NORTEK MANUALS

DVL Integrator’s Guide

333 | 500 | 1000 kHz

NORTEK

Table of Contents

1 Introduction

The primary objective of this manual is to provide the information needed to control the DVL, a Nortek product based on the AD2CP hardware platform. It is aimed at system integrators and engineers with interfacing experience, but it also includes examples on how to configure and start the instrument for more inexperienced integrators. The document's scope is limited to interfacing and does not address general performance issues of the instrument. For a more thorough understanding about how to operate the instrument, we recommend the Operations Manual.

Nortek online

At our website, www.nortekgroup.com, you will find technical support, user manuals, FAQs and the latest software and firmware. General information, technical notes, and user experience can also be found here.

Your feedback is appreciated

If you find errors, omissions or sections poorly explained, please do not hesitate to contact us. We appreciate your comments and your fellow users will as well.

Contact Information

We recommend first contacting your local sales representative before the Nortek main office. If you need more information, support or other assistance, you are always welcome to contact us or any of our subsidiaries by email or phone

Email: inquiry@nortekgroup.com (general inquiries), support@nortekgroup.com (technical support)
Phone: +47 67 17 45 00

You can also write us at:

Nortek AS
Vangkroken 2
1351 RUD
Norway

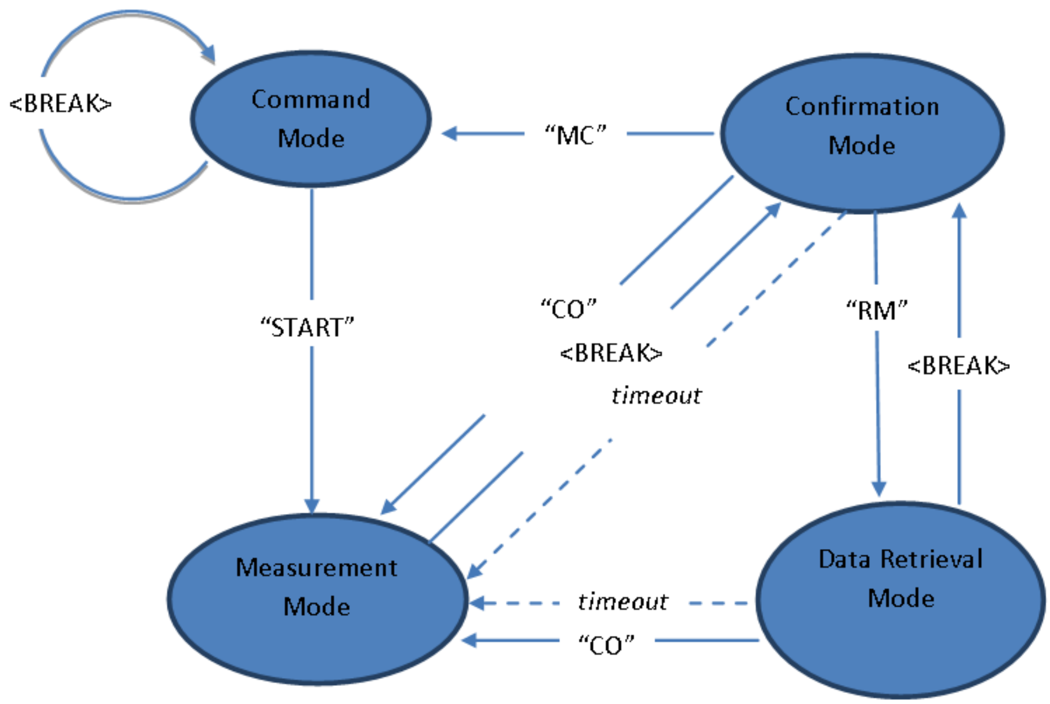
Version/revision	Date	Comments
Version 2016.1	03.2016	Initial version
Version 2016.2	05.2016	
Version 2016.3	22.11.2016	
Version 2017.1	30.05.2017	
Version 2017.2	05.07.2017	
Version 2017.3	20.10.2017	
Version 2018.1	05.10.2018	
Version 2020.1	09.10.2020	Major updates
Version 2021.1	08.03.2021	Minor updates
Version 2021.2	24.11.2021	Minor updates to commands/data formats

Version 2022.1	20.05.2022	Additional commands/data formats
Version 2022.2	05.10.2022	Updated contact information and new data format overview
Version 2023.1	22.03.2023	Created Operation Principles chapter, and added DVL333. Updated Data Formats.

