-time serial sensor data.

3. If you selected *Process .XML file uploaded from CTD memory*, click the Serial Sensors tab.

25*plus* collected serial sensor data if**SetEnableSer1=Y**. Serial sensor data was included in CTD .xml data file in memory if **SetInlineSer1=Y**. Otherwise, it was placed in a separate .txt file (which cannot be processed by SBE Data Processing); if in a .txt file, all selections are grayed out.

25*plus* collected serial sensor data if**SetEnableSer2=Y**. Serial sensor data was included in CTD .xml data file in memory if **SetInlineSer2=Y**. Otherwise, it was placed in a separate .txt file (which cannot be processed by SBE Data Processing); if in a .txt file, all selections are grayed out.

Select serial sensor that is on each serial channel: Not applicable, SBE 38 (secondary temperature), SBE 50 (secondary pressure), SBE 63 (optical DO), WET Labs sensor (up to 3 channels), or WET Labs SeaOWL UV-A.



**Note:**
Some serial sensors, such as the SBE 38, transmit data in engineering units. Consequently, no calibration coefficients are entered for these sensors.

Verify that the sensors match those on your 25*plus*, and that auxiliary sensors are assigned to the correct serial channels. Verify that calibration coefficients for all sensors are up-to-date.

**Note:**
This tab is grayed out if you selected *Process .XML file uploaded from CTD memory* on the first tab, because data is memory is always saved at 16 Hz, and NMEA, Surface PAR, and scan time data is not available in an uploaded file.

4.  If you selected *Collect real-time data with Seasave and/or process real-time .HEX file*, click the Real-Time Options tab.

Select if deck unit used, and select baud rate at which CTD was set to communicate.

Must agree with **SetHistoricRate=** in 25*plus*. See reply from **GetCD**.

**NMEA** - Select if NMEA navigation device used, **and** whether NMEA device is connected directly to Deck Unit **or** to computer. You can also append NMEA depth data (3 bytes) and NMEA time data (4 bytes) after Lat/Lon data. Seasave adds current latitude, longitude, and universal time code to data header; appends NMEA data to every scan; and writes NMEA data to .nav file every time Ctrl F7 is pressed or Add to .nav File is clicked. Note: Whether NMEA device was connected to a deck unit or directly to computer during data acquisition in Seasave has no effect on data file used by SBE Data Processing, and therefore has no effect on data processing.

Enter/verify calibration coefficients for Surface PAR sensor. See Application Note 47.

Select if using with deck unit connected to Surface PAR sensor. Seasave appends Surface PAR data to every scan.

Select if Seasave appended time (seconds since January 1, 1970 GMT) to each data scan.

### Configuration for the SBE 25plus Sealogger CTD

File, T, C, P, Voltage Sensors | Serial Sensors | **Real-Time Options**

○ No deck unit
● SBE 33 or SBE 36 deck unit
CTD baud rate (SetBaudConsole=)   [ 4800 ▼ ]

Real-time data output rate is calculated by the 25plus, based on number of voltages exported, CTD baud rate, and (if CTD baud rate is 4800), historic rate.

● Historic rate = 0; Real-time data output rate = 2 Hz or 4 Hz
○ Historic rate = 1; Real-time data output rate = 4 Hz or 8 Hz

☑ NMEA position data added                      ☐ NMEA depth data added
● NMEA device connected to deck unit        ☐ NMEA time added
○ NMEA device connected to PC

☑ Surface PAR voltage added                     ☐ Scan time added
[ Configure SPAR Sensor... ]

[ Report ]    [ Help ]    [ Exit ]    [ Cancel ]

Verify that the setup matches the setup of your 25*plus*.

5.  On the File, T, C, P, Voltage Sensors tab, click *Save* or *Save As* to save any changes to the .xmlcon file. Click Exit when done reviewing / modifying the configuration.

# Editing Raw Data File

Sometimes users want to edit the raw .hex data file before beginning processing, to remove data at the beginning of the file corresponding to instrument *soak* time, to remove blocks of bad data, to edit the header, or to add explanatory notes about the cast. **Editing the raw .hex file can corrupt the data, making it impossible to perform further processing using Sea-Bird software.** Sea-Bird strongly recommends that you first convert the data to a .cnv file (using Data Conversion in SBE Data Processing), and then use other SBE Data Processing modules to edit the .cnv file as desired.

The procedure for editing a .hex data file described below has been found to work correctly on computers running Windows 98, 2000, and NT. **If the editing is not performed using this technique, SBE Data Processing may reject the edited data file and give you an error message.**

1.  **Make a back-up copy of your .hex data file before you begin.**

2.  Run **WordPad**.

3.  In the File menu, select Open. The Open dialog box appears. For *Files of type*, select *All Documents (\*.\*)*. Browse to the desired .hex data file and click Open.

**Note:**
Although we provide this technique for editing a raw .hex file, **Sea-Bird's strong recommendation, as described above, is to always convert the raw data file and then edit the converted file.**

4.  Edit the file as desired, **inserting any new header lines after the System Upload Time line**. Note that all header lines must begin with an asterisk (\*), and \*END\* indicates the end of the header. An example is shown below (for an SBE 21), with the added lines in bold:
    ```
    * Sea-Bird SBE 21 Data File:
    * FileName = C:\Odis\SAT2-ODIS\oct14-19\oc15_99.hex
    * Software Version Seasave Win32 v1.10
    * Temperature SN = 2366
    * Conductivity SN = 2366
    * System UpLoad Time = Oct 15 1999  10:57:19
    * Testing adding header lines
    * Must start with an asterisk
    * Place anywhere between System Upload Time & END of header
    * NMEA Latitude = 30 59.70 N
    * NMEA Longitude = 081 37.93 W
    * NMEA UTC (Time) = Oct 15 1999  10:57:19
    * Store Lat/Lon Data = Append to Every Scan and Append to .NAV
    File When <Ctrl F7> is Pressed
    ** Ship:         Sea-Bird
    ** Cruise:       Sea-Bird Header Test
    ** Station:
    ** Latitude:
    ** Longitude:
    *END*
    ```

5.  In the File menu, select Save (**not** Save As). If you are running Windows 2000, the following message displays:
    ```
    You are about to save the document in a Text-Only
    format, which will remove all formatting. Are you
    sure you want to
    do this?
    ```
    Ignore the message and click *Yes*.

6.  In the File menu, select Exit.

# Section 5: Routine Maintenance and Calibration

This section reviews corrosion precautions, connector mating and maintenance, plumbing air valve maintenance, battery replacement, conductivity cell storage and cleaning, pressure sensor maintenance, pump maintenance, O-ring maintenance, TC duct disassembly/reassembly, and sensor calibration. The SBE 25*plus'* accuracy is sustained by the care and calibration of the sensors and by establishment of proper handling practices.

## Corrosion Precautions

Rinse the SBE 25*plus* with fresh water after use and prior to storage.

### Plastic Main Housing

Stainless steel screws connecting the plunger switch mount to the connector end cap have been generously lubricated with Blue Moly™. All other stainless steel screws that are exposed to salt water have been generously lubricated with DC-4. After each cruise, remove all these screws and re-lubricate. **These compounds are electrically conductive; use care to ensure they do not get on PCBs.**

The SBE 25*plus* has a large zinc anode screwed into the connector end cap and the battery end cap. Check the anode periodically to verify that it is securely fastened and has not been eaten away.

Avoid direct attachment of metal objects to the aluminum end cap.

### Aluminum Main Housing

All stainless steel screws that are exposed to salt water have been generously lubricated with Blue Moly™. After each cruise, remove these screws and re-lubricate. **This compound is electrically conductive; use care to ensure it does not get on PCBs.**

> **Note:**
> Sea-Bird recommends that you replace an anode when more than 50% of the material has eroded.

The SBE 25*plus* has a large zinc anode screwed into the battery end cap, the connector end cap, and the housing (near the connector end cap). Check the anodes periodically to verify that they are securely fastened and have not been eaten away.

The housing is insulated from the stainless steel guard cage and sea cable power circuits, preventing heavy corrosion of the housing. Avoid direct attachment of metal objects to the housing.

### Modular Sensors

The modular temperature and conductivity sensors have ring-shaped zinc anodes. Check the anodes periodically to verify that they are securely fastened and have not been eaten away.

## Connector Mating and Maintenance

> **Note:**
> See *Application Note 57: Connector Care and Cable Installation*.

> **CAUTIONS:**
> - **Do not use WD-40** or other petroleum-based lubricants, as they will damage the connectors.
> - For wet-pluggable MCBH connectors: **Silicone lubricants in a spray can** may contain ketones, esters, ethers, alcohols, or glycols in their propellant. **Do not use these sprays, as they will damage the connector.**



Locking sleeve

Clean and inspect connectors, cables, and dummy plugs before every deployment and as part of your yearly equipment maintenance. Inspect connectors that are unmated for signs of corrosion product around the pins, and for cuts, nicks or other flaws that may compromise the seal.

When remating:

1. Lightly lubricate the inside of the dummy plug/cable connector with silicone grease (DC-4 or equivalent).
2. **XSG / AG Connector** - Install the plug/cable connector, aligning the raised bump on the side of the plug/cable connector with the large pin (pin 1 - ground) on the SBE 25*plus*. Remove any trapped air by *burping* or gently squeezing the plug/connector near the top and moving your fingers toward the end cap.     **OR**
   **MCBH Connector** – Install the plug/cable connector, aligning the pins.
3. Place the locking sleeve over the plug/cable connector. Tighten the locking sleeve finger tight only. **Do not overtighten the locking sleeve and do not use a wrench or pliers.**

Verify that a cable or dummy plug is installed for each connector on the system before deployment.

## Plumbing Maintenance



Y-fitting with air release valve

For an SBE 25*plus* deployed in a vertical orientation -
**A clogged air release valve can trap air, preventing the pump from functioning properly**; this will affect the data quality. Periodically clean the air release valve:

1. Use a 0.4 mm (0.016 inches) diameter wire (you can use #26 AWG wire) to clean the valve. The easiest way to do this is to remove the Tygon tubing above the air release valve, and use needle-nosed pliers to force the wire through the hole.
2. Blow through the air release valve to ensure it is open.
3. (if applicable) Replace the Tygon tubing above the air release valve.

Air release valve detail



0.5 mm (0.02 in.)    UP

82
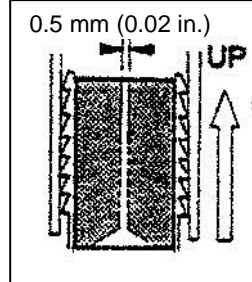
# Replacing Batteries

Alkaline D-cell
(MN1300, LR20)
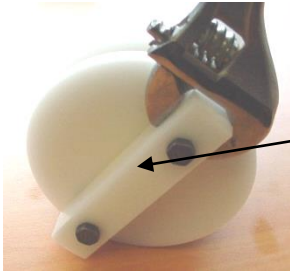
Unthread cap
by rotating
counter-
clockwise

If removing
battery pack from
housing: Unthread
post using 9/64
inch hex drive

Pull up handle, and
unthread cover by
rotating counter-
clockwise

When reinstalling
cover: tighten until
there is no gap
between bottom
O-ring and housing

Batteries
+ end up
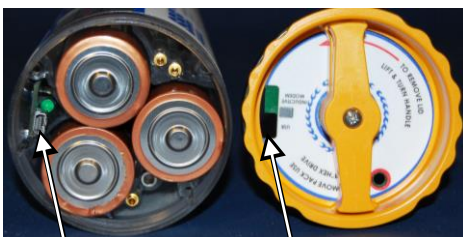
USB
connector

Opening in
cover plate for
USB connector
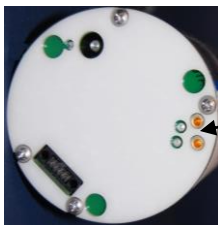
Banana plug
sockets for
checking
voltage

**CAUTION:**
**Do not use Parker O-Lube, which**
**is petroleum based; use only**
***Super* O-Lube.**

The SBE 25*plus* uses 12 alkaline D-cells (Duracell MN1300, LR20), dropped into the battery pack. Leave the batteries in place when storing the 25*plus* to prevent depletion of the back-up lithium batteries by the real-time clock. Even *exhausted* main batteries will power the clock (70 microAmps) almost indefinitely. If the 25*plus* is to be stored for long periods, **replace batteries yearly to prevent battery leakage** (which could damage the 25*plus*).

If desired, batteries can be replaced without removing the battery pack from the housing.

1. Remove the main housing battery end cap (end cap without connectors):
   A. Wipe the outside of the end cap and housing dry, being careful to remove any water at the seam between them.
   B. Unthread the end cap by rotating counter-clockwise (use a wrench on the white plastic bar if necessary).
   C. Remove any water from the O-ring mating surfaces inside the housing with a lint-free cloth or tissue.
   D. Put the end cap aside, being careful to protect the O-ring from damage or contamination.

2. If desired, remove the battery pack from the housing (not required):
   A. Use a 9/64 inch hex drive to loosen the captured post securing the battery pack in the housing.
   B. Pull the handle up and then pull the battery pack out of the housing.

3. Open the battery pack and replace the batteries:
   A. If the battery pack is still in the housing, pull the handle up.
   B. Using either the handle, or simply holding the edge of the battery pack cover, rotate the cover counter-clockwise to unthread the cover from the pack.
   C. Put the cover aside.
   D. Turn the battery pack over and remove the batteries.
   E. Install the new batteries, with the + terminals up (see the markings on the side of the battery pack).

4. Reinstall the battery pack cover:
   A. The cover plate fits into the battery pack only one way. Looking at the cover plate, note that the opening adjacent to the *USB* marking must align with the USB connector in the battery pack.
   B. **Tighten the cover until there is no gap between the bottom O-ring and the battery pack housing.**

5. If you removed the battery pack from the housing (Step 2):
   A. Check the battery voltage on the banana plug sockets on the battery pack bottom plate. It should be approximately 19.5 volts.
   B. Reinstall the battery pack in the housing.

6. Reinstall the main housing battery end cap in the 25*plus* housing:
   A. Remove any water from the O-rings and mating surfaces with a lint-free cloth or tissue. Inspect the O-rings and mating surfaces for dirt, nicks, and cuts. Clean or replace as necessary. Apply a light coat of o-ring lubricant (Parker Super O Lube) to O-rings and mating surfaces.
   B. Carefully fit the end cap into the housing and rethread the end cap into place. Use a wrench on the white plastic bar to ensure the end cap is tightly secured.

