



KONGSBERG

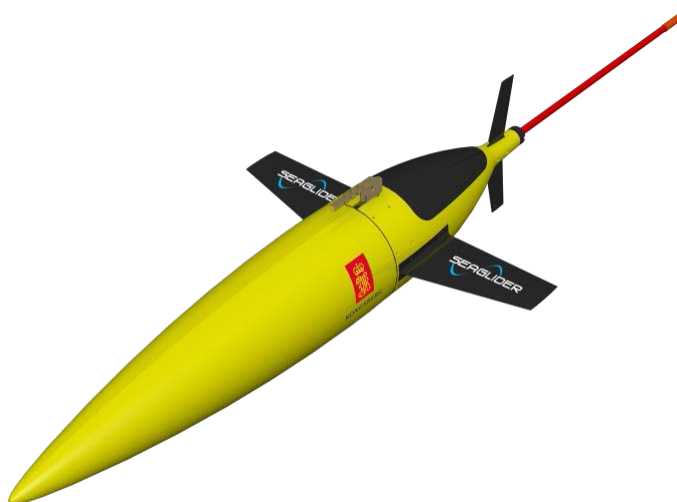
# *Seaglider Serdev User Guide*

***Document Number: 4900001***

## Summary

The Seaglider *Serdev* interface is for simple serial sensors. This document is intended as:

- A guide for integrating new serial sensors with the Seaglider,
- A reference for adding and removing serial sensors from the Seaglider, and
- A reference for installing new versions of Seaglider software and serial sensor configuration files.



Kongsberg Underwater Technology, Inc.

19210 33rd Avenue West, WA 98036-4707 Lynnwood United States of America

Telephone +1 425 712 1107 Telefax +1 425 712 1197 <http://www.km.kongsberg.com>

Enterprise number 760687050

The information contained in this document may be subject to change at a later date (due, for instance, to availability of components). Notice will be given only in case such a change is deemed to be of any consequence for customers. Kongsberg Maritime AS shall not be liable for incidental or consequential damages in connection with the furnishing, performance, or use of this document.

© 2014 Kongsberg Maritime AS. All rights reserved. No part of this work covered by the copyright hereon may be reproduced or otherwise copied without prior permission from Kongsberg Maritime AS.

## Table of contents

<b>1</b>	<b>OVERVIEW .....</b>	<b>5</b>
1.1	Serial Sensors .....	5
1.2	Serdev .....	5
<b>2</b>	<b>CNF FILE .....</b>	<b>6</b>
2.1	CNF Files.....	6
2.2	Fields .....	6
2.3	Scale and Offset.....	9
<b>3</b>	<b>GETTING STARTED WITH A NEW DEVICE.....</b>	<b>11</b>
3.1	Connection.....	11
3.2	Write Simple CNF File.....	12
3.3	Load CNF and Configure Device.....	14
<b>4</b>	<b>USE MENU SYSTEM TO ACCESS DEVICE .....</b>	<b>15</b>
4.1.1	Try Simple Device Commands.....	18
<b>5</b>	<b>EDITING THE CNF FILE.....</b>	<b>19</b>
5.1	Reloading the CNF File.....	19
<b>6</b>	<b>SCIENCE FILE.....</b>	<b>20</b>
<b>7</b>	<b>EXAMPLE – PAR SENSOR.....</b>	<b>22</b>
<b>8</b>	<b>EXAMPLE – WET LABS ECO PUCK .....</b>	<b>24</b>
<b>9</b>	<b>EXAMPLE – CONTROS OPTODE .....</b>	<b>26</b>
<b>10</b>	<b>REMAINING TASKS.....</b>	<b>28</b>
10.1	Put .cnf on Basestation .....	28
10.2	Update Science File .....	28
10.3	Test Launch .....	28
10.4	Data into .netCDF File .....	28
10.5	Calibration Data.....	29
10.6	Plotting .....	29

<b>APPENDIX A: LOAD CNF FILE .....</b>	<b>30</b>
A.1 Check for Existing Filename .....	30
A.2 Transfer CNF file to Seaglider .....	30
A.3 Strip CNF file .....	31
A.4 Load CNF into Sensor Library .....	31
<b>APPENDIX B: CONFIGURE DEVICE .....</b>	<b>35</b>
B.1 Check Existing Configuration .....	35
B.2 Know What Port to Use .....	36
B.3 Add New Serial Sensor .....	36
B.4 Save Changes .....	39
<b>APPENDIX C: REMOVE DEVICE.....</b>	<b>40</b>
<b>APPENDIX D: UNLOAD CNF FILE .....</b>	<b>42</b>
<b>APPENDIX E: RELOADING THE CNF FILE .....</b>	<b>45</b>

# 1 OVERVIEW

This document describes Seaglider *serdev* interface.

## 1.1 Serial Sensors

For the purpose of a Seaglider, serial sensors are defined by set behaviors. When powered on, serial sensors take a reading and produce a line of output on the serial port. Serial sensors typically do not have the ability to store multiple readings in built-in, on-board memory. Seaglider typically powers on these *serdev* sensors, takes a single sample, and then powers them off according to the sampling schedule defined in the Seaglider *science* file.

Serial sensors thus differ from another class of sensors, often called loggers or logger devices. Logger devices are powered on and collect data in on-board memory until subsequently powered off. The data logged in memory is retrieved from the logger sensor at some later point in time, e.g. after Seaglider returns to the surface. For information on integrating a logger sensor on Seaglider please refer to the Seaglider Logdev User's Guide.

## 1.2 Serdev

The Seaglider *serdev* interface provides software support for serial devices on Seaglider. It allows the user to add a new serial sensor to the Seaglider without changing the Seaglider code base. An ASCII configuration file (.cnf file) specifies the device configuration and access to be described to the Seaglider run-time software without requiring source code changes.

## 2 CNF FILE

This section introduces the various fields in a *serdev* .cnf file and how they work.

### 2.1 CNF Files

A Seaglider *serdev* .cnf file is an ASCII file that describes how Seaglider should configure and access a specific serial sensor.

The filename must conform to the DOS 8.3 file naming convention. The file name and extension must be lowercase. By convention, Seaglider .cnf files have a ‘.cnf’ extension. If a .cnf file does not conform to the 8.3 rules, it will not load properly on Seaglider.

The .cnf file must be copied to the Seaglider and added to the sensor library; instructions for doing this are given in the Appendices.

The .cnf file must also exist on the basestation so that the sensor data sent to the basestation by the Seaglider can be extracted into the Seaglider data files. Kongsberg basestation software includes .cnf files for Kongsberg supported serial sensors. If you are creating your own .cnf file, you must load the .cnf file on the basestation. See 10.1 for instructions on how to do this.

### 2.2 Fields

The *serdev* .cnf file has the following fields:

name=	Name of the serial sensor. This name will appear in the Seaglider menu system for device configuration. Maximum of 15 characters. Can be mixed-case, but by convention, is typically all uppercase. This name should be unique and reflect the sensor manufacturer name and the sensor model number. Required field.  Example: name=WL_BBFL2IRB
prefix=	The prefix string is written into the data files generated by Seaglider to identify data collected by this sensor. Each column name (see column= field below) is prefixed by this string, and separated from this string by a period. Maximum of 15 characters. May be mixed-case, but by convention, is typically all lowercase. This name should reflect the sensor manufacturer name and the sensor model number if