2 Basic interface concept

The Nortek DVL is based on the AD2CP hardware platform. It operates in distinct modes. These modes will have several explicit commands in order to control the instrument. The majority of the commands are initiated from the Command mode. The possible modes for the instrument are:

- Command Command and control
- Data Retrieval Data download from recorder
- Measurement Data collection mode
- Confirmation Confirmation mode



Instrument modes of operation

In measurement mode the DVL is always "awake" by default, and does not enter sleep mode. This is to significantly shorten the time it takes the DVL to respond to a trigger. The DVL can optionally be configured to sleep while waiting for a trigger, in the interest of saving power (see SETDVL,FASTTRIG).

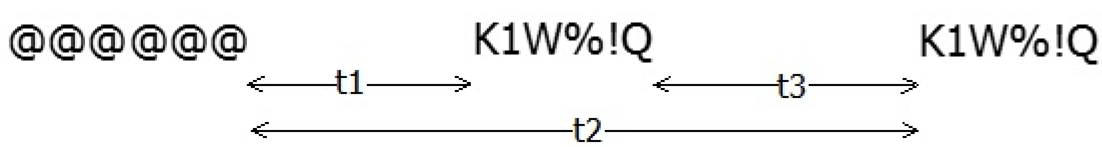
Initializing communication with the instrument is performed by sending a **<BREAK>**, which is defined below. The **<BREAK>** will either set the instrument in Confirmation mode or restart Command mode. The options for changing mode depends on the present mode of the instrument (see diagram above for clarity). The timeout shown in the diagram occurs if no commands are received in the various modes. A timer will then ensure that instrument operation continues. The timeout value in the Confirmation and Data Retrieval Modes is 60 seconds. There is also a timeout in Command Mode when operating over the serial interface. If no commands are received for 5 minutes, a break or a sequence of **@@@@@@** must be sent to wake up the processor.

<BREAK> over the serial RS232/RS422 interface is defined as:

@@@@@@ <delay 150 milliseconds> **K1W%!Q** <delay 400 milliseconds> **K1W%!Q**

The **@@@@@@** are used to wake up the processor when it is in sleep mode. The second sequence of the actual break characters is there to ensure that a break is detected even when the instrument is waking up due to some other cause (e.g. alarm from the real time clock). This ensures that the processor will interpret the following command correctly.

The figure and the table below show the specified timing of the **<BREAK>** sequence:



Symbol	Parameter	Min.	Typical	Max.	Unit
t1	Time from end of @-sequence to start of ***first*** K1W%!Q-sequence.	100	150		ms
t2	Time from end of @-sequence to start of ***second*** K1W%!Q-sequence.	500	1000	2000	ms
t3	Time between ***first*** and ***second*** K1W%!Q-sequence.	300	400		ms

Table 1: Timing of the <BREAK> sequence.

Default values are not listed for all commands in this document as some of them depend on the actual instrument in use. Default parameters can be retrieved be setting default configuration (**SETDEFAULT,CONFIG**) and reading out the desired parameter through the appropriate **GET** command.

The same is the case for some of the minimum and maximum values that depend on the actual instrument in use. The parameter range for the various arguments can be retrieved through the appropriate **GET[...]LIM** command, e.g. **GETDVLLIM,SR** to read the valid range of cell sizes.

All command parameters should be set explicitly, e.g.

```
SETDVL,SR=1.0,SA=35.0
OK
```

A configuration of the instrument should always start with setting the default configuration, e.g.

```
SETDEFAULT,CONFIG
OK
```

3 Interfaces

In addition to the traditional serial port interface for real time data output there are several options for communication over Ethernet. The Ethernet communication is handled by a dedicated processor in the instrument. This network processor runs a Linux operating system, which makes it possible to connect to the instrument via Telnet, raw connections and FTP. The network processor mainly provides Ethernet connectivity. The other processor in the instrument, called the Doppler processor, is where the commands end up and where they are used to perform the measurements as specified. Below are some details.