Verify that the plunger switch is pulled out to the Off position, so the SBE 25*plus* will be in quiescent (sleep) state.
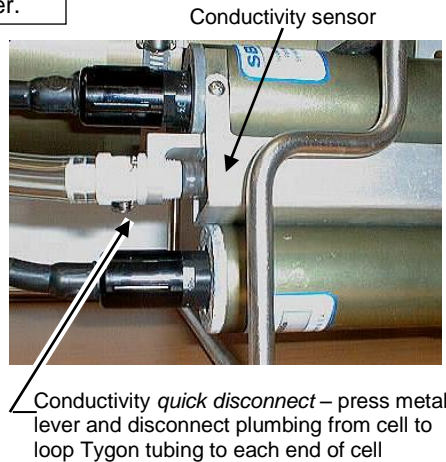
## Conductivity Cell Maintenance

**CAUTIONS:**
- **Do not put a brush or any object inside the conductivity cell to dry or clean it.** Touching and bending the electrodes can change the calibration. Large bends and movement of the electrodes can damage the cell.
- **Do not store the SBE 4 with water in the conductivity cell.** Freezing temperatures (for example, in Arctic environments or during air shipment) can break the cell if it is full of water.

The SBE 4 conductivity cell is shipped dry to prevent freezing in shipping. Refer to *Application Note 2D: Instructions for Care and Cleaning of Conductivity Cells* for conductivity cell cleaning and storage procedures and materials.

Sea-Bird supplies a conductivity cell filling and storage kit to use in rinsing and cleaning the cell. Disconnect the plumbing from the conductivity cell to use the filling and storage kit.



Conductivity sensor

Conductivity *quick disconnect* – press metal lever and disconnect plumbing from cell to loop Tygon tubing to each end of cell
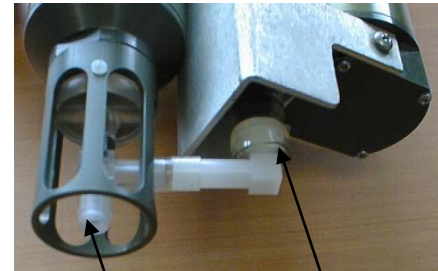
Do not remove TC Duct for rinsing, cleaning, or storage – attach Tygon tubing to end of TC Duct to loop Tygon tubing to each end of cell

Conductivity sensor

## Pressure Sensor Maintenance



Pressure sensor port

Periodically (approximately once a year) inspect the pressure port for particles and debris. Rinse the pressure port with warm, de-ionized water to remove any particles, debris, etc.

## Pump Maintenance

See *Application Note 75: Maintenance of SBE 5T, 5P, and 5M Pumps*.

## O-Ring Maintenance

**Note:**
For details on recommended practices for cleaning, handling, lubricating, and installing O-rings, see the *Basic Maintenance of Sea-Bird Equipment* module in the Sea-Bird training materials on our website.

**CAUTION:**
**Do not use Parker O-Lube, which is petroleum based; use only *Super* O-Lube.**

Recommended inspection and replacement schedule:
- For battery end cap O-rings – inspect each time you open the housing to replace the batteries; replace approximately once a year.
- For O-rings that are not normally disturbed (for example, on the connector end cap) - approximately every 3 to 5 years.

Remove any water from the O-rings and mating surfaces in the housing with a lint-free cloth or tissue. Inspect O-rings and mating surfaces for dirt, nicks, and cuts. Clean or replace as necessary. Apply a light coat of O-ring lubricant (Parker Super O Lube) to O-rings and mating surfaces.

# TC Duct Disassembly / Reassembly



Temperature sensor guard
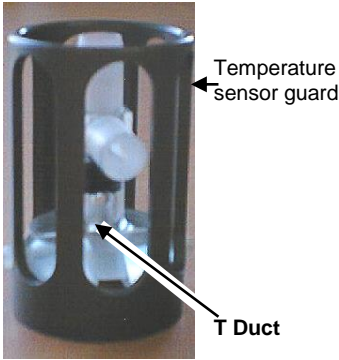
**T Duct**

The TC Duct consists of two parts:
- T Duct installed in the temperature sensor guard
- C Duct assembly



Tygon tubing

C Duct

Tygon tubing
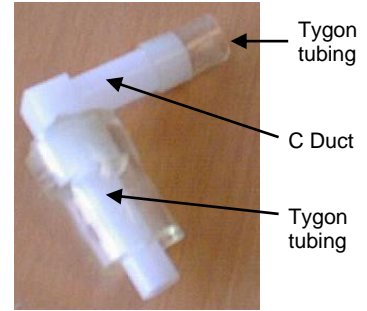
**C Duct Assembly**

**Note:**
**Photos below show conductivity and temperature sensors on an SBE 9*plus* CTD**; procedures are similar for an SBE 25*plus*.

**For sensor shipment, calibration, and acid cleaning, the C Duct must be disconnected from the T Duct and removed from the conductivity cell.** Note that the temperature sensor should be shipped with the temperature sensor guard, which includes the installed T Duct, to prevent damage to the thermistor.

These procedures require slow, deliberate movements to avoid breaking the temperature or conductivity sensor. Therefore, Sea-Bird recommends that you perform these procedures in a lab, not on deck



Conductivity sensor

Conductivity *quick disconnect* – press metal lever and disconnect plumbing from cell
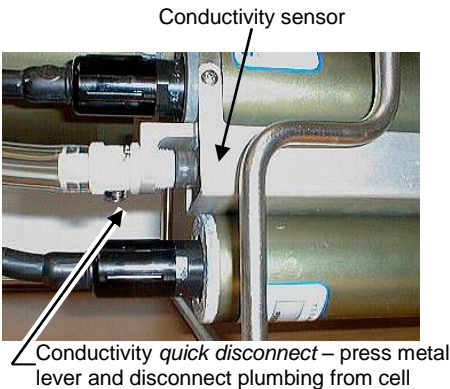
## Disassembly

1. Lay the cage on its side.

2. Remove the conductivity and temperature sensors and mounting bracket assembly from the 25*plus* main housing:
   A. Unscrew the locking sleeves from the temperature and conductivity sensor cable connectors by hand. **If you must use a wrench or pliers, be careful not to loosen the bulkhead connector instead of the locking sleeve**.
   B. Remove the cable connector from each sensor by pulling the connector firmly away from the sensor.
   C. Remove the *quick disconnect* from the conductivity cell.
   D. Remove the screws that secure the TC mounting bracket to the 25*plus* cage. Remove the mounting bracket from the cage.

3. Disconnect the T Duct from the C Duct:
   A. Slowly rotate the temperature sensor guard by turning it counter-clockwise approximately 90 degrees. At the same time, rotate the C Duct clockwise slightly. The Tygon tube that joins the ducts flexes and slips off the T Duct as the T Duct twists out of its position. **Note: Do not force the C Duct**, as this could break the conductivity cell. If rotating the C Duct is difficult, pour water over the duct area to dissolve buildup around the duct.



Conductivity sensor

Temperature sensor
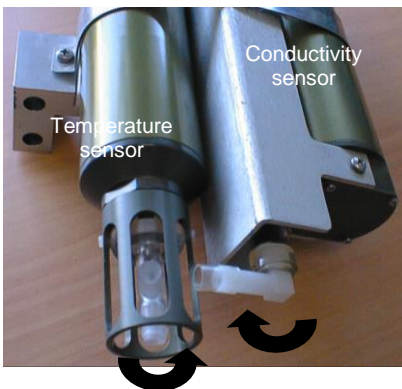
Temperature sensor guard rotated 90° counterclockwise while C Duct rotated clockwise slightly

4. Remove the C Duct from the conductivity cell:
   A. Carefully rotate the C Duct clockwise more. **Do not force the C Duct**, as this could break the conductivity cell. If rotating the C Duct is difficult, pour water over the duct area to dissolve buildup around the duct.
   B. Once loosened, gently pull the C Duct straight out from the conductivity cell.
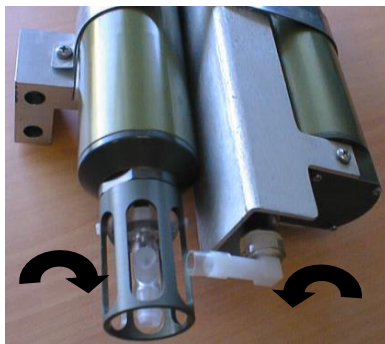   C. Store the C Duct for reinstallation.

Rotate C Duct clockwise    Pull C Duct straight out from conductivity cell

5. Retighten the temperature sensor guard by hand.

6. Loosen the mounting bracket strap screw, and slide the sensor(s) from the bracket. Note that if both sensors are being shipped, the sensors can be shipped in their mounting bracket.

Mounting strap screw

7. If shipping the sensors:
   A. Rinse the cell with clean, de-ionized water and drain. Remove larger droplets of water by blowing through the cell. **Do not use compressed air**, which typically contains oil vapor.
   B. Attach a length of Tygon tubing from one end of the conductivity cell to the other, to prevent dust and aerosols from entering the cell during shipment.

Slide sensor(s) out of bracket

## Reassembly

1. If applicable, reinstall the temperature and conductivity sensors in the mounting bracket:
   A. Slide the temperature sensor into the sensor mounting bracket until the end of the temperature sensor guard extends beyond the conductivity sensor guard approximately 29 mm (1 1/8 inch).
   B. Retighten the mounting strap screw.

29 mm (1 1/8 in.)

2. Reconnect the T Duct and C Duct:
   A. Rotate the temperature sensor guard counter-clockwise approximately 90 degrees, so the T Duct is out of the way.
   B. Install the C Duct on the conductivity cell, leaving it in the *rotated* position described in Step 3 of *Disassembly* above.
   C. Slowly rotate the temperature sensor guard back into position. As you rotate, insert the end of the T Duct into the Tygon tubing on the C Duct. Continue to rotate both the temperature sensor guard and the C Duct until the T Duct and C Duct are aligned. **Do not apply force perpendicular to the end of the glass conductivity cell, as doing so will break the cell.** If the ducts are not aligned, loosen the mounting strap screw slightly and carefully rotate / slide the temperature sensor for better alignment; retighten the screw and repeat Step C.
   D. Center the Tygon tubing on the joint between the C Duct and T Duct.

3. Reinstall the conductivity and temperature sensors and mounting bracket assembly in the 25*plus* cage:
   A. Reinstall the T & C mounting bracket in the cage.
   B. Reinstall the quick-release plug on the conductivity cell.
   C. Reinstall the cables and locking sleeves.

Rotate temperature sensor guard and C Duct into place

# Sensor Calibration

> **Note:**
> After recalibration, Sea-Bird ships the sensors back to the user with an updated configuration (.xmlcon) file and Calibration Certificates showing the new coefficients.

Sea-Bird sensors are calibrated by subjecting them to known physical conditions and measuring the sensor responses. Coefficients are then computed, which may be used with appropriate algorithms to obtain engineering units. The conductivity, temperature, and pressure sensors on the SBE 25*plus* are supplied fully calibrated, with coefficients printed on their respective Calibration Certificates (see back of manual) and stored in the instrument configuration (.xmlcon) file.

We recommend that the SBE 25*plus* be returned to Sea-Bird for calibration.

## Conductivity Sensor

The SBE 4 conductivity sensor incorporates a fixed precision resistor in parallel with the cell. When the cell is dry and in air, the sensor's electrical circuitry outputs a frequency representative of the fixed resistor. This frequency is recorded on the Calibration Certificate and should remain stable (within 1 Hz) over time.

The primary mechanism for calibration drift in conductivity sensors is the fouling of the cell by chemical or biological deposits. Fouling changes the cell geometry, resulting in a shift in cell constant.

Accordingly, the most important determinant of long-term sensor accuracy is the cleanliness of the cell. We recommend that the conductivity sensors be calibrated before and after deployment, but particularly when the cell has been exposed to contamination by oil slicks or biological material.

## Temperature Sensor

The primary source of temperature sensor calibration drift is the aging of the thermistor element. Sensor drift will usually be a few thousandths of a degree during the first year, and less in subsequent intervals. Sensor drift is not substantially dependent upon the environmental conditions of use, and — unlike platinum or copper elements — the thermistor is insensitive to shock.

## Pressure Sensor

The strain-gauge pressure sensor is capable of meeting the SBE 25*plus'* error specification with some allowance for aging and ambient-temperature induced drift.

Pressure sensors show most of their error as a linear offset from zero. A technique is provided below for making small corrections to the pressure sensor calibration using the *offset* calibration coefficient term by comparing SBE 25*plus* pressure output to readings from a barometer.

Allow the SBE 25*plus* to equilibrate (with power on) in a reasonably constant temperature environment for at least 5 hours before starting. Pressure sensors exhibit a transient change in their output in response to changes in their environmental temperature. Sea-Bird instruments are constructed to minimize this by thermally decoupling the sensor from the body of the instrument. However, there is still some residual effect; allowing the SBE 25*plus* to equilibrate before starting will provide the most accurate calibration correction.

---

**Note:**
The 25*plus* pressure sensor is an absolute sensor, so its **raw** output includes the effect of atmospheric pressure (14.7 psi). As shown on the Calibration Sheet, Sea-Bird's calibration (and resulting calibration coefficients) is in terms of psia. However, when outputting pressure in **engineering units**, Seasave outputs pressure relative to the ocean surface (i.e., at the surface the output pressure is 0 decibars). Seasave uses the following equation to convert psia to decibars:
Pressure (db) =
[pressure (psia) - 14.7] * 0.689476

---

1.  Place the 25*plus* in the orientation it will have when deployed.

2.  In Seasave, in the .xmlcon file, set the pressure offset to 0.0.

3.  Acquire data, and display the 25*plus* pressure sensor output in decibars.

4.  Compare the 25*plus* output to the reading from a good barometer at the same elevation as the 25*plus'* pressure sensor.
    Calculate *offset* = barometer reading – 25*plus* reading

5.  Enter calculated offset (positive or negative) in the .xmlcon file.

---

*Absolute* pressure measured by a barometer is 1010.50 mbar. Pressure displayed from 25*plus* is -2.5 dbars.
Convert barometer reading to dbars using the relationship: mbar * 0.01 = dbar
Barometer reading = 1010.50 mbar * 0.01 = 10.1050 dbar
Seasave outputs gage pressure, using an assumed value of 14.7 psi for atmospheric pressure. Convert SBE 25*plus* reading from gage to absolute by adding 14.7 psia to the SBE 25*plus* output:
-2.5 dbars + (14.7 psi * 0.689476 dbar/psia) = -2.5 + 10.13 = 7.635 dbars
*Offset* = 10.1050 – 7.635 = + 2.47 dbars
Enter offset in .xmlcon file.

---

For demanding applications, or where the sensor's air ambient pressure response has changed significantly, calibration using a dead-weight generator is recommended. The SBE 25*plus* end cap's 5/16-24 straight thread permits mechanical connection to the pressure source. Use a fitting that has an o-ring tapered seal, such as Swagelok-200-1-4ST, which conforms to MS16142 boss.

# Section 6: Troubleshooting

This section reviews common problems in operating the SBE 25*plus*, and provides the most likely causes and solutions.

Each SBE 25*plus* is shipped with a configuration (.xmlcon) file that matches the configuration of the instrument (number and type of auxiliary sensors, etc.) and includes the instrument calibration coefficients.