	<p>possible. Required field.</p> <p>Example: prefix=bbfl2irb</p> <p>Example:</p> <p>If the .cnf file contains ‘prefix=ctd’, ‘column=cond’, and ‘column=temp’, then the resultant columns in the Seaglider data files will have the names ctd.cond and ctd.temp.</p>
baud=	<p>Baud rate to use for serial communications with the device. Typically one of the following values: 2400, 4800, 9600, 19200, 38400, 57600. Required field.</p> <p>Example: baud=9600</p>
warmup=	<p>Specifies how long the Seaglider software should wait (milliseconds) after powering on the device before trying to access the device, e.g. by sending the query string. Integer value. Typical values include 10, 50, 200, 500, 2000, etc. Required field.</p> <p>Example: warmup=2000</p>
timeout=	<p>Specifies how long Seaglider software should wait (milliseconds) for a response from the device after sending the query string. Integer value. Required field.</p> <p>Example: timeout=500</p>
voltage=	<p>Voltage (10 or 24) required by the device. This value is used when calculating battery usage. Required field.</p> <p><b>Note</b>     <i>Use voltage=10 for 15V Seagliders.</i></p> <p>Example: voltage=10</p>
current=	<p>Current (amps) drawn by the device when powered on. This value is used when calculating battery usage. Required field.</p> <p>Example: current=0.5</p>
format=	<p>Format of the data lines returned by the instrument. The format string can be a maximum of 256 characters. Required field.</p> <p>If the format string contains spaces, the entire format string should be enclosed in double quotes.</p> <p>The format string may contain special sequences. These sequences describe how to read and parse data from the device output string. The format string special sequences are described after the example.</p>

	<p>Example: format=" %00 %01 %02"</p> <p>The above format string indicates that the Seaglider expects 3 numeric values from each line of output from the device. The three values will be associated with the first 3 column= definitions.</p> <p>Special Sequences:</p> <p>%00, %01, %02, ... - read a number from the response string and associate it with column 0, 1, 2, etc.</p> <p>%[n]00, %[n]01, %[n]02 – read ‘n’ digits from the response string and associate the result with column 0, 1, 2, etc.</p> <p>%f – read and skip over a floating point value</p> <p>%d – read and skip over an integer value</p> <p>%s – read and skip over string data up to next white space</p>
column=	<p>The column field identifies one of the fields of data returned by the instrument as specified in the format command. This field can appear multiple times in the .cnf file. Use a name that closely matches the type of data that was collected. The column name will be appended to the prefix field as discussed in the prefix= section of this table.</p> <p>Column names can be up to 64 characters. At least one column name is required. Up to 7 column names can be specified.</p> <p>The column= field also provides a way to specify scale and offset values to be applied to the collected data. The syntax is to immediately follow the column name with (scale, offset), where both scale and offset are integer values. If not specified, a scale of 1 and an offset of 0 will be applied.</p> <p>Example: column=PARuV(1000000,0)</p>
query=	<p>String that Seaglider should send to the device after powerup and timeout to cause the device to output a sample. Maximum of 256 characters. Optional field. If not provided, the sensor must automatically transmit data when powered on.</p> <p>If the query string contains spaces, the entire query string should be enclosed in double quotes.</p> <p>The query string may contain special sequences. These sequences will be interpreted and expanded before the query string is sent to the</p>



	<p>device. The query string special sequences are described after the examples.</p> <p>Example: query=sample%r</p> <p>Example: query="do sample%r"</p> <p>Example: query=%Fget%[done%r]</p> <p>Special Sequences:</p> <p>%1 ... %9 – DelayMilliSecs(10) .. DelayMilliSecs(90)</p> <p>%r – replaced by &lt;CR&gt; (13)</p> <p>%n – replaced by &lt;LF&gt; (10)</p> <p>%b – replaced by &lt;BREAK&gt;</p> <p>%% - allows a '%' character to be included in the query string</p> <p>%e – replaced by &lt;ESCAPE&gt; (27)</p> <p>%#xx – send hex byte xx - must specify both characters</p> <p>%F – flush any data previously sent by the sensor but not yet read by the Seaglider</p> <p>%[] – wait for a sequence of characters. Characters between the [] pair can include any ASCII character. To wait for '%' use '%%'. To wait for ']' use '%]'. Use %r to wait for &lt;CR&gt;. Use %n to wait for &lt;LF&gt;. Use %#xx to wait for arbitrary bytes.</p>
--	--

## 2.3 Scale and Offset

The *serdev* .cnf file 'column' fields allow the user to specify a scale and offset value to be applied to data destined for that column.

***When should you set scale and offset to something other than their default values of 1 and 0?***

Use `scale` to help preserve precision when floating point values are read from a sensor using the `%00`, `%01`, etc. format string substitution sequences. Seaglider transmits serial sensor data in long integer format. Suppose your sensor returns a field of data with three decimal digits of precision (ex. 249.038). If you do nothing, then the default scale value of 1 will be applied, and the value 249 will end up in the data files on the basestation. If you instead set `scale` to 1000, then the data value 249.038 will be *multiplied* by 1000 before the data is sent, and 249038 will be transmitted to the basestation. Since you also put a copy of `.cnf` file on the basestation (as described in Section 10.1), the basestation knows to convert the 249038 back to 249.038 before outputting the final value into the `.eng` file on the basestation.

Use `offset` when a sensor data sheet reports that a field has had a constant value included in the reported output value. The `offset` value you specify can then ‘undo’ this change to obtain the true value of the field. Seaglider *subtracts* the `offset` value you specify (or 0, the default) from the value read in the sensor data line before applying the `scale` factor (or 1, the default) as specified above.

## 3 GETTING STARTED WITH A NEW DEVICE

*Suppose your team has a new serial sensor that needs to be integrated onto your Seaglider. Where do you begin?*

### 3.1 Connection

The first step is to get the device connected to Seaglider so you can experiment with it. Once connected, you'll be able to send commands to the device via a serial connection and see the device respond to your commands.

Start by locating the user documentation for the device (printed manual, online manual, etc.). Make sure the device will operate on either 10V or 24V unregulated battery power if your Seaglider is a 10V/24V system or 15V unregulated battery power if your glider is a 15V system. Make sure the device is a typical serial device, not a frequency mode or logger device.

Work with your team to connect the sensor to your Seaglider. The default end-cap port configuration is provided in the Table 3-1 below. Kongsberg Seagliders are shipped with an endcap port map that shows exactly how your Seaglider is wired between the endcap and aft terminal board. If previous modifications to your Seaglider have been performed, hopefully your team has kept good records. Otherwise determining the end-cap port configuration may require partial disassembly of your vehicle.

**Note** *If you have a Seaglider manufactured by University of Washington or iRobot, consult with Kongsberg Seaglider Support, as we may have archive records of how the vehicle was originally configured.*

**Table 3-1 Typical Kongsberg Seaglider endcap port map.**

Endcap port	Aft term board label	Port number	Nominally	Voltage	Freq. Mode?
C	TPU 6/7	3	“Optics 1”	10   15	
D or Zinc	TPU 8/9	4	“O2”	10   15	Yes
B	MUX0	5	“Optics 2”	10   15	
E	MUX1	6	“aux 1”	10 or 24   15	

Endcap port	Aft term board label	Port number	Nominally	Voltage	Freq. Mode?
F	MUX2	7	“aux 2”	10 or 24   15	

Connect the new serial sensor to the endcap port using an IE55 cable. Cables are typically built into the device, or provided by the sensor manufacturer. You may have to procure one.

Remember both the port number and ‘nominally’ name; you will need these when configuring a device that uses your .cnf file.

**Warning:** *Most 10V/24V Seagliders are configured to output 10V only on all ports. If your Seaglider was not built by Kongsberg and/or this history is unknown, measure the voltage before connecting the new serial device. Contact Kongsberg if you wish to change the output voltage on Ports E or F.*

### 3.2 Write Simple CNF File

Once the device is connected you need to tell Seaglider a little bit about the device so that Seaglider can access it. Write a very simple .cnf file to do this.

Use a UNIX or Linux system to create the .cnf file so that it will have the right line-ending characters for the Seaglider. If you must use Windows for this then be sure to use an editor that understands Linux line-endings, such as WordPad, TextPad, or gedit.

Your simple .cnf file needs to have just a dozen lines or so. Here is an example:

```
name=MYDEV
prefix=mydev
baud=9600
timeout=2000
warmup=500
voltage=10
current=.018
format="%00 %01"
column=col1
column=col2
query="do sample%r"
```

The name field is the name of your new device. This name will appear in the Seaglider menu system when you are working with the device. Replace MYDEV with a name that makes sense for your new serial sensor.

The prefix field will be combined with each column field to generate the column names for your data in the dive data generated by Seaglider and delivered to the basestation. Replace 'mydev' with a prefix that reflects the data delivered by your sensor.