3.1 Command interface

The command interface makes it possible to communicate with a DVL instrument using terminal software, the serial port and a set of commands. The interface is also available over Telnet. Some highlights:

- ASCII based and line oriented. Commands are terminated with CR/LF
- Optional capsulation of commands using NMEA style prefix and checksum to ensure data integrity
- NMEA style commands will return argument names in their response
- Argument limits can be retrieved through commands
- Comprehensive validation and error handling is implemented.
- Invalid configurations return the erroneous argument with limits directly, so that each subsequent error can be handled until a valid configuration is achieved
- A single command can be used to retrieve the complete configuration of the instrument with optional output to file
- Commands to set default parameters
- External controllers can use commands to store data in the raw data file (e.g. GPS position)

3.2 Ethernet Operation

The AD2CP uses TCP (transmission control protocol) for both command processing and data transmission. The Internet Protocol uses a combination of the IP address and port to uniquely identify a communications channel between two computers. For the AD2CP, different ports represent different means of communicating with the instrument. TCP ports 9001, 2002, 9004 are assigned for the following uses:

- Port 9000 is a telnet-protocol ASCII interface (require username / password authentication)
- Port 9001 is a raw (binary) interface (requires username / password authentication)
- Port 9002 is a binary data only channel (no input accepted)
- Port 9004 is an ASCII data only channel (no input accepted).
- Port 9010/9011 - Additional Output Data Format. See chapter SETAODF.

The password entry is ignored if password authentication, as shown in the web page configuration, is disabled (so any input, including an empty password, is accepted). The command and data record formats for the interfaces are the same as for the serial port.

3.3 Telnet Connection

The telnet interface (TCP/IP port 9000) is used for user interaction with the instrument. This dedicated port can be used for entering commands and getting human readable responses (ASCII). The supported command set is available in the Commands section. The Windows telnet client can be used to connect into the instrument using the command line telnet ip_address 9000. You will get prompted for a username (nortek) and password (hit Enter if password protection hasn't been enabled via the Web interface).

Signature Username: nortek
Password:
Nortek Signature Command Interface

The interface is very similar to the direct serial interface over RS232/RS422 but some additions are made to simplify the interfacing. Most notable is the ability to send a <break> to the Doppler processor just by using Ctrl-C (ASCII 0x03). The internal application takes care of waking up the Doppler DSP and timing the delivery of the break string. The telnet server is not configured to echo characters, so users wishing to see and/or edit commands before sending them to the instrument should enable local echo and local line editing. If those features are desired, a telnet client capable of supporting local echo and local line editing must be used (e.g. PuTTY).

Port 9000 is dedicated for ASCII only communication whereas the ports described in the next section provide the complete set of data, including binary output of the measurements. A telnet client should not be used to access these ports. Read more about this in the next section.

To terminate the telnet connection, enter Ctrl-X (ASCII 0x18).

3.4 Raw Connections

A port can be understood as a address point between two communicating parts. When first connecting to a data listening port, the string "\r\nNortek name Data Interface\r\n" (name is replaced by the instrument host name) is sent to identify the instrument that has responded to the connection request. TCP ports 9001, 9002 and 9004 are assigned for the following uses:

- Port 9001 is used for machine driven control. This port requires username/password. The serial port data is translated directly into TCP/ IP over Ethernet. Binary data generated in measurement mode is visible on this port. Standard streaming record delineation techniques must be used in order to make sure that the received data is properly synchronized for decoding. A break can be sent by sending the string K1W%!Q<CR><LF> to the instrument or a Ctrl-C character (ASCII 0x03) (Ctrl-C has to be sent on its own and not embedded in any command). The internal application takes care of the appropriate timing of the break sent over the internal serial port. This port require username / password authentication. Refer to previous section for example. The password entry is ignored if password authentication, as shown in the web page configuration, is disabled (so any input, including an empty password, is accepted). The command and data record formats for the interfaces are the same as for the serial port.
- Port 9002 is a data only channel which will output all data that is configured for serial output. This can, for example, be used by display only software while configuration is done by another application.
- Port 9004 outputs ASCII data (no binary) that is configured for serial output.

A telnet client should not be used to access these ports. Telnet incorporates its own binary protocol which is neither interpreted nor sent via the raw connection. Using a telnet client on these ports will result in extraneous characters being sent and certain binary characters being interpreted by the client.