## Problem 1: Unable to Communicate with SBE 25*plus* via RS-232

The S> prompt indicates that communications between the SBE 25*plus* and computer have been established. Before proceeding with troubleshooting, attempt to establish communications again by clicking Connect in the Communications menu in Seater232 or pressing the Enter key several times.

**Cause/Solution 1**: The I/O cable connection may be loose. Check the cabling between the 25*plus* and computer for a loose connection.

**Cause/Solution 2**: The instrument type and/or its communication settings may not have been entered correctly in Seaterm232. Verify the settings in the Serial Port Configuration dialog box (Communications menu -> *Configure*). The settings should match those on the instrument Configuration Sheet.

**Cause/Solution 3**: The I/O cable may not be the correct one. The I/O cable supplied with the 25*plus* permits connection to standard 9-pin RS-232 interfaces. See *Main Housing End Cap Connectors* in *Section 2: Description of SBE 25plus* for cable pinouts.

## Problem 2: No Data Recorded

**Cause/Solution 1**: The memory may be full; once this occurs, no further data is recorded. Verify using **GetSD**; **GetSD** will show *<SamplesFree>0</SamplesFree>* if the entire memory is full. Sea-Bird recommends that you upload all previous data before beginning another deployment. Once the data is uploaded, use **DeleteAll** to reset the memory. After the memory is reset, **GetSD** will show *<Samples>0</Samples>*.

## Problem 3: Unable to View Data in Seasave (Real-Time Data Acquisition program)

**Note:**
If *Check scan length* is enabled in Seasave's Options menu, Seasave checks the data scan length against the expected length (based on the configuration file setup), and provides a warning that there is an error. Sea-Bird recommends using this setting.

**Cause/Solution 1**: If the scan length does not match the selected .xmlcon configuration file (for example, the configuration file indicates that NMEA position data is added through a NMEA device connected to the computer, but you did not connect the NMEA device to the computer), **Seasave continues with real-time data acquisition (saving the raw data to a file for later processing), but does not show or plot data in the Seasave displays.** The data quality is not affected; you can correct the configuration file error in SBE Data Processing (post-processing), and process the data as required. Verify that the settings in the configuration file match the current CTD configuration - number and type of auxiliary sensors, sensors assigned to correct channels, NMEA and Surface PAR selected if applicable.

## Problem 4: Scan Length Error in SBE Data Processing

**Note:**
To view the Diagnostics file, select *Diagnostics* in Data Conversion's Options menu. In the Diagnostics dialog box, click on *Display Log File*.

**Cause/Solution 1**: If the scan length in the data file does not match the selected .xmlcon configuration file (for example, the configuration file indicates that NMEA position data was added, but the data file does not contain NMEA data), SBE Data Processing's Data Conversion module will not process the data. If you look in the Diagnostics file, you will see a *scan length error*. Check the configuration file against the header in the data file, and correct the configuration file as required.

## Problem 5: Nonsense or Unreasonable Data

The symptom of this problem is a data file that contains nonsense values (for example, 9999.999) or unreasonable values (for example, values that are outside the expected range of the data).

**Cause/Solution 1**: A data file with nonsense values may be caused by incorrect instrument configuration:
- Bad data may be caused by incorrect setup in the 25*plus*. Send **GetCD** to verify the 25*plus* setup matches the instrument Configuration Sheet (voltage sensors assigned to correct channels, etc.).
- Bad data may be caused by incorrect configuration in the instrument .xmlcon configuration file. Verify the settings in the configuration file match the instrument Configuration Sheet.

**Note:**
Each 25*plus* is shipped with a configuration (.xmlcon) file that matches the configuration of the instrument (number and type of auxiliary sensors, etc.) and includes the instrument calibration coefficients.

**Cause/Solution 2**: A data file with unreasonable (i.e., out of the expected range) values for temperature, conductivity, etc. may be caused by incorrect calibration coefficients:
- If you are viewing real-time data in Seasave, or used Seaterm232's Upload menu to upload data and are processing the data in Seasave or SBE Data Processing - Bad data may be caused by incorrect calibration coefficients in the instrument .xmlcon configuration file. Verify the calibration coefficients in the configuration file match the instrument Calibration Certificates.
- For serial sensors interfacing to the 25*plus*: Bad data may be caused by incorrect calibration coefficients programmed into the serial sensor. Connect the sensor directly to the computer and use Seaterm232 or Seaterm to verify the calibration coefficients match the instrument Calibration Certificate.

## Problem 6: Unable to Communicate with SBE 25*plus* via Battery Pack's USB Connector

**Cause/Solution 1**: Some USB cable connectors do not seat properly on the USB port in the SBE 25*plus* battery pack, preventing communications. Because the power and ground pins are longer, the indicator light may come on even if all of the pins are not securely connected. If you cannot establish communications via the USB port (but are able to communicate via RS-232), try again with a different USB cable.

# Glossary

**Batteries –** Twelve alkaline D-cells (Duracell MN1300, LR20).

**Fouling –** Biological growth in the conductivity cell during deployment.

**PCB –** Printed Circuit Board.

> **Note:**
> All Sea-Bird software listed was designed to work with a computer running Windows XP service pack 2 or later, Windows Vista, or Windows 7 (32-bit or 64-bit).

**SBE Data Processing –** Sea-Bird's Windows data processing software, which calculates temperature, conductivity, and pressure, and derives variables such as salinity and sound velocity.

**Scan –** One data sample containing temperature, conductivity, pressure, and optional auxiliary inputs.

**Seasave V7 –** Sea-Bird's Windows software used to acquire, convert, and display real-time or archived raw (unprocessed) data.

**Seasoft V2–** Sea-Bird's complete Windows software package, which includes software for communication, real-time data acquisition, and data analysis and display. Seasoft V2 includes *SeatermV2*, *SeatermAF*, *Seasave V7*, *SBE Data Processing*.

**SeatermAF –** Sea-Bird's Windows terminal program used to communicate with the SBE 25*plus* when it is used with a water sampler operated autonomously (SBE 32 Carousel Water Sampler with Auto Fire Module (AFM) **or** SBE 55 ECO Water Sampler). See the AFM manual or SBE 55 manual.

**Seaterm V2 –** Windows terminal program *launcher*. Depending on the instrument selected, it launches Seaterm232 (RS-232 instruments), Seaterm485 (RS-485 instruments), SeatermIM (inductive modem instruments), or SeatermUSB (USB instruments).

**Seaterm232 –** Windows terminal program used with Sea-Bird instruments that communicate via an RS-232 interface, and that were developed or redesigned in 2006 and later. The common feature of these instruments is the ability to output status information in XML.

> **CAUTION:**
> **Do not use Parker O-Lube, which is petroleum based; use only** *Super* **O-Lube.**

**Super O-Lube –** Silicone lubricant used to lubricate O-rings and O-ring mating surfaces. Super O-Lube can be ordered from Sea-Bird, but should also be available locally from distributors. Super O-Lube is manufactured by Parker Hannifin (www.parker.com/ead/cm2.asp?cmid=3956).

**TCXO –** Temperature Compensated Crystal Oscillator.

**Triton X-100 –** Reagent grade non-ionic surfactant (detergent), used for cleaning the conductivity cell. Triton can be ordered from Sea-Bird, but should also be available locally from chemical supply or laboratory products companies. Triton is manufactured by Avantor Performance Materials (www.avantormaterials.com/commerce/product.aspx?id=2147509608).

# Appendix I: Functional Description and Circuitry

## Sensors and Sensor Interface

### Primary Sensors

The temperature sensor (SBE 3F) is a compact module containing a pressure-protected, high-speed thermistor and Wein bridge oscillator interface electronics. The thermistor is the variable element in the Wein bridge, while a precision Vishay resistor and two ultra-stable capacitors form the fixed components. The conductivity sensor (SBE 4) is similar in operation and configuration to the temperature sensor, except that the Wein bridge variable element is the cell resistance. The output of both of these sensors is frequency, which is measured by the 25*plus*.

Pressure is acquired by applying an AC excitation to the strain-gauge pressure bridge; a 24-bit A/D converter digitizes the output. AC excitation and ratiometric comparison avoids errors caused by parasitic thermocouples, offset voltages, leakage currents, and reference errors. A silicon diode embedded in the pressure bridge is used to measure the temperature of the bridge. This temperature is used to perform offset and span corrections on the measured pressure signal.

### Auxiliary Sensors

Optional sensors for dissolved oxygen, pH, light transmission, fluorescence, etc., generally produce slow changing voltage outputs. Signals from the auxiliary sensors are acquired using a conventional voltage-input multiplexed 16-bit A/D converter. All 8 channels are measured in the first 5 milliseconds of each scan and recorded. Depending on the number of channels enabled and the baud rate of the serial interface, the values from these channels are forwarded in real-time. The SBE 25*plus* provides four bulkhead connectors for optional auxiliary sensor inputs. Each connector provides +14.3 volts power and permits access to two differential input / low pass filtered digitizer channels.

The A/D channels are rated as Category I for attachment to non-mains derived voltage sources only. Acceptable examples include sensors powered by the SBE 9*plus*, batteries or low voltage power supplies isolated from mains voltage. Voltage input is not to exceed 5V for any time.

### CTD Channel Range, Resolution and Accuracy

The conductivity and temperature sensors produce variable frequency outputs. Period counting is used to obtain the high encoding speed and resolution required by profiling applications. Each sensor has its own counting electronics circuit, so both sensors are sampled simultaneously. Each sensor uses two 32-bit counter registers. One accumulates the number of rising edge sensor output transitions during the sample interval (50 milliseconds), and the other measures the accumulated counts of the reference clock. The frequency of the input signal is determined using the reference frequency to measure a succession of periods within the measurement window.

93

CTD resolution degrades as scan rate increases and scan window decreases. Resolution increases with faster reference frequency. Sensor resolution can be calculated as follows:

Resolution (Hz/bit) = Scan Rate * Fs/Fr

*where*

- Scan rate = 20 Hz
- Fs = sensor frequency
- Fr = CTD reference frequency (5 MHz)

At 31°C, or approximately 4 KHz, the temperature sensitivity is approximately 76 Hz/°C, resulting in the resolution:

Res (Hz/bit) = (20) * (4000/5000000) = 0.016 Hz/bit

Res (°C/bit) = 0.016 Hz/bit / 76 Hz/°C = 0.0002 °C/bit

As temperatures and frequencies decrease to typical oceanographic values, the sensitivity and resolution increases. At -1°C, the sensitivity is approximately 48 Hz/°C.

System accuracy is determined by the accuracy of the time-base used to perform the frequency counting. The SBE 25*plus* uses a TCXO to provide a low-jitter, higher-frequency reference that is stable across the operating temperature range of the instrument. This clock is stable to ±2.5ppm across -30 to +85°C, with a drift of ±1ppm per year.

### Auxiliary Voltage Sensor Range, Resolution, and Accuracy

Up to eight voltages with the range 0 to +5 volts DC can be acquired. The voltages are selected sequentially by a microcontroller, and digitized to 16 bits by successive approximation. All eight A/D channel voltages are acquired during an interval of approximately 5 milliseconds at the beginning of each scan (shortly after measurement of the frequency channels begins). Differential input amplifiers followed by 2-pole Butterworth anti-aliasing filters are used in each A/D channel.

The A/D converter input range of 0 to 5 volts is converted to digital values between 65536 and 0. The resolution at the SBE 25*plus* A/D input ports is therefore 5 volts/65536 = 0.000076 volts/bit. Special care is exercised to maintain symmetry between the differential inputs and low-pass filters to minimize the influence of temperature gradients encountered in ocean profiling.

## Real-Time Clock

To minimize power and improve clock accuracy, a temperature-compensated crystal oscillator (TCXO) is used as the real-time-clock frequency source. The TCXO is accurate to ± 1 minute per year (0 °C to 40 °C).

## Battery Wiring

The SBE 25*plus* main battery is a battery pack that holds 12 alkaline D cells (Duracell MN1300, LR20). If external power of 9 volts or higher is applied through connector J7, the SBE 25*plus* runs off of the external power, even if the main battery voltage is higher.

The SBE 25*plus* contains two ½ AA (LS14250) lithium cells (non-hazardous) which are diode OR'd with the main battery (and external power source, if used). The lithium supply is capable of supporting all 25*plus* functions and serves to permit orderly shutdown in the event of failed or exhausted main batteries. The main batteries can be changed without disturbing memory or the real-time clock. If the back-up lithium battery voltage (*vLithium* in the **GetSD** response) falls below 3.8 volts, they will not provide backup; replace the back-up batteries before they reach 3.8 volts.

The SBE 25*plus* also contains one coin cell, CR2330/1HF, which acts as a back-up to the ½ AA lithium cells to maintain the real-time clock. If the coin cell voltage (*vRTC* in the **GetSD** response) falls below 2.8 volts, it will not provide backup; replace the coin cell before it reaches 2.8 volts.

# Appendix II:
# Electronics Disassembly/Reassembly

Shown with sensors removed, but procedure does **not** always require sensor removal

Unthread cap by rotating counter-clockwise

Unthread post using 9/64 inch hex drive

Pull out handle, and pull out battery pack

Pull out and **completely** remove plunger switch (see next photo)

Plunger switch mount, with plunger switch removed
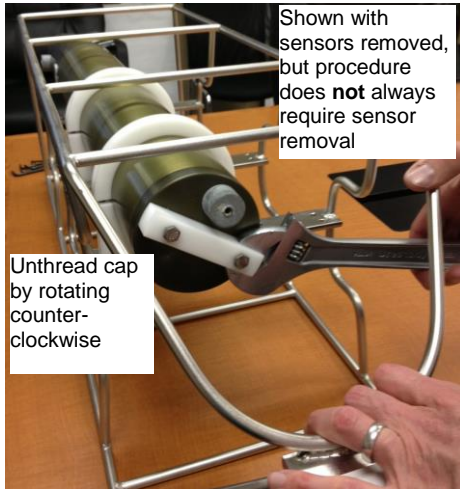
This procedure provides access to the electronics in the SBE 25*plus* main housing, for troubleshooting and repairs **in consultation with Sea-Bird**. We recommend that you leave the 25*plus* in its cage for this procedure; the cage provides more stability and a better grip when removing screws, anode, and end caps.

**As a precaution, upload all data in memory before beginning this procedure.**

**Disassembly**

1. Disconnect all cables from the 25*plus* main housing connector end cap. **Use tape to mark which cable goes to which connector.**

2. If there are any sensors mounted near the main housing connector end cap that may impede removal of the end cap, remove the sensors.

3. Remove the main housing battery end cap (end cap without connectors):
   A. Wipe the outside of the end cap and housing dry, being careful to remove any water at the seam between them.
   B. Unthread the end cap by rotating counter-clockwise (use a wrench on the white plastic bar if necessary).
   C. Remove any water from the O-ring mating surfaces inside the housing with a lint-free cloth or tissue.
   D. Put the end cap aside, being careful to protect the O-ring from damage or contamination.