The baud field gives the baud rate of the logger device. If the baud rate of your new sensor is something other than 9600 then change this field as needed.

The timeout and warmup fields come from your device's documentation. Warmup is how long it takes for the sensor to power on and become ready to process commands, take samples, or generate output. Timeout is how long it takes for the already powered-on sensor to produce a sample. Both values are in milliseconds.

The voltage field is based on the historic 10V/24V Seaglider systems. These systems estimate Amp-hours used on the two batteries separately. ***15V Seaglider should always use voltage=10 in the .cnf file.***

Current is in amps. Current is used for estimating the battery power consumed by the device. Change this value to match your device documentation.

The format field describes what the sensor's output looks like. Your sensor's documentation probably shows an example of the expected output. Use that to make an educated guess for the format string (see the table in the previous section; you can skip over fields, etc.). You can adjust the .cnf file after you have loaded the .cnf file onto the glider, configured the device, and powered it on, and confirmed the serial device output.

For each value you intend to read with your format string, you should have one 'column' field to describe that data. Thus, if you have %00 and %01 in your format string, you should have two 'column=' fields defined. Don't worry about scale and offset yet.

If your device needs to receive a command from you in order to generate output, then use the 'query' field to specify this command. The query field should be EXACTLY what you the user would type if you were talking directly to the device – therefore, do not forget the <CR> you would type to complete the command and send it off to the device.

Save the file as 'mydev.cnf' (again, replace 'mydev' with something resembling the name field in your .cnf) and exit the text editor. Reminder – the filename must conform to that old DOS 8.3 file naming scheme.

### **3.3 Load CNF and Configure Device**

Now you need to transfer your file to the Seaglider, strip it, load it into the serial device library, and configure a device that uses the .cnf file. Appendices A and B cover the required steps in detail. Once these steps have been completed you should be able to access the device using the Seaglider menu system as described in the next section.

## 4 USE MENU SYSTEM TO ACCESS DEVICE

With your sensor configured you can now try to use the Seaglider menu system to access the new device and continue development and debug of your CNF file.

Start by returning to the Seaglider main menu:

```
----- Main Menu -----
  1 [param ] Parameters and configuration
  2 [hw    ] Hardware tests and monitoring
  3 [modes ] Test operation modes and files
  4 [pdos  ] PicoDOS commands (and exit)
  5 [launch] Pre-launch
Enter selection (1-5,CR): 2
```

Make choice 2 [hw]. You should now see the Hardware menu:

```
----- Hardware Menu -----
*Motors and VBD
  1 [pitch ] Pitch control
  2 [roll  ] Roll control
  3 [vbd   ] VBD control
  4 [pressure] Pressure sensor
  5 [compass] Compass (tcm2)
  6 [gps   ] GPS
  7 [modem ] Modem (xmodem mode)
  8 [intpress] Internal pressure
  9 [altim ] Altimeter
 10 [sensors] Sensors
 11 [loggers] Loggers
*Other
 12 [batt  ] Batteries and fuel gauges
 13 [lowlevel] Low-level hardware (IO,A-D,CF)
 14 [misc  ] Miscellaneous (travel, timeouts, date/time)
 15 [develop] Developer tests
  CR) Return to previous
Enter selection (1-15,CR): 10
```

Make choice 10 [sensors] to do hardware testing on sensors. You should see the Sensors menu, and a list of serial sensors that you can test:

```
----- Sensors menu -----
  1 [qsp  ] QSP2150
    CR) Return to previous
Enter selection (1-1,CR): 1
```

***What if your sensor doesn't show up in the list?*** First, check to make sure you are looking at the Sensors menu as opposed to the Loggers menu (choice 11). Also make sure you configured the sensor properly using the instructions in Appendices A and B. Did you enter 'Y' to 're-initialize the hardware configuration' after adding the device (Appendix B). Does the sensor show up in the serlib serial sensor library?

Assume you just configured the QSP2150 and want to test it out. The sensor is included in the list, so enter choice 1 to select it. You should then see the serial device test menu:

```
----- serial device test menu -----
  1 [selftest] Basic self-test
  2 [direct  ] Direct comm with unit
  3 [capture ] Capture serial output
  4 [config  ] Show configuration
  5 [edit    ] Edit configuration
    CR) Return to previous
Enter selection (1-5,CR):
```

This is the menu you will use to check if your serial device is responding, and if it is properly generating and reporting samples. You can use it now, you can use it when bench-testing, and you can use it just prior to sending the Seaglider on real dives at sea.

To quickly check out your device, invoke the [selftest] option by entering choice 1:

```
Enter selection (1-5,CR): 1
668.400,HQSP2150,N,PARuV:9950
```

This [selftest] reads one sample from the device and displays the sample and associated data column name/s from the .cnf file. This shows whether or not the device is responding to power on and any query commands and whether or not the Seaglider can receive data from the device. It shows whether you are getting the expected data from the device, and whether the 'format' string is correct.

Another good test is the [direct] communications test, choice 2. This test puts you in direct communication with the device, just as if you had hooked up the sensor to a terminal emulator program (e.g. TeraTerm or minicom), as shown in the next example.



```
Enter selection (1-5,CR): 2
616.873,SUSR,N,Starting breakable loop - Ctrl-Q to end
.009951, 19.79, 9.383
.009954, 21.04, 9.383
.009950, 21.04, 9.404
.009950, 21.04, 9.383
.009951, 21.04, 9.383
.009952, 21.04, 9.383
.009946, 21.04, 9.404
.009950, 21.04, 9.383
.009952, 21.04, 9.383
.009950, 21.04, 9.404
.009948, 21.04, 9.383
.009950, 21.04, 9.383
.009949, 21.04, 9.383
.009946, 21.04, 9.383
.009948, 21.04, 9.404
.009946, 21.
```

Here, the device just outputs samples one line after another. Eventually the user enters Cntrl-Q to end the direct communications session. Many sensors require a <CR> or a command such as 's', 'sample', or 'do sample' before they generate a line of output – refer to your sensor's documentation to see if this is required, and if so, include the 'query' field in your .cnf file.

Direct mode is also helpful when configuring sensors. Your sensor documentation may list commands that can be used to alter the output format, number of fields, sample frequency, etc. Those commands can be entered here, along with any command to save the settings changes you made.

If direct mode produces garbled output you might have the baud rate set incorrectly in the .cnf file.

**Note**            *Depending on the device, you may need to turn on local echo in your terminal emulator in order to see the commands that you type.*

Another useful testing option is the [config] menu. This shows the device configuration defined by the .cnf file.

```
Enter selection (1-5,CR): 4
646.765,HQSP2150,N,name = QSP2150
646.845,HQSP2150,N,prefix = qsp
646.922,HQSP2150,N,baud = 9600
646.998,HQSP2150,N,warmup = 100
647.076,HQSP2150,N,timeout = 250
647.184,HQSP2150,N,current = 0.0044
647.267,HQSP2150,N,voltage = 10
647.342,HQSP2150,N,format = %00
647.452,HQSP2150,N,query = nil
647.580,HQSP2150,N,column = PARuV(1000000.000000,0.000000)
```

This option allows you to review the .cnf file settings. It allows you to double-check your results. The output shows that ‘format’ is only expecting one value, and there is only one ‘column’ definition. Recall that the [selftest] output above only produced a single output. Now refer back to the [direct] mode output – it produced three values per line. To get those other values, you would have to update the ‘format’ string and add more ‘column’ definitions.

#### 4.1.1 Try Simple Device Commands

Use the [hw] → [sensors] → [<your device>] menu and get back into direct mode. Refer to the device’s documentation. Enter commands and verify that you get output back from the device. The characters you enter as commands should be echoed correctly, and the device should respond as if you were talking to it via minicom.

## 5 EDITING THE CNF FILE

Now that you know how to send commands to the device and see output from the device, edit the 'format' and 'column' fields in the .cnf file until the [selftest] command returns the device data in the desired format.