3.5 FTP

The internal data recorder is accessed over Ethernet using a standard FTP (File Transfer Protocol) client. FTP serves as a simple way to download measured data from the instrument.

When an FTP connection is active, the internal state of the machine is changed so that commands are no longer processed (and an error is returned when commands are entered). Terminating the FTP connection or sending a BREAK followed by the CO command will switch the instrument back to the mode it was in before the FTP session began. If a break command is sent while an FTP transaction is in progress, the FTP connection will be forcibly terminated.

If an FTP connection is done when the instrument is in measurement mode (see Figure 1), the FTP connection is made through data retrieval mode. When the FTP connection is terminated, the instrument will then return to measurement mode. If there is no data transferred or FTP commands sent for 120 seconds, the FTP connection will terminate and the instrument will return to measurement mode.

3.6 HTTP

HTTP (Hypertext transfer protocol) can also be used for data transmission. For organizations with strong security / firewall restrictions, FTP access to the instrument may not be permitted. For that reason, a web page allowing individual data files to be downloaded has been implemented in the Ethernet processor. The web page can be accessed by clicking on the “Data Download (HTTP)” link from the main web page.

3.7 UDP

UDP (user datagram protocol) can also be used for data transmission. When using UDP, the data collection software simply waits for data to be sent from the instrument without having to establish a connection first. This may be useful for cases in which instrument power is intermittently interrupted and re-connecting to the instrument is not desirable. One downside to UDP communications is that transmission of the data is not guaranteed. On a noisy / error-prone connection, it is possible that the occasional datagram may be dropped. If every data record must be received, then TCP is recommended.

Multicast operation is also supported over UDP if data should be distributed to multiple clients. This is enabled by entering an UDP address in the multicast address range.

In order to use UDP in a power-safe configuration, the IP address of the data collection software and port must first be configured using the web interface. The IP address identifies the client to which the data is to be sent and the port may be used to uniquely identify the instrument to the application. The same port may be used for all instruments if the data collection software examines the IP address of the received datagram to identify the instrument. Once this information has been configured, the Ethernet processor will automatically send real-time data records to the configured address / port. An instrument in measurement mode re-enters measurement mode shortly after a power-cycle, so the data collection software will immediately receive new data without having to re-establish a connection.

3.8 Time Synchronization

Time Synchronization allows the internal clocks used for data collection to be synchronized to an external source. Either NTP (Network Time Protocol) or PTP (Precision Tim Protocol) can be selected.

Precision Time Protocol (IEEE-1588) is a standard used for distributing a high-resolution absolute time throughout an Ethernet network. The DVL can be configured to act as a slave to an existing PTP master clock (customer supplied) located in the same Ethernet LAN. The instrument contains a high-resolution clock which is synchronized and conditioned using PTP when enabled. The timestamps contained within the data records are then generated from this clock. When synchronized, these timestamps are typically aligned to within +/- 1 microseconds.

The PTP master clock must use UDP (layer three) and be configured for two-step operation with an end-to-end delay mode in order to be compatible with the DVL's PTP implementation. Using PTP does not affect the choice of UDP or TCP for the transport of data.

Network Time Protocol is a purely software based Internet time synchronization protocol. In comparison to PTP, NTP will typically take 1-2 minutes to fully achieve sync and will generally synchronize to within 1 ms for a local server. While there are no special requirements for the NTP time server in terms of hardware, synchronization is strongly affected by the network path between the server and the client and, for that reason, it is strongly recommended that the NTP server be located on the same local Ethernet network as the instrument.

Setting up time synchronization can be done through the web site, and the commands for enabling time synchronization is described in the Commands section.

Data collection cannot occur if the internal time has not been synchronized to the master clock.

3.9 Triggers

The DVL offers four main types of triggers: Internal Sampling, TTL trigger, RS-485 trigger and trig on command.