4. Remove the battery pack from the housing:
   A. Use a 9/64 inch hex drive to loosen the captured post securing the battery pack in the housing.
   B. Pull out the handle on the battery pack, and then pull the battery pack out of the housing.

5. Pull the plunger switch completely out of its mount, so that it is not in the way when you remove the connector end cap.

Remove anode from connector end of housing

6. At the connector end of the main housing, use a wrench to remove the anode **on the housing** (its connection goes through both the housing and the connector end cap).

Remove screws and cone washers (3) using 5/32 inch hex drive

7. At the connector end of the main housing, use a 5/32 inch hex drive to remove the 3 screws and cone washers that connect the end cap to the housing.
   - You probably need to rotate the main housing in the cage to access all 3 screws. If any sensors are mounted to the main housing, you may need to remove them before rotating the main housing. Loosen the mount screws that hold the housing in the cage, slowly rotate the housing to the desired position, and re-tighten the mount screws. When done, loosen the mount screws, slowly rotate the main housing back to its original position, and re-tighten the mount screws.

Screw and cone washer

(shown with plunger switch removed) Grab connector end cap and rotate counter-clockwise

Plunger switch mount

8. Carefully grab the connector end cap and rotate in either direction; **do not grab the connectors or put any side force on them**. The end cap will release as the plunger switch mount rolls out of the machined arc on the housing. Then pull the end cap straight out of the housing.
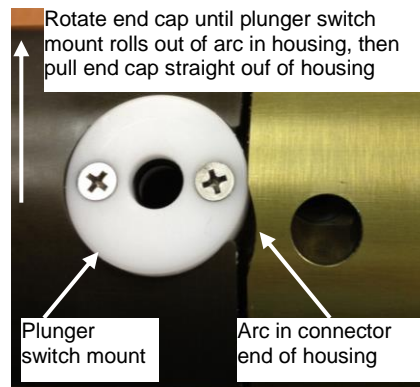
Rotate end cap until plunger switch mount rolls out of arc in housing, then pull end cap straight ouf of housing

Plunger switch mount

Arc in connector end of housing

97

**Note:**
Before delivery, a desiccant package is inserted in the housing (zip-tied to one of the 3 struts supporting the electronics assembly), and the electronics chamber is filled with dry Argon gas. These measures help prevent condensation. **If the electronics are exposed to the atmosphere, replace the desiccant package and dry gas backfill with Argon**. See *Application Note 71: Desiccant Use and Regeneration (drying)* for information on desiccant.

**CAUTION:**
**Do not use Parker O-Lube, which is petroleum based; use only** *Super* **O-Lube.**

**Reassembly**

1.  Reinstall the electronics:
    A.  Remove any water from the O-rings and mating surfaces with a lint-free cloth or tissue. Inspect the O-rings and mating surfaces for dirt, nicks, and cuts. Clean or replace as necessary. Apply a light coat of o-ring lubricant (Parker Super O Lube) to O-rings and mating surfaces.
    B.  Carefully fit the electronics in the housing, lining up the plunger switch mount with the arc on the housing. Push in the connector end cap to engage the O-rings.
    C.  At the connector end cap end of the main housing, reinstall the 3 screws and cone washers. Turn each screw sequentially until all three cone washers are seated in the housing. Seating may result in a light click. Repeat until the gap between the housing and end cap is completely closed and the screws are tight.
    D.  At the connector end cap end of the main housing, reinstall the anode.

2.  Reinstall the plunger switch in its mount.

3.  Reinstall the battery pack in the housing. Use a 9/64 inch hex drive to tighten the captured post to secure the battery pack in the housing.

4.  Reinstall the battery end cap in the 25*plus* housing:
    A.  Remove any water from the O-rings and mating surfaces with a lint-free cloth or tissue. Inspect the O-rings and mating surfaces for dirt, nicks, and cuts. Clean or replace as necessary. Apply a light coat of o-ring lubricant (Parker Super O Lube) to O-rings and mating surfaces.
    B.  Carefully fit the end cap into the housing and rethread the end cap into place. Use a wrench on the white plastic bar to ensure the end cap is tightly secured.

5.  If you removed any sensors at the start of the procedure (Step 2 in *Disassembly*), reinstall the sensors.

6.  Reconnect all cables to the connector end cap.

7.  Verify that the plunger switch is pulled out to the Off position, so the SBE 25*plus* will be in quiescent (sleep) state.

# Appendix III: Command Summary

| CATEGORY | COMMAND | DESCRIPTION |
|---|---|---|
| **Status** | **GetCD** | Display configuration (setup) data. |
| | **InitCD** | Reset all user-programmable parameters that appear in **GetCD** response to factory defaults. **Requires confirmation**. |
| | **GetSD** | Display status data. |
| | **GetCC** | Display pressure sensor calibration coefficients. |
| | **GetEC** | Display event counter data. |
| | **ResetEC** | Reset event counter |
| | **GetHD** | Display hardware data. |
| | **InitHD** | Reset all user-programmable parameters that appear in **GetHD** response to factory defaults (*Not Set*). **Requires confirmation**. |
| | **GetFiles** | Display all cast (.xml) and serial sensor (.txt) file names in memory. |
| | **GetFault** | Display fault status of all auxiliary sensor channels. |
| **General Setup** | **SetDateTime= yyyy-mm-ddThh:mm:ss** | Set real-time clock UTC date and time [yyyy = year, mm = month (01, 02, etc.), dd = day of month (01, 02, etc.), hh = hour (0-24), mm = minute, ss = second]. |
| | **SetBaudConsole=x** | **x=** baud rate (600, 1200, 2400, 4800, 9600, 14400, 19200, 38400, 57600, and 115200). *Default 9600*. |
| | **SetEchoConsole=x** | **x=Y** (*default*): Echo characters received from computer. **x=N**: Do not. |
| | **SetExecutedTag=x** | **x=Y** (*default*): Display XML Executing and Executed tags. **x=N**: Do not. |
| | **QS** | Enter quiescent (sleep) state. Memory retention unaffected. |
| **Pump Setup** | **SetMinCondFreq=x** | **x=** minimum conductivity sensor output frequency (Hz) to enable pump turn-on. *Default 3000*. |
| | **SetPumpDelay=x** | **x=** time (sec) to wait after **SetMinCondFreq=** is reached before turning pump on. Range 0 – 600 sec; *default 60 sec.* |
| **Voltage Sensor Setup** | **SetVAuxDelay0=x** | **x=** delay (sec) for supply of power to voltage channels 0 and 1 (J0 connector). Range 0 – 600 sec; *default 0*. |
| | **SetVAuxDelay1=x** | **x=** delay (sec) for supply of power to voltage channels 2 and **3** (J1 connector). Range 0 – 600 sec; *default 0*. |
| | **SetVAuxDelay2=x** | **x=** delay (sec) for supply of power to voltage channels 4 and 5 (J2 connector). Range 0 – 600 sec; *default 0*. |
| | **SetVAuxDelay3=x** | **x=** delay (sec) for supply of power to voltage channels 6 and 7 (J3 connector) and serial channels 1 and 2 (J6 connector). Range 0 – 600 sec; *default 0*. |

| CATEGORY | COMMAND | DESCRIPTION |
|---|---|---|
| **Serial Sensor Setup (#=1 or 2)** | **InitSer#** | Reset all serial sensor setup parameters to factory defaults. **Requires confirmation**. |
| | **SetEnableSer#=x** | **x=Y:** Enable measurement of serial sensor channel. **x=N (*default*):** Disable measurement of serial sensor channel. |
| | **SetNameSer#=x** | **x=** string (0-10 characters) to describe sensor on serial channel. Name becomes part of .txt file name (if **SetInlineSer#=N**), after date and time that cast started. |
| | **SetBaudSer#=x** | **x=** baud between 25*plus* and serial sensor (600, 1200, 2400, 4800, 9600, 14400, 19200, and 38400). *Default 9600*. |
| | **SetInlineSer#=x** | **x=Y (*default*):** Store serial sensor data in memory in-line with all other data (T, C, P, and auxiliary sensor voltage channels) in .xml file. **x=N:** Store serial sensor data in memory in a separate (.txt) file from all other data. |
| | **SetPromptSer#=x** | **x=** prompt (0-10 characters) that serial sensor sends to 25*plus* to indicate it is awake and ready to receive commands. Leave blank if no prompt. *Default S>.* |
| | **SetMeasStringSer#=x** | **x=** string (0-32 characters) that 25*plus* sends to serial sensor to command it to take 1 sample when 25*plus* is logging. 25*plus* sends **SetMeasStringSer#=** every **SetMeasIntervalSer#=** sec. If **SetMeasStringSer#=** (blank), no command will be sent. *Default TS.* |
| | **SetMeasIntervalSer#=x** | **x=** interval (sec) between each time 25*plus* sends command defined by **SetMeasStringSer#=** to serial sensor when 25*plus* is logging. If **SetMeasIntervalSer#=0**, 25*plus* sends command as soon as it receives reply to previous request for data. Range 0-600 sec; *default 0.* |
| | **SetStartStringSer#=x** | **x=** string (0-32 characters) 25*plus* sends to serial sensor to command it to sample autonomously when 25*plus* is logging. If **SetStartStringSer#=** (blank), no command will be sent. *Default Start.* |
| | **SetExecCharSer#=x** | **x=** decimal value (0-255) of command execution character that 25*plus* adds to end of command defined by **SetMeasStringSer#=** and **SetStartStringSer#=**. 254=carriage return line feed ('\r\n'); 255=no execution character. *Default 254*. |
| | **SetTermCharSer#=x** | **x=** decimal value (0-255) of termination character that serial sensor sends at end of reply. 254=carriage return line feed ('\r\n'); 255=no termination character. If 25*plus* does not receive termination character, it stops waiting after **SetFailoutSer#=**. *Default 254*. |
| | **SetSuppressSer#=x** | **x=Y (*default*)**: Termination character (**SetTermCharSer#=**) and prompt (**SetPromptSer#=**) sent by serial sensor to 25*plus* are not included in data file. **x=N**: Termination character and prompt are included in data file. |
| | **SetTimeoutSer#=x** | **x=** maximum time (sec) 25*plus* waits for prompt (**SetPromptSer#=**) from serial sensor, indicating serial sensor has completed processing a command. Range 1-600 sec; *default 3 sec*. |
| | **SetFailoutSer#=x** | **x=** maximum time (sec) 25*plus* waits for data from serial sensor. Range 1-600 sec; *default 60 sec*. |
| | **ToSer#=x** | **x=** command string (1-64 characters) to send to serial sensor. Can be any command recognized by serial sensor. |

| CATEGORY | COMMAND | DESCRIPTION |
|---|---|---|
| **Real-Time Output Setup** | **SetOutputFormat=x** | **x=0 (default):** Output all T, C, P data and selected auxiliary voltage sensor data in real-time, at baud set by **SetBaudConsole=**. See **SetVOut#=** commands for voltage sensor selection. For use with Seasave.<br>**x=1:** Output converted pressure and scan number at 1 Hz, at 9600 baud, regardless of **SetBaudConsole=**. For use with autonomous water sampling systems (SBE 32 Carousel Water Sampler with AFM, or SBE 55 ECO Water Sampler).<br>Note: Returns to baud set by **SetBaudConsole=** when you stop logging. |
| | **SetHistoricRate=x**<br>(applicable only if **SetBaudConsole=4800** and **SetOutputFormat=0**) | **x=0:** Slower. Output real-time data at 4 Hz if no external voltage sensors enabled for real-time output; otherwise output at 2 Hz. Use this setting if SBE 33 or 36 Deck Unit is appending NMEA and Surface PAR data.<br>**x=1:** Faster. Output real-time data at 8 Hz if no external voltage sensors enabled for real-time output; otherwise output at 4 Hz. |
| | **SetVOut0=x**<br>(if **SetOutputFormat=0**) | **x=Y:** Output data from voltage **channel 0** with real-time data.<br>**x=N:** Do not. |
| | **SetVOut1=x**<br>(if **SetOutputFormat=0**) | **x=Y:** Output data from voltage **channel 1** with real-time data.<br>**x=N:** Do no. |
| | **SetVOut2=x**<br>(if **SetOutputFormat=0**) | **x=Y:** Output data from voltage **channel 2** with real-time data.<br>**x=N:** Do not. |
| | **SetVOut3=x**<br>(if **SetOutputFormat=0**) | **x=Y:** Output data from voltage **channel 3** with real-time data.<br>**x=N:** Do not. |
| | **SetVOut4=x**<br>(if **SetOutputFormat=0**) | **x=Y:** Output data from voltage **channel 4** with real-time data.<br>**x=N:** Do not. |
| | **SetVOut5=x**<br>(if **SetOutputFormat=0**) | **x=Y:** Output data from voltage **channel 5** with real-time data.<br>**x=N:** Do not. |
| | **SetVOut6=x**<br>(if **SetOutputFormat=0**) | **x=Y:** Output data from voltage **channel 6** with real-time data.<br>**x=N:** Do not. |
| | **SetVOut7=x**<br>(if **SetOutputFormat=0**) | **x=Y:** Output data from voltage **channel 7** with real-time data.<br>**x=N:** Do not. |
| **Logging** | **StartNow** | Start logging now, storing data to memory. |
| | **Stop** | Stop logging. |
| **Data Upload and Memory Reset**<br>Stop logging before uploading. | **SetFile=x** | **x=** file index number (0 - 400) for file that you want to upload or delete (see **UploadData=** and **DeleteFile**).<br>**SetFile=** is cleared any time 1 file is deleted, all files are deleted, or a new cast is added to memory (logging is started and stopped).<br>If **x** is a large number, 25*plus* can take up to approximately 45 seconds to process command, because it must go through files in memory sequentially to assign appropriate file. To abort, click Esc key. |
| | **GetFile** | Get and display file index number and name for file selected with **SetFile=**. |
| | **UploadData=x,y** | Upload file selected with **SetFile=**, starting at character **x** (characters in each file start with 0) and uploading a total of **y** characters. Range for **x** and **y** is 0 to $(2^{32} - 1)$. |
| | **DeleteFile** | Delete file selected with **SetFile=**. **Requires confirmation.** 25*plus* updates all file index numbers larger than this one, after file is deleted. |
| | **DeleteAll** | Initialize logging - after all previous data has been uploaded, make entire memory available for recording. **DeleteAll** sets index number (cast number) to 0 (sampling will start with cast 0). **Do not send DeleteAll until all existing data has been uploaded. Requires confirmation.** |
| | **InitLogging** | Equivalent to **DeleteAll. Requires confirmation.** |