### 5.1 Reloading the CNF File

Each time you update the .cnf and want to test your changes, you will need to:

1. Remove the device (Appendix C),
2. Unload the .cnf file (Appendix D),
3. Load the updated CNF file (Appendix A),
4. Reconfigure the device (Appendix B), and
5. Test the results (Chapter 4).

Appendix E provides a quick summary of all the steps involved. You will be able to do this without referring to the instructions after the first handful of .cnf file edits.

## 6 SCIENCE FILE

The Seaglider science file controls when serial sensor data is collected. Each line in the science file describes a 'depth bin'. There are two bins in the following file – one for depths ranging from 0-30 meters, and one for depths between 30 meters and 150 meters. Each line describes the sampling behaviour that should occur when the glider is within the depth limits of that bin.

If a Seaglider is at a depth greater than the last bin, it will use the sampling behavior defined in the last bin.

A sample science file follows:

/depth	time	sample	gcint
30	5	111	60
150	5	122	120

The first field in the science file is the depth limit of the bin (meters).

The second field in the science file is the time interval (seconds) between samples. The practical minimum time interval is 5 seconds.

The third field in the science file is the sensor sampling multiplier, a value in the range 0-9, inclusive.

The fourth field in the science file is the Guidance & Control interval (seconds).

The sampling multiplier field (field 3) requires additional explanation. There should be one digit in this field for each serial sensor on the Seaglider. In this case, there are 3 digits in the field, so there appear to be three serial sensors installed on the Seaglider. If a fourth serial sensor were installed in the final serial sensor slot, a fourth digit would need to be added to this field for both the 30 and 150 m depth lines.

The first digit in the sampling multiplier field is for the sensor in serial slot 1. The second digit is for the sensor in serial slot 2, the third for serial slot 3, and the fourth (when present) is for serial slot 4. If necessary, you can use the Seaglider menu system to choose [param] → [config] → [show] to view the sensor configuration and see which sensor is in which serial slot number. For users updating software, refer back to old Self-Test capture files (.cap files).

The sensor sampling multiplier is applied to the second field (time interval) to determine when a sensor should be sampled. In this example we see that above 30m in depth we sample all the sensors at the same rate (every 5 seconds). For depths from 30m to 150m we continue to sample the sensor in slot 1 every 5 seconds, but we use the sampling multiplier to sample the sensors in slots 2 and 3 every 10 seconds.

The science file should be put into the Seaglider's home directory on the basestation. The glider will copy the science file from the basestation before each dive.

## 7 EXAMPLE – PAR SENSOR

In this section the .cnf file required to support the Biospherical Instruments Inc. QSP2150 PAR (Photosynthetically Active Radiation) sensor is presented and discussed.

The .cnf file for this sensor is called qsp2150.cnf. The contents are as follows:

```
name=QSP2150
prefix=qsp
timeout=250
baud=9600
warmup=100
voltage=10
current=0.00438
format="%00"
column=PARuV(1000000,0)
```

The fields above are described in Section 2.2.

The ‘name=’ field will appear in the Seaglider menu system, so be sure to choose a name that describes the identity and purpose of your sensor so that it will be easy to identify. Perhaps a better name could have been chosen here – the name does not indicate that this is a PAR sensor, nor does it indicate that it is from Biospherical Instruments. A better choice might have been PAR\_QSP2150 or BI\_PAR\_QSP2150.

The ‘prefix=’ field and each ‘column=’ field will be combined into data column names in the Seaglider data files. Here again a better choice of string could have been made. Perhaps prefix could have been ‘par’, and column could have been ‘uV’?

The timeout, baud, warmup, voltage and current fields come directly from the documentation provided with your sensor. If necessary, contact the sensor manufacturer to get any of these values that are not apparent from the documentation. The current field is used in battery gauge calculations, so make a good effort to provide an accurate value here. You may find it necessary to have an electrical engineer measure the current while you send the Seaglider on a test launch (from the main menu, [launch] → [test]).

The ‘format=’ field is determined from the device documentation. You may also need to use the Seaglider menu system to interact with the device in ‘direct’ mode to see the exact layout of the device output strings ([hw] → [sensors] → [<select device>] → [direct]). The QSP2150 outputs three numeric values on a single line after each reading, for example:

```
.009953, 23.53, 9.404
```

The first value is the uV value that we want, the others can be ignored. Therefore our format string just needs to be “%00”.

The column field includes a (scale,offset) specifier. The Seaglider saves and transmits data as long integer values, so you have to arrange to scale the data read from the device so as to not lose decimal precision. Our example uV reading was .009953, so by setting the scale to 1000000, the reading will be transmitted to the basestation as 9953. The basestation will ‘unscale’ the value back to .009953 using the copy of the .cnf file on the basestation.

**Note**            *Users creating their own .cnf files must load the .cnf file on the basestation. Refer to Section 10.1 for more information.*

*Users loading .cnf files for software upgrade are not required to put .cnf files on the basestation. The Kongsberg basestation software comes preloaded with all .cnf files for Kongsberg supported sensors.*

Notice that there is no ‘query=’ field in the file. This sensor will output a reading after powering on without requiring a command from the controlling user (or program).

## 8 EXAMPLE – WET LABS ECO PUCK

In this section the .cnf file required to support the WET Labs BBFL2 ECO Puck sensor is presented and discussed.

The .cnf file is named wlbfl2.cnf. The contents are as follows:

```
name=WL_BBFL2
prefix=wlbfl2
timeout=2500
baud=4800
warmup=1200
voltage=10
current=0.105
format="%00 %01 %02 %03 %04 %05 %06"
column=BB1ref
column=BB1sig
column=FL1ref
column=FL1sig
column=FL2ref
column=FL2sig
column=temp
```

The fields above are described in Section 2.2.

The ‘name=’ field will appear in the Seaglider menu system. Choose a name that describes the identity and purpose of your sensor and is easy to identify. Here the name includes an indication of the manufacturer (WL) as well as the sensor model number (BBFL2), so it should be easy to see within the menu system.

The ‘prefix=’ value is combined with each ‘column=’ value to generate the column names for the data in the Seaglider data files. The wlbfl2 will result in data file column names wlbfl2.BB1ref, wlbfl2.BB1sig, wlbfl2.FL1ref, etc.

The timeout, baud, warmup, voltage and current fields come directly from the documentation provided with your sensor. If necessary, contact the sensor manufacturer directly to get any of these values that are not apparent from the documentation. The current field is used in battery gauge calculations to make a good effort to provide an accurate value here. You may find it necessary to have an electrical engineer measure the current while you send the Seaglider on a test launch (in the menu system, [launch] → [test]).



The 'format=' field describes the data that is to be read from the sensor output line. This value is determined from the device documentation. You may also need to use the Seaglider menu system to interact with the device in 'direct' mode to see the exact layout of the device output strings ([hw] →[sensors] →[<select device>]→[direct]). The BBFL2 outputs seven fields after every reading, and we want to save all of them, so %00 through %06 are used (to skip a field, use %f for a float, %d for an integer, %s for a string). See Chapter 9 for an example of when this might be necessary.

The 'column=' fields each describe a column of data to be saved by Seaglider. The sensor only outputs integer data, so no scale value is required to preserve decimal precision.

On power-up, the sensor automatically takes a reading and outputs a result. Thus, no 'query=' field is required.

## 9 EXAMPLE – CONTROS OPTODE

In this section the .cnf file required to support the Contros Optode sensor is presented and discussed.

The .cnf file is named contopt.cnf. The contents are as follows:

```
name=CONTOPT
prefix=contopt
timeout=2500
baud=57600
warmup=1000
voltage=10
current=0.300
format="%d,%d,%s,%s,%s,%00,%01,%02"
column=Dphi
column=AirSat
column=Temp
query="%[CODT1,]"
```

The fields above are described in Section 2.2.

The ‘name=’ field will appear in the Seaglider menu system, so be sure to choose a name that describes the identity and purpose of your sensor and is easy to identify. Here the name is simply part of the manufacturer's name (CONT) as well as an abbreviation of the sensor type (OPT), so it should be easy to see within the menu system.

The ‘prefix=’ value is combined with each ‘column=’ value to generate the column names for the data in the Seaglider data files. The "contopt" value will result in data file column names contopt.Dphi, contopt.AirSat, and contopt.Temp.