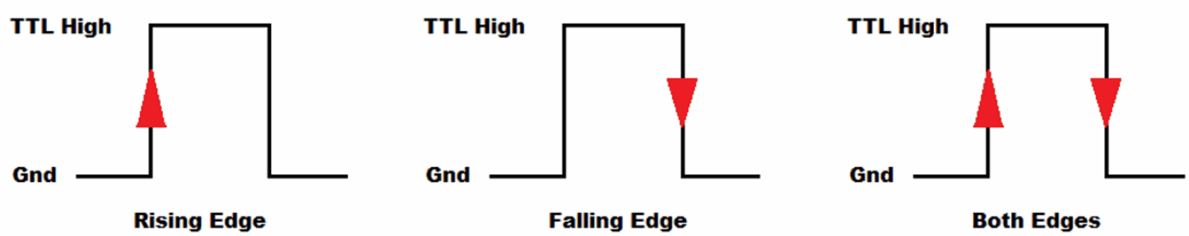
Internal Sampling (INTSR)

Sampling rates from 1 to 8 Hz are available. For long ranges the maximum sampling rate is reduced. The command GETMISCLIM,BTENDRANGE can be used to find the maximum range for the given sampling rate.

TTL Trigger

The DVL can trig on either Rising Edge, Falling Edge or Both Edges of a TTL Signal. When triggered the instrument will perform a complete ping (Tx and Rx) before it goes back to monitoring the trigger. Any triggers asserted during an ongoing ping will be ignored. By default, the Fast Trigger option is enabled and the DVL does not sleep between pings, remaining fully powered. The alternative option is a power-saving trigger, where there is a partial power-down between pings (FASTTRIG=0).

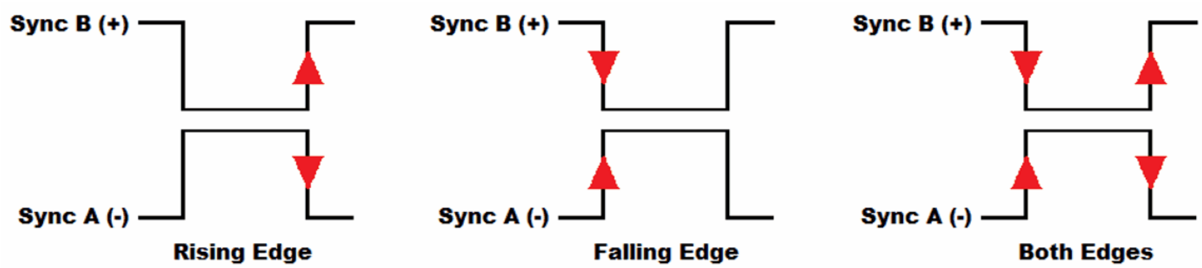
The requirements for the TTL input is V_{low} < 0.7 V and V_{high} > 2.5 V. The TTL input tolerate voltages between 0-5.5 V. The pulse length should be minimum 1 ms, the latency (trigger to start of transmit pulse with FASTTRIG disabled) is 8.3 ms, and the max length of transmit pulse is 13.3 ms for the DVL1000, and 26.6 ms for the DVL333/DVL500. The actual length of the transmit pulse varies with the distance to the bottom.



DVL trigger types

RS-485 Trigger

A RS-485 signal can be used to trig the DVL. The DVL can trig on either Rising Edge, Falling Edge or Both Edges of a RS-485 Signal. The following figure shows the polarities of the differential RS-485 signal pair for the trigger types. When triggered the instrument will perform a complete ping (Tx and Rx) before it goes back to monitoring the trigger. Any triggers asserted during an ongoing ping will be ignored. The pulse length should be minimum 1 ms, the latency (trigger to start of transmit pulse with FASTTRIG disabled) is 8.3 ms, and the max length of transmit pulse is 13.3 ms for the DVL1000, and 26.6 ms for the DVL333/DVL500. The actual length of the transmit pulse varies with the distance to the bottom.



DVL RS-485 Trigger types

Trig on Command
When the TRIG parameter of the SETDVL command is set to “SERIAL” the DVL is trigged by sending a “TRIG[CrLf]” command. The trigger time will be when the end of the [CrLf] is received.

4 Using the Command Interface

This section covers the commands that can be used to control an AD2CP instrument. Not every command will be appropriate for every instrument.