| CATEGORY | COMMAND | DESCRIPTION |
|---|---|---|
| **Testing** | **TS** | Apply power, take 1 measurement, output data in Hex (CTD and auxiliary voltage sensor data), and remove power |
| | **TSC** | Take continuous conductivity measurements and output frequency (Hz). Click Esc to stop and remove power. |
| | **TST** | Take continuous temperature measurements and output frequency (Hz). Click Esc to stop and remove power. |
| | **TSPR** | Take continuous pressure measurements and output raw pressure (counts) followed by raw pressure temperature (counts). Click Esc to stop and remove power. |
| | **TSP** | Take continuous pressure measurements and output pressure (dbars). Click Esc to stop and remove power. |
| | **TSVR** | Apply power, take continuous measurements of all auxiliary voltage channels (0–7) and output raw counts. Click Esc to stop and remove power. |
| | **TSV** | Apply power, take continuous measurements of all auxiliary voltage channels (0–7) and output voltages. Click Esc to stop and remove power. |
| | **GetVAux0** | Apply power, take continuous measurements of **voltage channels 0 & 1** (**J0** connector), and output voltages. Click Esc to stop and remove power. |
| | **GetVAux1** | Apply power, take continuous measurements of **voltage channels 2 & 3** (**J1** connector), and output voltages. Click Esc to stop and remove power. |
| | **GetVAux2** | Apply power, take continuous measurements of **voltage channels 4 & 5** (**J2** connector), and output voltages. Click Esc to stop and remove power. |
| | **GetVAux3** | Apply power, take continuous measurements of **voltage channels 6 & 7** (**J3** connector) and **serial channels 1 and 2** (**J6** connector) and output voltage data (does not output serial data). Click Esc to stop and remove power. |
| | **SetVAuxPower0=x** | **x=Y:** Apply power to **voltage channels 0 & 1** (**J0** connector). **x=N:** Remove power. |
| | **SetVAuxPower1=x** | **x=Y:** Apply power to **voltage channels 2 & 3** (**J1** connector). **x=N:** Remove power. |
| | **SetVAuxPower2=x** | **x=Y:** Apply power to **voltage channels 4 & 5** (**J2** connector). **x=N:** Remove power. |
| | **SetVAuxPower3=x** | **x=Y:** Apply power to **voltage channels 6 & 7** (**J3** connector) and to **serial channels 1 and 2** (**J6** connector). **x=N:** Remove power. |
| **Calibration Coefficients** | **SetPCalDate= yyyy-mm-dd** | Set pressure sensor calibration UTC date and time, *where* yyyy = year, mm = month (01, 02, etc.), dd = day of month (01, 02, etc.). |
| | **SetPA0=F** | **F**=pressure sensor A0. |
| | **SetPA1=F** | **F**=pressure sensor A1. |
| | **SetPA2=F** | **F**=pressure sensor A2. |
| | **SetPTempA0=F** | **F**=pressure temperature A0. |
| | **SetPTempA1=F** | **F**=pressure temperature A1. |
| | **SetPTempA2=F** | **F**=pressure temperature A2. |
| | **SetPTCA0=F** | **F**=pressure temperature compensation ptca0. |
| | **SetPTCA1=F** | **F**=pressure temperature compensation ptca1. |
| | **SetPTCA2=F** | **F**=pressure temperature compensation ptca2. |
| | **SetPTCB0=F** | **F**=pressure temperature compensation ptcb0. |
| | **SetPTCB1=F** | **F**=pressure temperature compensation ptcb1. |
| | **SetPTCB2=F** | **F**=pressure temperature compensation ptcb2. |
| | **SetPOffset=F** | **F**=pressure sensor offset correction (**decibars**). |
| | **SetPFullScale=F** | **F**=pressure sensor full scale range (**psia**); factory set. |

| CATEGORY | COMMAND AND DESCRIPTION |
|---|---|
| **Hardware Configuration** | *Temperature and Conductivity Sensor Settings - can be modified in the field to accommodate changes in T and C sensors cabled to 25plus. Limited to 11 characters; can accept '+', '-', and '+' (but '=' cannot be first character in string).*<br>**SetTempSer=**<br>**SetCondSer=**<br><br>*Auxiliary Sensor Settings- can be modified in the field to accommodate changes in auxiliary sensors cabled to 25plus. Limited to 11 characters; can accept '+', '-', and '+' (but '=' cannot be first character in string).*<br>**SetVoltType0=, SetVoltSN0=**<br>**SetVoltType1=, SetVoltSN1=**<br>**SetVoltType2=, SetVoltSN2=**<br>**SetVoltType3=, SetVoltSN3=**<br>**SetVoltType4=, SetVoltSN4=**<br>**SetVoltType5=, SetVoltSN5=**<br>**SetVoltType6=, SetVoltSN6=**<br>**SetVoltType7=, SetVoltSN7=**<br>**SetSerialTypeSer1=, SetSerialSNSer1=**<br>**SetSerialTypeSer2=, SetSerialSNSer2=** |

# Appendix IV: Integration of Serial Sensors

| Note: |
|---|
| **The 25*plus* cannot transmit real-time data from a serial output sensor.** |

This Appendix covers:
- Use of a WET Labs Triplet serial sensor with the SBE 25*plus*
- Use of a WET Labs SeaOWL serial sensor with the SBE 25*plus*
- Use of a Satlantic SUNA V2 serial sensor with the SBE 25*plus*
- General description of the serial interface, and discussion of how to set up the 25*plus* to interface with an RS-232 sensor.

## SBE 25plus Setup for Use with WET Labs Triplet

The Triplet outputs data as soon as power is applied, without the need to send it a command to take measurements. The Triplet can encompass a number of different WET Labs sensors. Different sensors have different parameters in each column, but the general output is:

| Date | <tab> | Time | <tab> | sig #1 | <tab> | sig #2 | <tab> | sig #3 | <tab> | therm | \r\n |
|---|---|---|---|---|---|---|---|---|---|---|---|

The Triplet baud rate default is 19200 (may vary), with a variable update rate based on instrument settings. Date and time are formatted DD/MM/YY and HH:MM:SS; if not available, the Triplet outputs 99/99/99 99:99:99.

Connect the Triplet directly to the computer and apply power; the Triplet should begin transmitting data immediately. Use WET Labs software to configure the Triplet as desired.

Then connect the Triplet to the SBE 25*plus*, and send the following setup commands to the 25*plus* (**#**=**1** or **2**, depending on which channel the Triplet is connected to):

| COMMAND | DESCRIPTION |
|---|---|
| **SetEnableSer#=Y** | Enable measurement of serial sensor channel. |
| **SetNameSer#= Triplet** | Describe sensor on serial channel. Name becomes part of .txt file name (if **SetInlineSer#=N**), after date and time that cast started. |
| **SetBaudSer#=** | Set serial channel to baud rate that matches Triplet baud rate. |
| **SetPromptSer#=** | Prompt that serial sensor sends to 25*plus* to indicate it is awake and ready to receive commands. Triplet has no prompt, so leave blank (hit Enter key after equal sign). |
| **SetMeasStringSer#=** | String that 25*plus* sends to serial sensor to command it to take 1 sample when 25*plus* is logging. Triplet automatically starts sampling when power is applied, so leave blank (hit Enter key after equal sign). |
| **SetMeasIntervalSer# =0** | Interval (sec) between each time 25*plus* sends command defined by **SetMeasStringSer#=** to serial sensor when 25*plus* is logging. Triplet has no measurement string, so set to 0. |
| **SetStartStringSer#=** | String that 25*plus* sends to serial sensor to command it to sample continuously when 25*plus* is logging. Triplet automatically starts sampling when power is applied, so leave blank (hit Enter key after equal sign). |
| **SetExecCharSer#= 13** | Decimal value of command termination character that 25*plus* adds to end of command defined by **SetMeasStringSer#=** and **SetStartStringSer#=**. Triplet has no measurement or start string, so termination character is not sent. |
| **SetTermCharSer#= 254** | Decimal value of termination character that Triplet sends at end of reply [254=carriage return line feed ('\r\n')]. If 25*plus* does not receive termination character, it stops waiting after **SetFailoutSer#=**. |
| **SetTimeoutSer#=4** | Maximum time (sec) 25*plus* waits for prompt (**SetPromptSer#=**) from serial sensor, indicating serial sensor has completed processing a command. With no prompt from Triplet, this applies power to Triplet 4 sec after **SetVAuxDelay3=** delay expires. |
| **SetFailoutSer#=60** | Maximum time (sec) 25*plus* waits for data from serial sensor. Set to approximately two times Triplet's update rate; too small a number will result in excess warning messages. |

The remaining serial sensor setup in the SBE 25*plus* is dependent on whether you want to store the Triplet data inline (in the 25*plus* .xml data file) or separately in a .txt file. See the appropriate section below.

## Storing Triplet Data In-Line (SetInlineSer#=Y)

When Triplet data is stored inline in the SBE 25*plus* .xml data file, the tabs separating the columns are suppressed and replaced with spaces. In the .xml file, the tab character, carriage return, and line feed are reserved. The latter two are removed from the serial stream before the data is saved. If the serial instrument output is multi-line, that aspect will be lost.

The SBE 25*plus* writes data from the serial sensor to the cast file at 16Hz. If the serial sensor updates faster, some data will be lost. If the serial sensor updates slower, serial data from the previous scan will be used until the serial sensor sends an updated string.

Send the following additional commands to set up the 25*plus* to store Triplet data in-line (**#**=**1** or **2** in commands below, depending on which channel the Triplet is connected to):

| COMMAND | DESCRIPTION |
|---|---|
| **SetNameSer#=**<br>**<user set>** | String (0-10 characters) to describe sensor on serial channel; for informational purposes only when data stored in-line. |
| **SetInlineSer#=Y** | Store serial sensor data in memory in-line with all other data (T, C, P, auxiliary sensor voltage channels) in .xml file. |
| **SetSuppressSer#=N** | Include termination character and prompt in data file. However, for this setup, there is no prompt, and termination character is carriage return line feed (which is automatically stripped out), so this parameter could be set to either Y or N. |

## Storing Triplet Data in Separate .txt File (SetInlineSer#=N)

When Triplet data is stored in a separate .txt file, the .txt file includes the data along with tabs, termination characters, and the rest.

Send the following additional commands to set up the 25*plus* to store Triplet data in a separate .txt file (**#**=**1** or **2** in commands below, depending on which channel the Triplet is connected to):

| COMMAND | DESCRIPTION |
|---|---|
| **SetNameSer#=**<br>**<user set>** | String (0-10 characters) to describe sensor on serial channel; this becomes part of .txt file name (after date and time cast started). |
| **SetInlineSer#=N** | Store serial sensor data in memory in a separate (.txt) file from all other data. |
| **SetSuppressSer#=N** | Include termination character (**SetTermCharSer#=**) and prompt (**SetPromptSer#=**) in data file. |

## SBE 25plus Setup for Use with WET Labs SeaOWL UV-A™

The SeaOWL UV-A™ outputs data as soon as power is applied, without the need to send it a command to take measurements. SeaOWL UV-A output is:

| Sig # 1 chlorophyll counts | <tab> | Sig #1 chlorophyll µg/l | <tab> | Sig #2 BB counts | <tab> | Sig #2 BB m $^{-1}$/steradian | <tab> | Sig #3 FDOM counts | <tab> | Sig #3 FDOM µg/l | <tab> | \r\n |
|---|---|---|---|---|---|---|---|---|---|---|---|---|

| *Example Output*: | | | | | |
|---|---|---|---|---|---|
| 49 | 0.000 | 49 | 2.942E-07 | 50 | 0.005 |
| 51 | 0.004 | 48 | 0 | 44 | -0.027 |
| 51 | 0.004 | 46 | -5.884E-07 | 46 | -0.016 |
| 49 | 0.000 | 46 | -5.884E-07 | 50 | 0.005 |
| 50 | 0.002 | 49 | 2.942E-07 | 48 | -0.005 |

SeaOWL UV-A baud rate default is 19200 (may vary), with a 1 Hz update rate.

Connect the SeaOWL UV-A directly to the computer and apply power; the SeaOWL UV-A should begin transmitting data immediately. Use WET Labs software to configure the SeaOWL UV-A as desired.

Then connect the SeaOWL UV-A to the SBE 25*plus*, and send the following setup commands to the 25*plus* (**#**=**1** or **2**, depending on which channel the SeaOWL UV-A is connected to):

| COMMAND | DESCRIPTION |
|---|---|
| **SetEnableSer#=Y** | Enable measurement of serial sensor channel. |
| **SetNameSer#= SeaOWLUVA** | Describe sensor on serial channel. Name becomes part of .txt file name (if **SetInlineSer#=N**), after date and time that cast started. |
| **SetBaudSer#=** | Set serial channel to baud rate that matches SeaOWL UV-A baud rate. |
| **SetInlineSer#=Y** | Store serial sensor data in memory in-line with all other data (T, C, P, and auxiliary sensor voltage channels) in .xml file. |
| **SetPromptSer#=** | Prompt that serial sensor sends to 25*plus* to indicate it is awake and ready to receive commands. SeaOWL UV-A has no prompt, so leave blank (hit Enter key after equal sign). |
| **SetMeasStringSer#=** | String that 25*plus* sends to serial sensor to command it to take 1 sample when 25*plus* is logging. SeaOWL UV-A automatically starts sampling when power is applied, so leave blank (hit Enter key after equal sign). |
| **SetMeasIntervalSer# =0** | Interval (sec) between each time 25*plus* sends command defined by **SetMeasStringSer#=** to serial sensor when 25*plus* is logging. SeaOWL UV-A has no measurement string, so set to 0. |
| **SetStartStringSer#=** | String that 25*plus* sends to serial sensor to command it to sample continuously when 25*plus* is logging. SeaOWL UV-A automatically starts sampling when power is applied, so leave blank (hit Enter key after equal sign). |
| **SetExecCharSer# =13** | Decimal value of command termination character that 25*plus* adds to end of command defined by **SetMeasStringSer#=** and **SetStartStringSer#=**. SeaOWL UV-A has no measurement or start string, so termination character is not sent. |
| **SetTermCharSer# =254** | Decimal value of termination character that SeaOWL UV-A sends at end of reply [254=carriage return line feed ('\r\n')]. If 25*plus* does not receive termination character, it stops waiting after **SetFailoutSer#=**. |
| **SetSuppressSer#=Y** | Termination character and prompt sent by serial sensor to 25*plus* are suppressed, so they do not appear in data file. |
| **SetTimeoutSer#=4** | Maximum time (sec) 25*plus* waits for prompt (**SetPromptSer#=**) from serial sensor, indicating serial sensor has completed processing a command. With no prompt from SeaOWL UV-A, this applies power to SeaOWL UV-A 4 sec after **SetVAuxDelay3=** delay expires. |
| **SetFailoutSer#=60** | Maximum time (sec) 25*plus* waits for data from serial sensor. Too small a number will result in excess warning messages. |

The remaining serial sensor setup in the SBE 25*plus* is dependent on whether you want to store the Triplet data inline (in the 25*plus* .xml data file) or separately in a .txt file. See the appropriate section below.