The timeout, baud, warmup, voltage and current fields come directly from the documentation provided with the sensor.

The ‘format=’ field describes the data that is to be read from the sensor output line. This value is determined from the device documentation. You may also need to use the Seaglider menu system to interact with the device in ‘direct’ mode to see the exact layout of the device output strings ([hw]→[sensors]→[<select device>]→[direct]). The Contros Optode outputs the following two strings after power-up:

```
$COVER,0,0,D,OPT V 09.04.2014
```

```
$CODT1,0,0,D,2014-07-16,18:50:05,22184,105729,23794,109,,,,*51
```

The \$COVER string identifies the sensor. We're going to ignore this string. The \$CODT string outputs several fixed values, the date and time of the sample, then the three data values we want, followed by more values that we don't care about.

The "format=" field uses the %d and %s operators to skip the unwanted values that precede the three data values that we want. Any values after our three desired values are ignored. Since the sensor uses commas to separate data values, we must also separate our format values with commas.

The 'column=' fields name each of the data columns to be saved by Seaglider.

We use the 'query=' field to instruct Seaglider to read the sensor output and ignore all the characters until the string "CODT1," is encountered. The following characters are then stored and parsed with the "format=" field.

## 10 REMAINING TASKS

Once you have a .cnf file and can read the output of your sensor, there are a few more tasks to do to complete the integration of your sensor.

### 10.1 Put .cnf on Basestation

Make sure to put the .cnf file on the basestation in /usr/local/basestation/Sensors. You also have to update the .sensors file in that directory so that your .cnf file will be evaluated when glider data is being processed.

**Note** *Users performing software upgrade are not required to put .cnf files on the basestation. The Kongsberg basestation software comes preloaded with all .cnf files for Kongsberg supported sensors.*

### 10.2 Update Science File

Update the science file in the Seaglider's home directory on the basestation to tell the Seaglider when you want it to read your sensor. For more information review Chapter 6.

### 10.3 Test Launch

Next, run a sim dive (from the Seaglider menu system, select [launch] and then [test]) and wait for the Seaglider data files to be uploaded to the basestation. Check the data files (.asc, .dat, .eng) to see that they contain the data columns named by the 'prefix=' and 'column=' fields in your .cnf file. Look at the values reported in the .eng file and make sure that they look correct.

For more information about sim dives refer to the Seaglider User Manual or training documents.

### 10.4 Data into .netCDF File

Data in the .eng file is populated into the .nc file by the basestation.

You can use the 'ncdump' utility on Linux to verify that your data is in the .nc file or use ncBrowse for Windows. On linux, if you wanted to verify that the data for 'FL1sig' column was present in the p1270046.nc file, you could use a command like:

```
# ncdump p1270046.nc | grep FL1sig
```

and then review the output for signs that the .nc file contains the specifications for that data column.

## 10.5 Calibration Data

Plotting of your data may require the presence of per-sensor calibration values. You may have to add these calibration values to the `sg_calib_constants.m` file on the basestation.

Users creating their own .cnf files may also need to take additional steps to get the calibration values into the .nc file. These tasks are beyond the scope of this document. KUTI will address these tasks in a future document. For more information contact Kongsberg Seaglider Support ([km.support.seaglider@kongsberg.com](mailto:km.support.seaglider@kongsberg.com)).

## 10.6 Plotting

Once the data (and optionally, calibration data) are stored in the .nc file, the data can be plotted using MATLAB or a tool of your choosing. The specifics of this implementation are outside the scope of this document; for more information Contact Kongsberg Seaglider Support ([km.support.seaglider@kongsberg.com](mailto:km.support.seaglider@kongsberg.com)).

**Note**            *Kongsberg provides MATLAB and Windows 7 tools for plotting Kongsberg supported sensors.*

## APPENDIX A: LOAD CNF FILE

This section describes how to transfer a .cnf file onto the Seaglider, and how to load the .cnf file into the glider's serial sensor library.

### A.1 Check for Existing Filename

Power on the Seaglider and wait until the main menu is displayed.

Before transferring your .cnf file to the Seaglider you need to make sure that the file you are about to transfer does not already exist on Seaglider's compact flash. If the file already exists the file transfer command will fail.

To check if the file exists, from the main menu select 4 [pdos].

Use the 'dir' command to check if the file exists:

```
picoDos>> dir mydev.cnf
```

Note that picoDOS always lists the directory contents using upper case.

If the file is found, you can rename it:

```
picoDos>> ren mydev.cnf mydev.bck
```

or you can simply remove it:

```
picoDos>> del mydev.cnf
```

**Software Upgrade Note**     *If you are upgrading glider software, use the procedure in [Document 4900002 – Updating Seaglider Software for backing up and removing .cnf files](#) instead of the steps in Appendix A.1.*

### A.2 Transfer CNF file to Seaglider

To transfer the file from the computer to the Seaglider use the 'xr' command:

```
picoDos>> xr mydev.cnf
```

After entering this command, use the terminal emulator's 'file send' feature to send the .cnf file to the Seaglider using xmodem. The exact mechanism for this depends on the terminal program you are using. Two examples are provided below.

If you are using TeraTerm go to the File drop-down menu and select Transfer → XMODEM → Send. Then navigate to the location of your .cnf file.

If you are using minicom, enter <CTRL>A<SHIFT>Z to get to the main minicom menu. Enter 'S' to send a file, and choose xmodem from the resulting pop-up menu. When the file browser appears use it to browse your PC's file system to locate the .cnf file you wish to transfer.

**Software Upgrade Note**     *If you are upgrading glider software, use the procedure in Document 4900002 – Updating Seaglider Software for loading glider software and .cnf files instead of the steps in Appendix A.2.*

### A.3 Strip CNF file

Next, strip pad bytes from the end of the .cnf file you just transferred:

```
picoDos>> strip1a mydev.cnf
```

The .cnf file is now ready for use. Restart the Seaglider software:

```
picoDos>>main
```

Wait for the Seaglider main menu to appear before continuing.

**Software Upgrade Note**     *If you are upgrading glider software, use the procedure in Document 4900002 – Updating Seaglider Software instead of Appendix A.2. You are not required to strip the files that were unpacked from glider.tar. Unlike single transfer files, pad bytes do not need to be stripped from a .tar file.*

### A.4 Load CNF into Sensor Library

With the Seaglider '----- Main Menu -----' displayed, you should see:

```
----- Main Menu -----  
1 [param ] Parameters and configuration  
2 [hw   ] Hardware tests and monitoring  
3 [modes ] Test operation modes and files
```

4 [pdos ] PicoDOS commands (and exit)  
5 [launch ] Pre-launch  
Enter selection (1-5,CR): **1**

Make choice 1 [param].

You should now see the '----- Edit parameters -----' menu:

```
----- Edit parameters -----
*Flight control and mission definition
  1 [basic ] Basic mission and glider parameters
  2 [dive  ] Dive parameters
  3 [flight ] Flight parameters
  4 [surface] Surface parameters
  5 [rafos  ] RAFOS parameters
  6 [password] Set/show glider login password
  7 [telnum ] Set/show basestation phone number
  8 [altnum ] Set/show basestation alternate phone number
*Pitch, roll, VBD
  9 [pitch ] Pitch parameters
 10 [roll  ] Roll parameters
 11 [vbd   ] VBD parameters
*Sensors and peripherals
 12 [config ] Hardware configuration parameters
 13 [pressure] Pressure (external) parameters
 14 [intpress] Pressure (internal) parameters
 15 [compass ] Compass parameters
 16 [altim  ] Altimetry parameters
 17 [seabird ] Seabird CT calibration
 18 [power  ] Power parameters
*Utility
 19 [all   ] Edit all parameters
 20 [validate] Validate parameters
 21 [details] Show parameter details
 22 [show  ] Show changed parameters
 23 [clear ] Clear changed parameters
 24 [save  ] Save parameters by name to file
 25 [dump  ] Dump parameters to screen
 26 [load  ] Load parameters from file
 27 [reset ] Reset to defaults
  CR) Return to previous
Enter selection (1-27,CR): 12
```

Make choice 12 [config].