## Storing SeaOWL UV-A Data In-Line (SetInlineSer#=Y)

When SeaOWL UV-A data is stored inline in the SBE 25*plus* .xml data file, the tabs separating the columns are suppressed and replaced with spaces. In the .xml file, the tab character, carriage return, and line feed are reserved. The latter two are removed from the serial stream before the data is saved. If the serial instrument output is multi-line, that aspect will be lost.

The SBE 25*plus* writes data from the serial sensor to the cast file at 16Hz. Since the SeaOWL UV-A updates slower (1 Hz), serial data from the previous scan will be used until the SeaOWL UV-A sends an updated string.

Send the following additional commands to set up the 25*plus* to store SeaOWL UV-A data in-line (**#**=**1** or **2** in commands below, depending on which channel the Triplet is connected to):

| COMMAND | DESCRIPTION |
|---|---|
| **SetNameSer#=** <br> **<user set>** | String (0-10 characters) to describe sensor on serial channel; for informational purposes only when data stored in-line. |
| **SetInlineSer#=Y** | Store serial sensor data in memory in-line with all other data (T, C, P, auxiliary sensor voltage channels) in .xml file. |
| **SetSuppressSer#=N** | Include termination character and prompt in data file. However, for this setup, there is no prompt, and termination character is carriage return line feed (which is automatically stripped out), so this parameter could be set to either Y or N. |

## Storing SeaOWL UV-A Data in Separate .txt File (SetInlineSer#=N)

When SeaOWL UV-A data is stored in a separate .txt file, the .txt file includes the data along with tabs, termination characters, and the rest.

Send the following additional commands to set up the 25*plus* to store SeaOWL UV-A data in a separate .txt file (**#**=**1** or **2** in commands below, depending on which channel the SeaOWL UV-A is connected to):

| COMMAND | DESCRIPTION |
|---|---|
| **SetNameSer#=** <br> **<user set>** | String (0-10 characters) to describe sensor on serial channel; this becomes part of .txt file name (after date and time cast started). |
| **SetInlineSer#=N** | Store serial sensor data in memory in a separate (.txt) file from all other data. |
| **SetSuppressSer#=N** | Include termination character (**SetTermCharSer#=**) and prompt (**SetPromptSer#=**) in data file. |

## SBE 25*plus* Setup for Use with Satlantic SUNA V2

The SUNA V2 outputs data as soon as power is applied, without the need to send it a command to take measurements.

Connect the SUNA V2 directly to the computer and apply power.
Use UCI software to configure the SUNA V2with the following settings:
- Baud Rate: 38400
- Operational Mode: Continuous
- Logging Level: WARN

Then connect the SUNA V2 to the 25*plus*, and send the following setup commands to the 25*plus* (**#**=**1** or **2**, depending on which channel the SUNA V2 is connected to):

| COMMAND | DESCRIPTION |
|---|---|
| SetEnableSer#=Y | Enable measurement of serial sensor channel. |
| SetNameSer#= SUNAV2 | Describe sensor on serial channel. Name becomes part of .txt file name (if **SetInlineSer#=N**), after date and time that cast started. |
| SetBaudSer#=38400 | Set serial channel baud to 38400 for compatibility with SUNA V2. |
| SetInlineSer#=N | Set serial channel to store SUNA V2 data in separate .txt file. File includes data along with tabs, termination characters, etc. |
| SetPromptSer#= | Set prompt that serial sensor sends to 25*plus* to indicate it is awake and ready to receive commands. SUNA V2 has no prompt, so leave blank (hit Enter key after equal sign). |
| SetMeasStringSer#= | Set string that 25*plus* sends to serial sensor to command it to take 1 sample when 25*plus* is logging. SUNA V2 automatically starts sampling when power is applied, so leave blank (hit Enter key after equal sign). |
| SetMeasIntervalSer# =0 | Set interval (sec) between each time 25*plus* sends command defined by **SetMeasStringSer#=** to serial sensor when 25*plus* is logging. SUNA V2 has no measurement string, so set to 0. |
| SetStartStringSer#= | Set string that 25*plus* sends to serial sensor to command it to sample continuously when 25*plus* is logging. SUNA V2 automatically starts sampling when power is applied, so leave blank (hit Enter key after equal sign). |
| SetExecCharSer#= 255 | Set decimal value of command termination character that 25*plus* adds to end of command defined by **SetMeasStringSer#=** and **SetStartStringSer#=** [255=no termination character]. SUNA V2 has no measurement or start string, so termination character is not sent. |
| SetTermCharSer#= 255 | Set decimal value of termination character that SUNA V2 sends at end of reply [255=no termination character]. If 25*plus* does not receive termination character, it stops waiting after **SetFailoutSer#=**. |
| SetSuppressSer#=Y | Do not include termination character (**SetTermCharSer#=**) and prompt (**SetPromptSer#=**) in data file. |
| SetTimeoutSer#=30 | Set maximum time (sec) 25*plus* waits for prompt (**SetPromptSer#=**) from serial sensor, indicating serial sensor has completed processing a command. With no prompt from SUNA V2, this applies power to SUNA V2 30 sec after **SetVAuxDelay3=** delay expires. |
| SetFailoutSer#=60 | Set maximum time (sec) 25*plus* waits for data from serial sensor. Too small a number will result in excess warning messages. |

**Note:**
The SBE 25*plus* is typically configured by Sea-Bird to communicate at 9600 baud. The large files that the 25*plus* generates internally can take a long time to upload at that baud. If uploading data through SeatermV2, increase the 25*plus* terminal baud rate for quicker uploads. See *Command Descriptions* in *Section 4: Deploying and Operating SBE 25plus*.

### Logging and Uploading Data

SBE 25*plus* T, C, P, and voltage sensor data is stored internally as an .xml file, and output in real time as hex (raw) data. With the **SetInlineSer#=N** setting shown in the table above, SUNA V2 data is stored internally by the 25*plus* as a separate .txt file. **Data from an instrument on a 25*plus* serial channel, such as the SUNA V2, is not output in real-time by the 25*plus*.**
- SUNA V2 data can be uploaded from 25*plus* memory using SeatermV2, or can be copied from 25*plus* memory via the USB connector at the top of the 25*plus* battery pack (requires opening the 25*plus* housing).
- Alternatively, if the SUNA V2 is also logging data internally, the data can be uploaded from the SUNA V2 memory.

> **Note:**
> UCI is a separate program (i.e., not installed as part of Seasoft) included on the software CD that shipped with your SUNA. A serial cable to power and communicate directly with the SUNA V2 is required; with this cable you can modify settings in the SUNA V2, such as the baud rate, or access internally logged data directly.

## Viewing and Processing Data using UCI

### To View Data in UCI:

To view data, you need the SUNA V2 instrument package file (.xml). The SUNA V2 calibration file (.cal) is also required to display total absorbance. These files are on the Satlantic CD that came with the SUNA V2, as well as on the Sea-Bird CD-ROM if the SUNA V2 was integrated with the 25*plus* at Sea-Bird.
*Example file names: SUNA_0247_001.xml and SNA0247C.CAL*

Proceed as follows:
1. In UCI's Data menu, select SUNA, then select Replay Logged Data.
2. In the Replay Logged Data dialog:
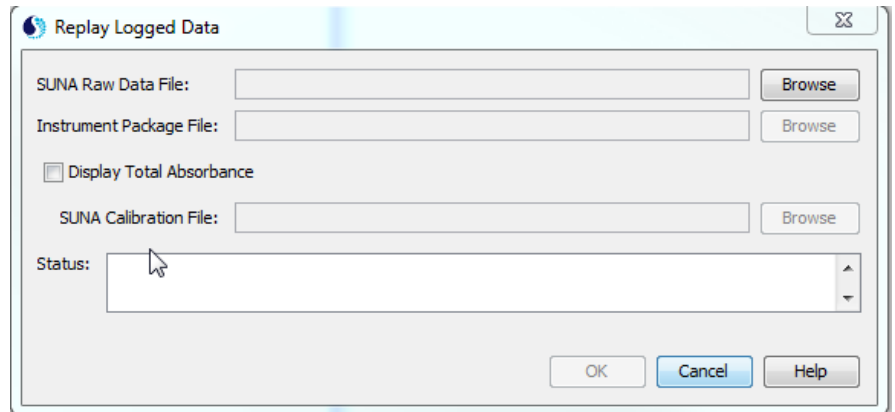


    A. SUNA Raw Data File (.txt) - Click Browse to select the.txt file (serial data file uploaded from 25*plus*). In the Select Raw File window, select *All files* for *Files of type* (this allows you to use the .txt file without changing the file extension).
    B. Instrument Package File ().xml - Click Browse to select the .xml file.
    C. If you click the check box for *Display Total Absorbance*, you need to select the SUNA V2 calibration file (.cal). Click Browse to select the .cal file.
    D. Click OK. UCI will plot the data.

**To Output Converted Data with UCI:**

To output converted data, you need the SUNA xml file, see above for information on where to locate this file. Proceed as follows:

1.  In UCI's Data menu, select SUNA, then select Convert Raw Data.

2.  In the Convert Raw Data Files to ASCII dialog:
    A.  Instrument Package File (.xml) - Click Browse to select the .xml file.
    B.  Raw Data File (.txt) - Click Browse to select the.txt file (serial data file uploaded from the 25*plus*). In the Select Raw File window, select *All files* for *Files of type* (this allows you to use the .txt file without changing the file extension). Select the desired file.
    C.  Click Options to configure the converted data as desired.
    D.  Converted Data File - Click Browse to select a location to store the output .csv file (which can be viewed with a spreadsheet program such as Excel).
    E.  Click OK once all files are selected; UCI will create the converted .csv data file.
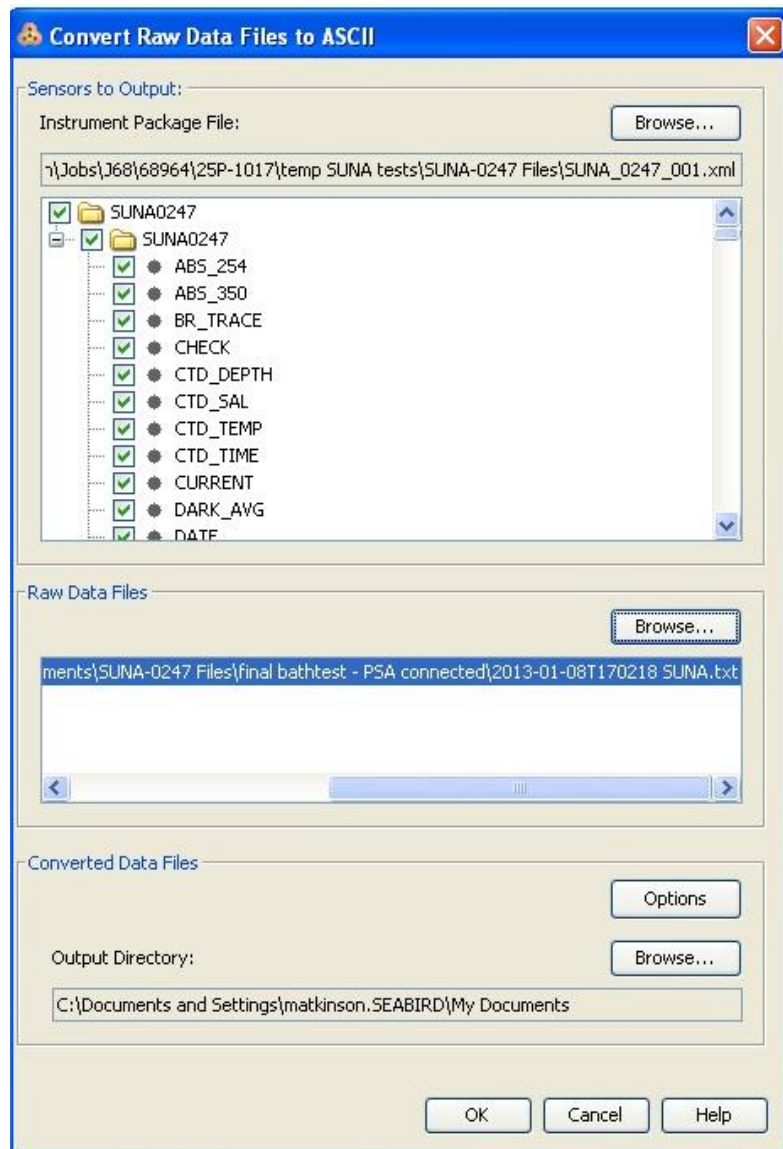
# Serial Instrument Interface

The SBE 25*plu*s can accommodate up to two serial sensors, each with individual control, sequencing, and configuration parameters. Each sensor must be compatible with RS-232 signal levels; this is a three-wire interface consisting of ground, TX, and RX with I/O directions referenced to the instrument connector.

The two sensors are named *Ser1* and *Ser2*. These are designated as **#** in the commands described in this section and in the Serial Sensor Setup Command descriptions in *Section 4: Deploying and Operating SBE 25plus*. For example, **SetEnableSer1=Y** enables acquisition of data from Ser1; **SetEnableSer2=Y** enables acquisition of data from Ser2.

The serial instrument interface is a flexible system that allows data to be collected asynchronously or on demand. Data collected from the serial instrument is not interpreted or parsed by the SBE 25*plus;* the 25*plus* stores the data as part of the instrument data (if **SetInlineSer#=Y**) or in its own file (if **SetInlineSer#=N**). In either case, the serial sensor data occupies the same non-volatile memory as the 25*plus* data, and counts against the 25*plus* available memory.

Configuring a new, third party sensor for use on the SBE 25*plus* is a task made challenging by the multitude of available sensors, their unique communication parameters, and often sparse documentation. The 25*plus* provides a number of commands for exploring serial instruments. With the capabilities in the 25*plus*, new serial instruments can be explored, and their required setup can be determined and stored. With that work done, the serial sensor can be moved between the serial channels on the 25*plus* or to other Sea-Bird instruments with a compatible serial measurement architecture.

# Exploring the Serial Interface

The key to logging serial instrument data in the SBE 25*plus* is understanding exactly which characters are sent and received in what order and when. Many times, characters are hidden from the user as unprintable in a terminal window or completely unavailable in other applications. Follow this procedure to collect the data needed to interface an unknown serial device with the 25*plus*.

1.  Connect the serial sensor to the 25*plus*. See *Main Housing End Cap Connectors* in *Section 2: Description of SBE 25plus*.

2.  Reset the serial sensor interface in the 25*plus* to factory settings with the **InitSer#** command.

3.  Set the baud rate with **SetBaudSer#=**. The other serial communication parameters are fixed at 8 data bits, 1 stop bit and no parity.