You should now see the ‘----- Edit hardware configuration parameters -----’ menu:

```

----- Edit hardware configuration parameters -----
 1 [show  ] Show current hardware configuration
 2 [motherbo] Configure motherboard
 3 [compass ] Configure compass
 4 [compass2] Configure spare compass
 5 [phone  ] Configure phone
 6 [gps    ] Configure GPS
 7 [xpdr   ] Configure XPDR
 8 [sensor ] Configure sensor
 9 [logger ] Configure logger sensor
10 [seradd ] Add device cnf to hardware library
11 [serdel ] Delete device cnf from hardware library
12 [serlib ] List device cnf files in hardware library
13 [logadd ] Add logger cnf to hardware library
14 [logdel ] Delete logger cnf from hardware library
15 [loglib ] List logger cnf files in hardware library
16 [pressure] Configure external pressure sensor
17 [param  ] Edit parameters directly
CR) Return to previous
Enter selection (1-17,CR): 12

```

Before loading the .cnf file, check to see if any serlib slots are available. Use choice 12 [serlib] to list the contents of the library. The output might look like this:

```

337.156,SSENSOR,N,Current user device files:
337.251,SSENSOR,N, 0: [empty]
337.328,SSENSOR,N, 1: [empty]
337.405,SSENSOR,N, 2: [empty]
337.481,SSENSOR,N, 3: [empty]

```

The above output indicates that all locations in serlib are available.

Now use choice 10 [seradd] on the ‘----- Edit hardware configuration parameters -----’ menu to add your configuration file. Always use the **lowest** user device number that is empty. Do not skip device numbers. Here the lowest device number that is empty is number 0:

```

Enter selection (1-17,CR): 10
337.156,SSENSOR,N,Current user device files:

```

337.251,SSENSOR,N, 0: [empty]  
337.328,SSENSOR,N, 1: [empty]  
337.405,SSENSOR,N, 2: [empty]  
337.481,SSENSOR,N, 3: [empty]

User device number (0-3,CR): **0**

.cnf file for new device: **aa4831.cnf** (Note: use lowercase here)

351.653,SSENSOR,N,reading cnf aa4831.cnf  
352.441,SPOWER,N,Iridium\_during\_init saved current=0.10300  
352.612,SPOWER,N,Iridium\_during\_connect saved current=0.16000  
352.788,SPOWER,N,Iridium\_during\_xfer saved current=0.22300  
352.959,SPOWER,N,GPS saved current=0.02696  
353.110,SPOWER,N,TT8 saved current=0.01496  
353.260,SPOWER,N,TT8\_Active saved current=0.01421  
353.420,SPOWER,N,TT8\_Sampling saved current=0.03743  
353.583,SPOWER,N,TT8\_CF8 saved current=0.04717  
353.740,SPOWER,N,TT8\_Kalman saved current=0.05915  
353.902,SPOWER,N,Analog\_circuits saved current=0.01200  
354.069,SPOWER,N,Compass saved current=0.01573

The command output should show that your new .cnf file was read. If the .cnf file was valid then no errors should be displayed.

**Note**        *You must specify the .cnf file name using all lower case.*

## APPENDIX B: CONFIGURE DEVICE

Once your .cnf file has been loaded into the *serlib* you can add a device of that type to the Seaglider hardware configuration.

### B.1 Check Existing Configuration

Before adding your new serial sensor device you should review the current Seaglider sensor configuration to see which serial sensor slots are available.

From the main menu, choose 1 [param] followed by 12 [config] to get to the ‘----- Edit hardware configuration parameters -----’ menu:

```
----- Edit hardware configuration parameters -----
 1 [show  ] Show current hardware configuration
 2 [motherbo] Configure motherboard
 3 [compass ] Configure compass
 4 [compass2] Configure spare compass
 5 [phone  ] Configure phone
 6 [gps    ] Configure GPS
 7 [xpdr   ] Configure XPDR
 8 [sensor ] Configure sensor
 9 [logger ] Configure logger sensor
10 [seradd ] Add device cnf to hardware library
11 [serdel ] Delete device cnf from hardware library
12 [serlib ] List device cnf files in hardware library
13 [logadd ] Add logger cnf to hardware library
14 [logdel ] Delete logger cnf from hardware library
15 [loglib ] List logger cnf files in hardware library
16 [pressure] Configure external pressure sensor
17 [param  ] Edit parameters directly
CR) Return to previous
Enter selection (1-17,CR): 1
```

Make choice 1 [show] to show the current sensor configuration. Your output might look something like this:

```
321.149,SUSR,N,Sensor in slot 1 is SBE_CT on port 2, TPU04/05, nominally 'CT'
321.255,SUSR,N,Sensor in slot 2 is not installed
321.360,SUSR,N,Sensor in slot 3 is not installed
321.466,SUSR,N,Sensor in slot 4 is not installed
```

```
321.571,SUSR,N,Sensor in slot 5 is not installed
321.677,SUSR,N,Sensor in slot 6 is not installed
321.785,SUSR,N,Logger Sensor in logger slot 1 is not installed
321.912,SUSR,N,Logger Sensor in logger slot 2 is not installed
322.040,SUSR,N,Logger Sensor in logger slot 3 is not installed
322.167,SUSR,N,Logger Sensor in logger slot 4 is not installed
322.288,SUSR,N,Motherboard is Rev.B.1
322.374,SUSR,N,Phone is Iridium9522
322.465,SUSR,N, Phone is not configured for flow control and carrier detect
322.619,SUSR,N, Phone supply is battery 2 with modeled current measurement
322.758,SUSR,N,GPS is Garmin_GPS15H
322.852,SUSR,N,Compass is SP3003
322.935,SUSR,N, Spare compass is not installed
323.037,SUSR,N,XPDR is AAE_955
323.149,SUSR,N,External pressure sensor gain (128.000000) consistent with Paine
```

At the top of the above output the configured serial sensors are displayed. In this case there are no serial sensors configured. (Only the Sea-Bird CT Sail is installed). Thus, sensor slots 2-6 are available.

## B.2 Know What Port to Use

Before proceeding you should know the port and ‘nominally’ name where your new serial device will be connected. For example, ‘port 6, nominally aux 1’. See section 3.1 for additional information.

## B.3 Add New Serial Sensor

Now that you know what sensor slots are free and what port your sensor will be connected to, you can configure the device. From the main menu, use 1 [param] and then 12 [config] to get to the ‘----- Edit hardware configuration parameters -----’ menu:

```
----- Edit hardware configuration parameters -----
 1 [show  ] Show current hardware configuration
 2 [motherbo] Configure motherboard
 3 [compass ] Configure compass
 4 [compass2] Configure spare compass
 5 [phone  ] Configure phone
 6 [gps    ] Configure GPS
 7 [xpdr   ] Configure XPDR
 8 [sensor ] Configure sensor
```

```

9 [logger ] Configure logger sensor
10 [seradd ] Add device cnf to hardware library
11 [serdel ] Delete device cnf from hardware library
12 [serlib ] List device cnf files in hardware library
13 [logadd ] Add logger cnf to hardware library
14 [logdel ] Delete logger cnf from hardware library
15 [loglib ] List logger cnf files in hardware library
16 [pressure] Configure external pressure sensor
17 [param ] Edit parameters directly
CR) Return to previous
Enter selection (1-17,CR): 8

```

To add a serial sensor use choice 8 [sensor]. You will see output similar to what follows. The list of attached sensors is a combination of sensors whose support is built-in to the Seaglider software, and sensors that have been added via .cnf files. Recently configured sensors will be found near the bottom of this list. Here the new serial sensor's device name is 'AA4831' (the device name comes from the 'name' field in the .cnf file). It is being added to serial sensor slot 2 (always use the lowest free slot number). It is being installed on port 5 (nominally 'Optics 2'):

```

Configure device in slot (1-6,CR): 2
377.301,SUSR,N,
Configuring attached sensor:
 1) SBE_CT
 2) SBE_O2
 3) WL_BB2FLVMG
 4) AA3830
 5) WL_BBFL2VMT
 6) AA4330
 7) WL_BB2FLVMT
 8) AA4831
 9) not installed