4.  If the serial instrument is externally powered, send **SetVauxPower3=Y** to power the connector to which the serial instrument is attached.

**Notes:**
- #=1 or 2, depending on which serial channel you are setting up.
- **SetVAuxPower3=Y** applies power to voltage channels 6 and 7 (J3 end cap connector) and to serial channels 1 and 2 (J6 end cap connector).

5.  After initialization (**InitSer#**), the execution and termination characters are both set to carriage return line feed (ASCII 254) and the prompt is set to S>. The execution character is sent at the end of a command to alert a serial device that a new command is available, while the termination character is used by the 25*plus* to determine if a reply is complete. The prompt is sent from the serial sensor to the 25*plus* to indicate that it is awake and ready to receive commands. If the execution and termination characters and prompt for the serial sensor are known, set them with **SetExecCharSer#=**, **SetTermCharSer#=**, and **SetPromptSer#=** respectively. The interface will seem slow if these are set incorrectly, because **SetTimeoutSer#=** and **SetFailoutSer#=** are timeouts that control how long the 25*plus* waits for a response and for data.

6.  Test your setup by sending a command to the serial sensor with the **ToSer#=** command (for example, **ToSer1=ts** sends *ts* to the serial sensor; if this is a command that the serial sensor recognizes, it will respond.)

---

*Example* (user input in bold): Reset all parameters for serial channel 1 to defaults. Then set the termination character to 13, and the prompt to S>, which are your initial guesses for these parameters. Leave the execute character at 254 (the default value):
```
S>initser1
S>settermcharser1=13
S>setpromptser1=S>
```

Send TS to the serial sensor:
```
S>toser1=ts
<Returned>
23.3456\r\nS>
</Timeout>
</Returned>
```
The reply (23.3456) indicates that the command (ts) and the execution character (default ASCII 254) were correct. And the S> at the end of the reply indicates that the prompt was set correctly. So the Timeout in the reply indicates that the termination character was incorrect (the 25*plus* never received the expected termination character - ASCII 13).

Reset the termination character, and try again:
```
S>settermcharser1=254
S>toser1=ts
<Returned>
23.3456\r\nS>
</Returned>
```
This time there was no timeout, so it appears that the termination character is correctly set.

Let's see what happens if we set the prompt incorrectly, and try again:
```
S>setpromptser1=blah
S>toser1=ts
<Returned>
23.3456\r\nS>
</Timeout>
</Returned>
```
Note that the 25*plus* received S> in the response, but not the *blah* prompt that it was expecting, so again we have a Timeout in the reply.

---

Common execute commands include linefeed (ASCII 10) or linefeed-carriage return (ASCII 254). In many serial sensors, the execute and termination character are the same. Experimenting with sending commands from a conventional terminal may be required. If the instrument was asleep, try running the command again to see if serial port traffic wakes the instrument. While experimenting, it may be worthwhile to disable any power down functions in the serial sensor.

If a string is longer than 64 characters, the logic for finding the termination is applied, however, characters beyond 64 are suppressed an event is recorded.

The prompt allows multiple lines to be recorded. A termination character may be found early in a multi-line reply, however, the string will be recorded until the termination character followed by the prompt is discovered. In these cases, the nonprinting characters (carriage return, line feed, tab) will be suppressed.
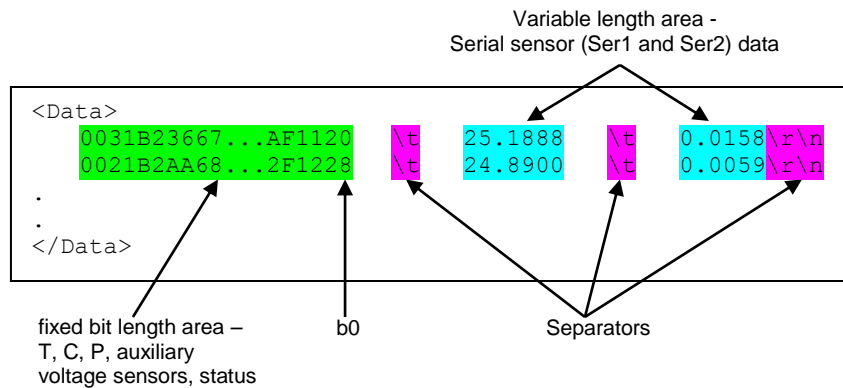
Many serial instruments will work with this call-and-response system and this is, typically, the easiest way to configure the SBE 25*plus* to control them. However, some sensors continuously output data after power is applied or a command to start sampling is sent. These can be analyzed with the same methodology, but, some careful timing may be necessary.

## Storing Data

There are two ways that the output from a serial device can be recorded based on what kind of processing the serial stream will undergo after the data is recovered:
- In-line with all other data (T, C, P, and auxiliary sensor voltage channels)
- Separate file

**When stored in-line**, the SBE 25*plus* stores the serial sensor data with the scan. This allows for the tightest integration with the 25*plus* primary sensors and the best, temporal registration. Some typical lines from an uploaded cast with external serial sensors are shown below:



Variable length area -
Serial sensor (Ser1 and Ser2) data

fixed bit length area –
T, C, P, auxiliary
voltage sensors, status

b0

Separators

The received serial string is appended to the instrument data and separated from it by the tab character (ASCII 9). The serial channels are then appended in order (Ser1 and Ser2) separated, again, by tabs. If a serial channel is disabled or no serial data has fully arrived during that scan, the space between separators contains no characters. If a serial output stream spans a scan, it is appended to the scan in which the termination character falls. If the termination character falls within 10 millisec of the end of a scan, the serial information is appended to the following scan. If the serial information is not updated when a scan is concluded, the previous serial data is used (for example, if the update rate is slower than the measurement rate).

**Storing the serial sensor data in a separate file** is useful if there are third party tools to read and analyze the recorded serial data. The file is named with the date and time of the beginning of the cast and the user-input name (**SetNameSer#=**).

# Appendix V: Integration of Auxiliary Sensors with High Current Draw

**Notes:**
- See *Specifications* for power requirements.
- See *Battery Endurance* in *Section 2: Description of SBE 25plus* for example battery endurance calculations.

The SBE 25*plus* can supply up to a total of 1.2 Amps at 12.5V to the four auxiliary voltage sensor connectors (J0, J1, J2, J3) and the auxiliary serial sensor connector (J6). This supply is not connected directly to the output pins; it goes through a current monitoring device that isolates a connector if the load, even momentarily, exceeds 1 Amp. The limit is electronic; to reset it, the connector power is cycled off and on. The limit is in place to avoid rapidly discharging the battery pack or disturbing adjacent instruments in the event of a failure or short circuit.

There are a small number of instruments that exceed the 1 Amp limit in normal operation. The Satlantic SUNA is a good example. The SUNA draws more than 1 Amp for several tens of milliseconds when the illumination source is activated approximately 5 seconds into operation. To avoid triggering the current limit, the limit can be disabled for connector J3 (voltage sensor connector) and J6 (serial sensor connector) by placing a shorting jumper into position JP5 (shown below). It is important that the SUNA or similarly high current instrument be connected to J3 or J6, as applicable, for proper operation.



Jumper in place to remove current limit on connectors J3 and J6
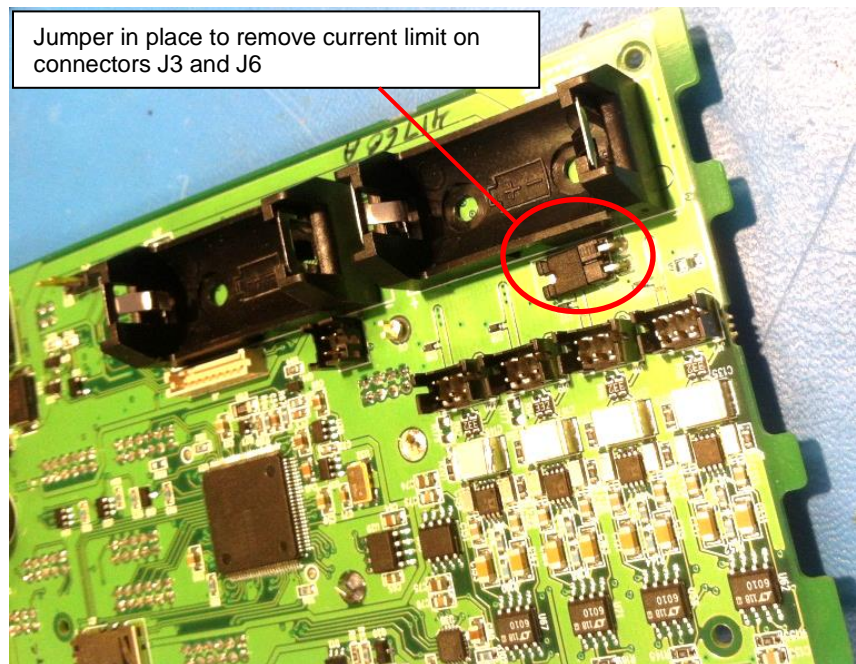
# Appendix VI: Character Map and Values

The character map is used to set the serial sensor command termination character (**SetExecCharSer#=x**) and sensor reply termination character (**SetTermCharSer#=x**), as described in *Section 4: Deploying and Operating SBE 25plus*. 254 is reserved for carriage return line feed ('\r\n'). Use the decimal value for the desired termination character in the command (for example, SetExecCharSer1=0 sets the termination character for serial sensor channel 1 to the null character).

| Dec | Hx | Oct | Char | | Dec | Hx | Oct | Html | Chr | Dec | Hx | Oct | Html | Chr | Dec | Hx | Oct | Html | Chr |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 000 | NUL | (null) | 32 | 20 | 040 | &#32; | Space | 64 | 40 | 100 | &#64; | @ | 96 | 60 | 140 | &#96; | ` |
| 1 | 1 | 001 | SOH | (start of heading) | 33 | 21 | 041 | &#33; | ! | 65 | 41 | 101 | &#65; | A | 97 | 61 | 141 | &#97; | a |
| 2 | 2 | 002 | STX | (start of text) | 34 | 22 | 042 | &#34; | " | 66 | 42 | 102 | &#66; | B | 98 | 62 | 142 | &#98; | b |
| 3 | 3 | 003 | ETX | (end of text) | 35 | 23 | 043 | &#35; | # | 67 | 43 | 103 | &#67; | C | 99 | 63 | 143 | &#99; | c |
| 4 | 4 | 004 | EOT | (end of transmission) | 36 | 24 | 044 | &#36; | $ | 68 | 44 | 104 | &#68; | D | 100 | 64 | 144 | &#100; | d |
| 5 | 5 | 005 | ENQ | (enquiry) | 37 | 25 | 045 | &#37; | % | 69 | 45 | 105 | &#69; | E | 101 | 65 | 145 | &#101; | e |
| 6 | 6 | 006 | ACK | (acknowledge) | 38 | 26 | 046 | &#38; | & | 70 | 46 | 106 | &#70; | F | 102 | 66 | 146 | &#102; | f |
| 7 | 7 | 007 | BEL | (bell) | 39 | 27 | 047 | &#39; | ' | 71 | 47 | 107 | &#71; | G | 103 | 67 | 147 | &#103; | g |
| 8 | 8 | 010 | BS | (backspace) | 40 | 28 | 050 | &#40; | ( | 72 | 48 | 110 | &#72; | H | 104 | 68 | 150 | &#104; | h |
| 9 | 9 | 011 | TAB | (horizontal tab) | 41 | 29 | 051 | &#41; | ) | 73 | 49 | 111 | &#73; | I | 105 | 69 | 151 | &#105; | i |
| 10 | A | 012 | LF | (NL line feed, new line) | 42 | 2A | 052 | &#42; | * | 74 | 4A | 112 | &#74; | J | 106 | 6A | 152 | &#106; | j |
| 11 | B | 013 | VT | (vertical tab) | 43 | 2B | 053 | &#43; | + | 75 | 4B | 113 | &#75; | K | 107 | 6B | 153 | &#107; | k |
| 12 | C | 014 | FF | (NP form feed, new page) | 44 | 2C | 054 | &#44; | , | 76 | 4C | 114 | &#76; | L | 108 | 6C | 154 | &#108; | l |
| 13 | D | 015 | CR | (carriage return) | 45 | 2D | 055 | &#45; | - | 77 | 4D | 115 | &#77; | M | 109 | 6D | 155 | &#109; | m |
| 14 | E | 016 | SO | (shift out) | 46 | 2E | 056 | &#46; | . | 78 | 4E | 116 | &#78; | N | 110 | 6E | 156 | &#110; | n |
| 15 | F | 017 | SI | (shift in) | 47 | 2F | 057 | &#47; | / | 79 | 4F | 117 | &#79; | O | 111 | 6F | 157 | &#111; | o |
| 16 | 10 | 020 | DLE | (data link escape) | 48 | 30 | 060 | &#48; | 0 | 80 | 50 | 120 | &#80; | P | 112 | 70 | 160 | &#112; | p |
| 17 | 11 | 021 | DC1 | (device control 1) | 49 | 31 | 061 | &#49; | 1 | 81 | 51 | 121 | &#81; | Q | 113 | 71 | 161 | &#113; | q |
| 18 | 12 | 022 | DC2 | (device control 2) | 50 | 32 | 062 | &#50; | 2 | 82 | 52 | 122 | &#82; | R | 114 | 72 | 162 | &#114; | r |
| 19 | 13 | 023 | DC3 | (device control 3) | 51 | 33 | 063 | &#51; | 3 | 83 | 53 | 123 | &#83; | S | 115 | 73 | 163 | &#115; | s |
| 20 | 14 | 024 | DC4 | (device control 4) | 52 | 34 | 064 | &#52; | 4 | 84 | 54 | 124 | &#84; | T | 116 | 74 | 164 | &#116; | t |
| 21 | 15 | 025 | NAK | (negative acknowledge) | 53 | 35 | 065 | &#53; | 5 | 85 | 55 | 125 | &#85; | U | 117 | 75 | 165 | &#117; | u |
| 22 | 16 | 026 | SYN | (synchronous idle) | 54 | 36 | 066 | &#54; | 6 | 86 | 56 | 126 | &#86; | V | 118 | 76 | 166 | &#118; | v |
| 23 | 17 | 027 | ETB | (end of trans. block) | 55 | 37 | 067 | &#55; | 7 | 87 | 57 | 127 | &#87; | W | 119 | 77 | 167 | &#119; | w |
| 24 | 18 | 030 | CAN | (cancel) | 56 | 38 | 070 | &#56; | 8 | 88 | 58 | 130 | &#88; | X | 120 | 78 | 170 | &#120; | x |
| 25 | 19 | 031 | EM | (end of medium) | 57 | 39 | 071 | &#57; | 9 | 89 | 59 | 131 | &#89; | Y | 121 | 79 | 171 | &#121; | y |
| 26 | 1A | 032 | SUB | (substitute) | 58 | 3A | 072 | &#58; | : | 90 | 5A | 132 | &#90; | Z | 122 | 7A | 172 | &#122; | z |
| 27 | 1B | 033 | ESC | (escape) | 59 | 3B | 073 | &#59; | ; | 91 | 5B | 133 | &#91; | [ | 123 | 7B | 173 | &#123; | { |
| 28 | 1C | 034 | FS | (file separator) | 60 | 3C | 074 | &#60; | < | 92 | 5C | 134 | &#92; | \ | 124 | 7C | 174 | &#124; | | |
| 29 | 1D | 035 | GS | (group separator) | 61 | 3D | 075 | &#61; | = | 93 | 5D | 135 | &#93; | ] | 125 | 7D | 175 | &#125; | } |
| 30 | 1E | 036 | RS | (record separator) | 62 | 3E | 076 | &#62; | > | 94 | 5E | 136 | &#94; | ^ | 126 | 7E | 176 | &#126; | ~ |
| 31 | 1F | 037 | US | (unit separator) | 63 | 3F | 077 | &#63; | ? | 95 | 5F | 137 | &#95; | _ | 127 | 7F | 177 | &#127; | DEL |