```

```

Enter selection: 8
 0) port 0, TPU-1/TPU-1, nominally 'null'
 1) port 1, TPU02/TPU03, nominally 'compass'
 2) port 2, TPU04/TPU05, nominally 'CT'
 3) port 3, TPU06/TPU07, nominally 'Optics 1'
 4) port 4, TPU08/TPU09, nominally 'O2'
 5) port 5, TPU10/TPU11 (mux channel 0), nominally 'Optics 2'
 6) port 6, TPU10/TPU11 (mux channel 1), nominally 'aux 1'
 7) port 7, TPU10/TPU11 (mux channel 2), nominally 'aux 2'
 8) port 8, TPU10/TPU11 (mux channel 3), nominally 'altimeter'

```

```
Enter selection: 5
p = 131, port_idx = 3, dev_idx = 8
397.111,HTT8,N,Updating parameter $DEVICE2 to 133
397.268,SSENSOR,N,assigned AA4831 to sensor slot 2 (p = 133 [index 8; port 5])
```

Verify that the configuration was updated by choosing option 1 [show] from the ‘----- Edit hardware configuration parameters -----’ menu to show the current configuration:

```
Enter selection (1-17,CR): 1
321.149,SUSR,N,Sensor in slot 1 is SBE_CT on port 2, TPU04/05, nominally ‘CT’
321.255,SUSR,N,Sensor in slot 2 is AA4831 on port 5, TPU10/TPU11 (mux channel 0),
nominally ‘Optics 2’
321.360,SUSR,N,Sensor in slot 3 is not installed
321.466,SUSR,N,Sensor in slot 4 is not installed
321.571,SUSR,N,Sensor in slot 5 is not installed
321.677,SUSR,N,Sensor in slot 6 is not installed
321.785,SUSR,N,Logger Sensor in logger slot 1 is not installed
321.912,SUSR,N,Logger Sensor in logger slot 2 is not installed
322.040,SUSR,N,Logger Sensor in logger slot 3 is not installed
322.167,SUSR,N,Logger Sensor in logger slot 4 is not installed
322.288,SUSR,N,Motherboard is Rev.B.1
322.374,SUSR,N,Phone is Iridium9522
322.465,SUSR,N, Phone is not configured for flow control and carrier detect
322.619,SUSR,N, Phone supply is battery 2 with modeled current measurement
322.758,SUSR,N,GPS is Garmin_GPS15H
322.852,SUSR,N,Compass is SP3003
322.935,SUSR,N, Spare compass is not installed
323.037,SUSR,N,XPDR is AAE_955
323.149,SUSR,N,External pressure sensor gain (128.000000) consistent with Paine
```

Notice that the sensor slot 2 is now occupied by a serial sensor called AA4831.

If your new device name does not appear in the list on the ‘----- Configuring attached sensor -----’ menu, you will have to do some investigation. First, check if your .cnf file is listed in the serial sensor library (on the ‘----- Edit hardware configuration parameters -----’ menu, choose 12 [serlib]). If the .cnf file is not listed, then perhaps the [seradd] step failed. If the .cnf file is listed in the serlib output, then make sure you specified the .cnf filename on the [seradd] step using lower case (if you use upper case, the .cnf file gets loaded into the library but does not appear on the list of serial devices that you can configure). If necessary, use [serdel] to delete the .cnf file from the serial sensor library, and then repeat the [seradd] step using lowercase for the .cnf file name.

## B.4 Save Changes

You must save the changes to the hardware configuration. Exit the menu by entering <CR> and the respond 'Y' when prompted to 'Re-initialize hardware configuration':

Enter selection (1-17,CR):

Re-initialize hardware configuration [N] **y**

402.153,SSENSOR,N,reading cnf aa4831.cnf

402.872,SSENSOR,N,assigned SBE\_CT to sensor slot 1 (p = 2 [index 0; port 2])

403.476,SSENSOR,N,assigned AA4831 to sensor slot 2 (p = 133 [index 8; port 5])

412.848,SPOWER,N,Iridium\_during\_init saved current=0.10300

413.020,SPOWER,N,Iridium\_during\_connect saved current=0.16000

413.196,SPOWER,N,Iridium\_during\_xfer saved current=0.22300

413.367,SPOWER,N,GPS saved current=0.02696

413.518,SPOWER,N,TT8 saved current=0.01496

413.668,SPOWER,N,TT8\_Active saved current=0.01421

413.828,SPOWER,N,TT8\_Sampling saved current=0.03743

413.991,SPOWER,N,TT8\_CF8 saved current=0.04717

414.148,SPOWER,N,TT8\_Kalman saved current=0.05915

414.310,SPOWER,N,Analog\_circuits saved current=0.01200

414.477,SPOWER,N,Compass saved current=0.01573

414.691,SPOWER,N,SBE\_CT saved current=0.02397

## APPENDIX C: REMOVE DEVICE

If you need to change the port a sensor uses, update a .cnf file, or remove a sensor from the Seaglider, you must first remove the device from the Seaglider configuration.

To do this, start at the main menu and choose 1 [param] followed by 12 [config]. You should now see the ‘----- Edit hardware configuration parameters -----’ menu:

```
----- Edit hardware configuration parameters -----
 1 [show  ] Show current hardware configuration
 2 [motherbo] Configure motherboard
 3 [compass ] Configure compass
 4 [compass2] Configure spare compass
 5 [phone  ] Configure phone
 6 [gps    ] Configure GPS
 7 [xpdr   ] Configure XPDR
 8 [sensor ] Configure sensor
 9 [logger ] Configure logger sensor
10 [seradd ] Add device cnf to hardware library
11 [serdel ] Delete device cnf from hardware library
12 [serlib ] List device cnf files in hardware library
13 [logadd ] Add logger cnf to hardware library
14 [logdel ] Delete logger cnf from hardware library
15 [loglib ] List logger cnf files in hardware library
16 [pressure] Configure external pressure sensor
17 [param  ] Edit parameters directly
CR) Return to previous
Enter selection (1-17,CR): 1
```

Devices are removed by specifying the slot number they current occupy. To see the device installed in each slot, use option 1 [show]. The output might look like the following:

```
321.149,SUSR,N,Sensor in slot 1 is SBE_CT on port 2, TPU04/05, nominally 'CT'
321.255,SUSR,N,Sensor in slot 2 is AA4831 on port 5, TPU10/TPU11 (mux channel 0),
nominally 'Optics 2'
321.360,SUSR,N,Sensor in slot 3 is not installed
321.466,SUSR,N,Sensor in slot 4 is not installed
321.571,SUSR,N,Sensor in slot 5 is not installed
321.677,SUSR,N,Sensor in slot 6 is not installed
321.785,SUSR,N,Logger Sensor in logger slot 1 is not installed
321.912,SUSR,N,Logger Sensor in logger slot 2 is not installed
322.040,SUSR,N,Logger Sensor in logger slot 3 is not installed
```



322.167,SUSR,N,Logger Sensor in logger slot 4 is not installed  
 322.288,SUSR,N,Motherboard is Rev.B.1  
 322.374,SUSR,N,Phone is Iridium9522  
 322.465,SUSR,N, Phone is not configured for flow control and carrier detect  
 322.619,SUSR,N, Phone supply is battery 2 with modeled current measurement  
 322.758,SUSR,N,GPS is Garmin\_GPS15H  
 322.852,SUSR,N,Compass is SP3003  
 322.935,SUSR,N, Spare compass is not installed  
 323.037,SUSR,N,XPDR is AAE\_955  
 323.149,SUSR,N,External pressure sensor gain (128.000000) consistent with Paine

To remove a serial sensor from the configuration, use choice 8 [sensor] from the ‘----- Edit hardware configuration parameters -----’ menu. When prompted, enter the slot number for the serial device you want to remove. Then enter the choice labelled ‘not installed’ to uninstall the device:

Enter selection (1-17,CR): **8**

Configure device in slot (1-6,CR): **2**

2753.343,SUSR,N,Current sensor in slot 2 is AA4831 on port 5, TPU10/TPU11 (mux channel 0), nominally 'Optics 2'

2753.516,SUSR,N,

Configuring attached sensor:

- 1) SBE\_CT
- 2) SBE\_O2
- 3) WL\_BB2FLVMG
- 4) AA3830
- 5) WL\_BBFL2VMT
- 6) AA4330
- 7) WL\_BB2FLVMT
- 8) AA4831
- 9) not installed

Enter selection: **9**

p = -1, port\_idx = -1, dev\_idx = -1

2755.895,HTT8,N,Updating parameter \$DEVICE2 to -1

The serial sensor has now been removed from the system. Be sure to save the changes to the configuration by exiting the current menu (by entering <CR>) and then responding ‘Y’ when prompted to ‘Re-initialize hardware configuration’.