Source: www.LookupTables.com

## Extended ASCII Codes

| | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 128 | Ç | 144 | É | 160 | á | 176 | ░ | 192 | └ | 208 | ╨ | 224 | α | 240 | ≡ |
| 129 | ü | 145 | æ | 161 | í | 177 | ▒ | 193 | ┴ | 209 | ╤ | 225 | ß | 241 | ± |
| 130 | é | 146 | Æ | 162 | ó | 178 | ▓ | 194 | ┬ | 210 | ╥ | 226 | Γ | 242 | ≥ |
| 131 | â | 147 | ô | 163 | ú | 179 | │ | 195 | ├ | 211 | ╙ | 227 | π | 243 | ≤ |
| 132 | ä | 148 | ö | 164 | ñ | 180 | ┤ | 196 | ─ | 212 | ╘ | 228 | Σ | 244 | ⌠ |
| 133 | à | 149 | ò | 165 | Ñ | 181 | ╡ | 197 | ┼ | 213 | ╒ | 229 | σ | 245 | ⌡ |
| 134 | å | 150 | û | 166 | ª | 182 | ╢ | 198 | ╞ | 214 | ╓ | 230 | µ | 246 | ÷ |
| 135 | ç | 151 | ù | 167 | º | 183 | ╖ | 199 | ╟ | 215 | ╫ | 231 | τ | 247 | ≈ |
| 136 | ê | 152 | ÿ | 168 | ¿ | 184 | ╕ | 200 | ╚ | 216 | ╪ | 232 | Φ | 248 | ° |
| 137 | ë | 153 | Ö | 169 | ⌐ | 185 | ╣ | 201 | ╔ | 217 | ┘ | 233 | Θ | 249 | ∙ |
| 138 | è | 154 | Ü | 170 | ¬ | 186 | ║ | 202 | ╩ | 218 | ┌ | 234 | Ω | 250 | · |
| 139 | ï | 155 | ¢ | 171 | ½ | 187 | ╗ | 203 | ╦ | 219 | ■ | 235 | δ | 251 | √ |
| 140 | î | 156 | £ | 172 | ¼ | 188 | ╝ | 204 | ╠ | 220 | ■ | 236 | ∞ | 252 | ⁿ |
| 141 | ì | 157 | ¥ | 173 | ¡ | 189 | ╜ | 205 | ═ | 221 | ▌ | 237 | φ | 253 | ² |
| 142 | Ä | 158 | ₧ | 174 | « | 190 | ╛ | 206 | ╬ | 222 | ▐ | 238 | ε | 254 | ■ |
| 143 | Å | 159 | ƒ | 175 | » | 191 | ┐ | 207 | ╧ | 223 | ▀ | 239 | ∩ | 255 | |

Source: www.LookupTables.com

# Appendix VII: Replacement Parts

| Part Number | Part | Application Description | Quantity in 25*plus* |
|---|---|---|---|
| 22018 | Alkaline D-cell battery, Duracell MN 1300 (LR20) | Power SBE 25*plus* | 12 |
| 802113 | Battery pack (batteries **not** included) | Holds alkaline batteries | 1 |
| 22107 | 3.6V lithium battery, ½ AA, LS14250 | Back-up lithium cells | 2 |
| 22086 | Lithium battery, coin type with tabs, CR2330/1HF | Back up lithium cell for real-time clock | 1 |
| 30411 | Triton X-100 | Octyl Phenol Ethoxylate – Reagent grade non-ionic cleaning solution for conductivity cell (supplied in 100% strength; dilute as directed) | 1 |
| 30388 | Tygon tube, ½ inch ID x ¾ inch OD | Main plumbing tubing | - |
| 30579 | Tygon tube, 3/8 inch ID x ½ inch OD | 13 mm (0.5 inch) long pieces used on SBE 43 intake and exhaust to fit to main plumbing | - |
| 171888 | 25-pin DB-25S to 9-pin DB-9P cable adapter | For use with computer with DB-25 connector | 1 |
| 50062 | Pump O-ring kit | Includes:<br>• 30010 Pump thrust washer (insulates each end of impeller against pump shaft)<br>• 30095 Parker 2-002N674-70 (pump impeller retainer)<br>• 30571 Parker 2-124N674-70 (pump head to impeller housing)<br>• 30082 Parker 2-213N1000-70 (end cap to housing, 5T pump with retaining ring)<br>• 31011Parker 2-026 N674-70 (end cap to housing, 5T pump without retaining ring) | - |
| 50391 | Pump impeller replacement kit | Includes:<br>• 30009 Pump impeller<br>• 30010 Pump thrust washer (insulates each end of impeller against pump shaft)<br>• 30095 Parker 2-002N674-70 (pump impeller retainer)<br>• 30571 Parker 2-124N674-70 (pump head to impeller housing)<br>• 30082 Parker 2-213N1000-70 (end cap to housing, 5T pump with retaining ring)<br>• 31011Parker 2-026 N674-70 (end cap to housing, 5T pump without retaining ring) | - |

*Continued on next page*

*Continued from previous page*

| Part Number | Part | Application Description | Quantity in 25*plus* |
|---|---|---|---|
| 50507 | Small hardware and o-ring kit | O-rings and hardware, including:<br>• 50087 TC Duct cell filling & storage kit<br>• 30377 Bolt, ¼-20 x 1-1/4 socket SS (secures mount blocks to cage)<br>• 30254 Washer, ¼" split ring lock (for 30377)<br>• 234556 Screw insulator<br>• 311146 Machine screw, ¼-28 x ¾ flat head socket, SS (secures connector end cap to housing)<br>• 30816 Parker 2-234E603-70 (battery end cap to housing piston seal)<br>• 30090 Parker 2-153N674-70 (battery end cap to housing face seal)<br>• 30072 Parker 2-017N674-70, XSG / AG bulkhead connector O-ring<br>• 30325 Parker 2-017N674-70, MCBH bulkhead connector O-ring | |
| 50509 | Sealogger CTD Support Kit, SBE 25*plus* with **XSG / AG connectors** | Includes cables, connectors, dummy plugs, maintenance supplies, and other mechanical spares:<br>• 17043 Locking sleeve (secure cables & dummy plugs)<br>• 17045 3-pin RMG-3FS dummy plug<br>• 17046 4-pin RMG-4FS dummy plug<br>• 17047 6-pin AG-206 dummy plug<br>• 17653 3-pin XSG-3-BCL-HP-SS bulkhead connector<br>• 17628 6-pin AG-306-HP-SS bulkhead connector<br>• 30044 Seacat anode, 1"<br>• 30384 Tygon tubing, 7/16 x 9/16 (conductivity soaker hose)<br>• 30388 Tygon tubing, ¾' x ½" (main CTD plumbing)<br>• 30409 Teflon tape, adhesive 1 side (insulation for metal parts to hose clamps)<br>• 30411 Triton X-100 (conductivity cell cleaning solution)<br>• 30457 Parker Super 'O' Lube (silicone-based o-ring lubricant)<br>• 23155.1 Plunger switch assembly<br>• 50507 SBE 25*plus* small hardware & o-ring kit (see above)<br>• 17086 3-pin RMG-3FS to 3-pin RMG-3FS cable, 0.6 m (2.1 ft) long (temperature & conductivity sensor cables)<br>• 801385 4-pin RMG-4FS to 9-pin DB-9S and power leads, 2.4 m (8 ft) data I/O cable (for bench use, from 25*plus* J7 Y-cable 17709 to computer)<br>• 801380 4-pin RMG-4FS to 9-pin DB-9S shielded I/O cable, 20 m (66 ft) long (for real-time logging, from 25*plus* J7 Y-cable 17709 to computer)<br>• 17709 6-pin AG-206 to 2-pin RMG-2FS and 4-pin RMG-4MP Pump / Data I/O Y-cable (from 25*plus* J7 to pump (2-pin) and data I/O (4-pin) | |

*Continued from previous page*

| Part Number | Part | Application Description | Quantity in 25*plus* |
|---|---|---|---|
| 50508 | Sealogger CTD Support Kit, SBE 25*plus* with **wet-pluggable connectors** | Includes cables, connectors, dummy plugs, maintenance supplies, and other mechanical spares:<br>• 171192 Locking sleeve (secure cables & dummy plugs)<br>• 171500 3-pin MCDC-3-F dummy plug<br>• 171398 4-pin MCDC-4-F dummy plugs<br>• 171498 6-pin MCDC-6-F dummy plug<br>• 172024 3-pin MCBH-3MP(WB) bulkhead connector<br>• 172026 6-pin MCBH-6MP(WB) bulkhead connector<br>• 30044 Seacat anode, 1"<br>• 30384 Tygon tubing, 7/16 x 9/16 (conductivity soaker hose)<br>• 30388 Tygon tubing, ¾' x ½" (main CTD plumbing)<br>• 30409 Teflon tape, adhesive 1 side (insulation, metal parts to hose clamps)<br>• 30411 Triton X-100 (conductivity cell cleaning solution)<br>• 30457 Parker Super 'O' Lube (silicone-based o-ring lubricant)<br>• 23155.1 Plunger switch assembly<br>• 50507 SBE 25*plus* small hardware & o-ring kit (see above)<br>• 171669 3-pin MCIL-3FS to 3-pin MCIL-3FS cable, 0.7 m (2.5 ft) (temperature & conductivity sensor cables)<br>• 801374 4-pin MCIL-4FS to 9-pin DB-9S, 2.4 m (8 ft) data I/O cable (for bench use, from 25*plus* J7 Y-cable 171883 to computer)<br>• 801421 4-pin MCIL-4FS to 9-pin DB-9S, I/O cable, 20 m (66 ft) (for real-time logging, from 25*plus* J7 Y-cable 171883 to computer)<br>• 171883 6-pin MCIL-6FS to 2-pin MCIL-2FS and 4-pin MCIL-4MP Pump / Data I/O Y-cable (from 25*plus* J7 to pump (2-pin) and data I/O (4-pin) | - |

# Appendix VIII: Manual Revision History

| Manual Version | Date | Description |
|---|---|---|
| 001 | 04/12 | • Initial release of production units. |
| 002 | 05/12 | • Seasave and SBE Data Processing changes that affect documentation: SBE 25plus Real-Time Options Configuration dialog changed.<br>• Update schematics showing integration with SBE 33 and 36 Deck Units; Surface PAR channels in Deck Units now standard (was optional).<br>• Correct end cap drawing (Serial Inputs 1 and 2 were reversed). |
| 003 | 05/13 | • Add quiescent current specification if powering externally.<br>• Derate Amp-hour rating for use with high current draw auxiliary sensors; add example in Battery Endurance calculations.<br>• Add cable wiring diagrams.<br>• Add TC duct disassembly/reassembly procedure in *Section 5: Routine Maintenance and Calibration*.<br>• Add information on use with Satlantic SUNA V2 in *Appendix IV: Integration of Serial Sensors*.<br>• Add information on StripNullChars.exe, to remove null characters from an uploaded data file that includes data from RS-232 serial sensors.<br>• Add *Appendix V: Integration of Auxiliary Sensors with High Current Draw*.<br>• Add information on sensors and sensor interface in *Appendix I: Functional Description and Circuitry*.<br>• Add disassembly/reassembly to access electronics in *Appendix II: Electronics Disassembly/Reassembly*.<br>• Add information about limitations with 115200 baud rate.<br>• Add note about seating of USB cable on internal connector in battery pack.<br>• Add more information on file manipulation when communicating via USB.<br>• Update software compatibility information.<br>• Update screen captures for Seaterm232 Serial Port Configuration dialog box.<br>• Update Upload Data dialog box, and add information on Header Form tab.<br>• Update diagram for integration with SBE 55.<br>• Fix typos. |
| 004 | 02/15 | • Correct memory specification – holds 1320 hours of data from T,C,P, 8 voltage channels .<br>• Clarify serial sensor command nomenclature; for example, change **SetEnable#=** to **SetEnableSer#=**, so that # only describes channel (1 or 2) instead of string (Ser1 or Ser2).<br>• Add to specifications: sampling rate and more weight information.<br>• Mention in more places that serial sensor data is stored in memory but not output in real time.<br>• Add information that diagnostic data is stored in memory but not output in real time.<br>• Provide more information on converting raw data.<br>• Add information on string length and character types allowed for **SetVoltType#=** commands.<br>• Correct description of GetVAux3 output (does not output serial data).<br>• Update Declaration of Conformity<br>• Add O-ring maintenance section.<br>• Add caution on using spray can lubricants on MCBH connectors.<br>• Add caution on using Parker Super O-lube, not Parker O-lube (which is petroleum based).<br>• Update pressure fitting size in section on performing pressure calibration.<br>• Remove *standard* and *optional* language related to instrument features.<br>• Update language on where to find updated software on website.<br>• Use new Sea-Bird Scientific manual cover template.<br>• Fix typos. |

*Continued on next page*

*Continued from previous page*

| 005 | 06/16 | • Add information to *Appendix IV: Integration of Serial Sensors* on use with WET Labs SeaOWL UV-A$^{TM}$ in-situ oil-in-water sensor with RS-232 output.<br>• Update software configuration file description for SBE Data Processing to show selection for WET Labs SeaOWL UV-A$^{TM}$, which was added in Seasave and SBE Data Processing 7.26.0.<br>• Change information on baud rate required when using with WET Labs Triplet; baud may vary.<br>• Document **SetTempSer=** and **SetCondSer=** in Hardware Configuration command list, allowing customers to set temperature and conductivity sensor serial numbers if they replace the sensors.<br>• Update Declaration of Conformity.<br>• Update SeatermV2 main screen.<br>• Update cable part number in 50509 kit.<br>• Fix typos. |
|---|---|---|
| 006 | 05/17 | • Replace information on SUNACom software with UCI software. |

# Index