## APPENDIX D: UNLOAD CNF FILE

If you need to update a .cnf file or free up the *serlib* slot it uses, then you must unload the .cnf file from [serlib]. Before doing this you should be sure that any devices using the .cnf file have been removed from the configuration (see preceding section).

To remove the .cnf file from *serlib*, start at the main menu and choose 1 [param] followed by 12 [config]. You should now see the ‘----- Edit hardware configuration parameters -----’ menu. Enter choice 12 [serlib] to list the serial sensor .cnf files stored in *serlib*. Enter choice 11 [serdel] to remove one of the files from *serlib*. When done, exit the ‘----- hardware configuration parameters -----’ menu and respond ‘Y’ when prompted to ‘Re-initialize hardware configuration’.

```
Enter selection (1-17,CR): 12
2770.171,SSENSOR,N,Current user device files:
2770.268,SSENSOR,N, 0: aa4831.cnf
2770.350,SSENSOR,N, 1: [empty]
2770.428,SSENSOR,N, 2: [empty]
2770.506,SSENSOR,N, 3: [empty]

----- Edit hardware configuration parameters -----
 1 [show  ] Show current hardware configuration
 2 [motherbo] Configure motherboard
 3 [compass ] Configure compass
 4 [compass2] Configure spare compass
 5 [phone  ] Configure phone
 6 [gps    ] Configure GPS
 7 [xpdr   ] Configure XPDR
 8 [sensor ] Configure sensor
 9 [logger ] Configure logger sensor
10 [seradd ] Add device cnf to hardware library
11 [serdel ] Delete device cnf from hardware library
12 [serlib ] List device cnf files in hardware library
13 [logadd ] Add logger cnf to hardware library
14 [logdel ] Delete logger cnf from hardware library
15 [loglib ] List logger cnf files in hardware library
16 [pressure] Configure external pressure sensor
17 [param  ] Edit parameters directly
   CR) Return to previous
Enter selection (1-17,CR): 11
2778.328,SSENSOR,N,Current user device files:
```

2778.425,SSENSOR,N, 0: aa4831.cnf  
 2778.507,SSENSOR,N, 1: [empty]  
 2778.585,SSENSOR,N, 2: [empty]  
 2778.664,SSENSOR,N, 3: [empty]

User device number (0-3,CR): **0**

2781.958,SSENSOR,N,assigned SBE\_CT to sensor slot 1 (p = 2 [index 0; port 2])  
 2783.369,HTT8,N,Writing NVRAM...done.  
 2792.111,SPOWER,N,Iridium\_during\_init saved current=0.10300  
 2792.284,SPOWER,N,Iridium\_during\_connect saved current=0.16000  
 2792.461,SPOWER,N,Iridium\_during\_xfer saved current=0.22300  
 2792.633,SPOWER,N,GPS saved current=0.02696  
 2792.786,SPOWER,N,TT8 saved current=0.01496  
 2792.937,SPOWER,N,TT8\_Active saved current=0.01421  
 2793.099,SPOWER,N,TT8\_Sampling saved current=0.03743  
 2793.263,SPOWER,N,TT8\_CF8 saved current=0.04717  
 2793.421,SPOWER,N,TT8\_Kalman saved current=0.05915  
 2793.585,SPOWER,N,Analog\_circuits saved current=0.01200  
 2793.753,SPOWER,N,Compass saved current=0.01573

----- Edit hardware configuration parameters -----

- 1 [show ] Show current hardware configuration
- 2 [motherbo] Configure motherboard
- 3 [compass ] Configure compass
- 4 [compass2] Configure spare compass
- 5 [phone ] Configure phone
- 6 [gps ] Configure GPS
- 7 [xpdr ] Configure XPDR
- 8 [sensor ] Configure sensor
- 9 [logger ] Configure logger sensor
- 10 [seradd ] Add device cnf to hardware library
- 11 [serdel ] Delete device cnf from hardware library
- 12 [serlib ] List device cnf files in hardware library
- 13 [logadd ] Add logger cnf to hardware library
- 14 [logdel ] Delete logger cnf from hardware library
- 15 [loglib ] List logger cnf files in hardware library
- 16 [pressure] Configure external pressure sensor
- 17 [param ] Edit parameters directly
- CR) Return to previous

Enter selection (1-17,CR): **(press Enter)**

Re-initialize hardware configuration [N] **y**

2806.642,SSENSOR,N,assigned SBE\_CT to sensor slot 1 (p = 2 [index 0; port 2])  
 2808.588,SPOWER,N,Iridium\_during\_init saved current=0.10300

2808.761,SPOWER,N,Iridium\_during\_connect saved current=0.16000  
2808.938,SPOWER,N,Iridium\_during\_xfer saved current=0.22300  
2809.111,SPOWER,N,GPS saved current=0.02696  
2809.263,SPOWER,N,TT8 saved current=0.01496  
2809.414,SPOWER,N,TT8\_Active saved current=0.01421  
2809.576,SPOWER,N,TT8\_Sampling saved current=0.03743  
2809.740,SPOWER,N,TT8\_CF8 saved current=0.04717  
2809.898,SPOWER,N,TT8\_Kalman saved current=0.05915  
2810.062,SPOWER,N,Analog\_circuits saved current=0.01200  
2810.231,SPOWER,N,Compass saved current=0.01573

## APPENDIX E: RELOADING THE CNF FILE

Each time you edit the .cnf file and want to test the changes, you need to remove the device (Appendix C), unload the .cnf file (Appendix D), rename or delete the old .cnf file (Appendix A.1), load the updated .cnf file (Appendix A), and reconfigure the device (Appendix B).

Here is a quick summary of the steps involved:

1. Power on Seaglider.
2. At main menu enter choice 1 to go to [param] mode.
3. At '----- Edit parameters -----' menu enter choice 12 to go to [config] mode.
4. At '----- Edit hardware configuration parameters -----' menu enter choice 8 to go to (un)configure a [sensor].
5. Enter the slot number where the sensor is installed (or enter <CR> to return to '----- Edit hardware configuration parameters -----' menu and choose choice 1 to [show] the configuration and see the sensor slots that are in use).
6. After entering the sensor slot number, choose 'not installed' to uninstall it.
7. At '----- Edit hardware configuration parameters -----' menu enter choice 11 [serdel].
8. Choose the number corresponding to the .cnf file you want to remove from *serlib*.
9. At '----- Edit hardware configuration parameters -----' menu hit <CR> to exit the menu.
10. At the 'Re-initialize hardware' prompt enter 'Y'.
11. Return to the main menu.
12. Enter 4 [pdos].
13. In picodos, rename or delete the .cnf file you want to update.
14. Now you can reinstall the updated .cnf file
15. In picoDOS, use '**xr new.cnf**' file to receive the new file. Use the minicom command menu to send over that file from the PC.
16. In picoDOS, '**strip1a new.cnf**'.
17. In picoDOS, enter '**main**' to restart the glider software and menu system.
18. Use menu system [param] → [config] → [seradd] to add the updated .cnf file to the *serlib* library. Watch to make sure no errors are reported when the device is parsed.
19. Use menu system [param] → [config] → [sensor] to configure a device using the new .cnf file.
20. Re-initialize the hardware [Y].
21. Use [hw] → [sensors] to test your serial sensor.