Some pointers:

- A configuration of the instrument should always start with setting the default configuration, e.g.:
  ```
  SETDEFAULT,CONFIG
  OK
  ```

- All command parameters should be set explicitly, e.g.:
  ```
  SETDVL,TRIG="TTLRISE"
  OK
  ```

- Sometimes you may get an ERROR response after tying to save the configuration or start/deploy the instrument. This doesn't necessarily mean that something is wrong with the instrument, but is most often a sign that the configuration isn't going to work. Any ERROR response can be interrogated with **GETERROR**, e.g.:
  ```
  SAVE,CONFIG
  ERROR
  GETERROR
  GETERROR,NUM=56,STR="Invalid setting: Avg Average Interval too low for the configured number of pings and profiling distance",LIM="GETAVG1LIM,AI=([360;1800])"
  OK
  ```
  Here, the instrument is reporting that we have set the average interval to be too short, and it provides the limits for the AI that are allowed if we are going to keep the same number of pings and profiling distance. You could increase the average interval, decrease the number of pings, or decrease the profiling distance (i.e. number/size of cells) to fix the error.

Command Limit Formats

The limits for the various arguments are returned as a list of valid values, and/or ranges, enclosed in parenthesis **()**. An empty list, **()**, is used for arguments that are unused/not yet implemented. Square brackets **[]** signify a range of valid values that includes the listed values. String arguments are encapsulated with **“”**, like for normal parameter handling. A semicolon, **;**, is used as separator between limits and values.

The argument format can also be inferred from the limits, integer values are shown without a decimal point, floating point values are shown with a decimal point and strings are either shown with the string specifier, “”, or as a range of characters using ‘’ for specifying a character.

Examples:

`[1;128]` – Integer value, valid from 1 to 128.

`([1300.00;1700.00];0.0)` – Floating point value, valid values are 0.0 and the range from 1300.00 to 1700.00.

`(['0';'9'];['a';'z'];['A';'Z'];'.')` – String argument with valid characters being . and the character ranges a-z, A-Z, 0-9.

`("BEAM")` – String argument with BEAM being the only valid string.

`(0;1)` – Integer value with two valid values, 0 and 1.

NMEA interface example:

```
$PNOR,GETCURPROFLIM*7E
$PNOR,GETCURPROFLIM,NC=([1;200]),CS=([0.50;4.00]),BD=([0.50;68.00]),C
Y=("BEAM";"XYZ"),PL=([-20.0;0.0];-
100.0),VP=(),VR=([1.00;5.00]),DF=(3;100;101;102;103;104;150),NB=([0;4
]),CH=([0;4321])*2B
$PNOR,OK*2B
```

Regular interface example:

```
GETDVLLIM
(0;[2;20]),
("INTSR";"TTLEDGE";"TTLRISE";"TTLFALL";"RS485EDGE";"RS485RISE";"RS485
FALL";"SERIAL"),(1.0;2.0;3.0;4.0;5.0;6.0;7.0;8.0),(['0';'9'];
['a';'z'];['A';'Z'];'_';'.'),([1300.00;1700.00];0.0),([0.0;50.0])
OK
```

4.1 User Cases

4.1.1 Internal 4 Hz trigger

Internal trigger at 4 Hz using measured sound velocity calculated using a salinity of 35.0 ppt. Velocity range 5 m/s along beam and a range of 10 meters:
(Since the instrument is in measurement mode, the first step is to get it into Command Mode)

```
Send Break
CONFIRM
MC
DVL - NORTEK AS.
Version 4041_10 (Nov 15 2017 14:38:16)
COMMAND MODE
OK

SETDEFAULT,CONFIG
OK
SETDVL,CP=0,TRIG="INTSR",SR=4.0,FN="",SV=0.0,SA=35.0
OK
SETBT,RANGE=10.00,VR=5.00,NB=4,CH=0,DF=21,PL=-
2.0,WT="OFF",WTDF=22,BD=0.01,PLMODE=”MAX”
OK
SAVE,CONFIG
OK
START
OK
```

4.1.2 External trigger, rising edge

External, rising edge, TTL trigger using fixed sound velocity at 1500.0 m/s. Velocity range 2.5 m/s along beam and a range of 30 meters. This example also shows retrieval of argument limits and checking error conditions as the range is here first set erroneously to 100 meters: (Since the instrument is in measurement mode, the first step is to get it into Command Mode)

```
Send Break
CONFIRM
MC
DVL - NORTEK AS.
Version 4041_10 (Nov 15 2017 14:38:16)
COMMAND MODE
OK

SETDEFAULT,CONFIG
OK
GETDVLLIM,TRIG
("INTSR";"TTLEDGE";"TTLRISE";"TTLFALL";"RS485EDGE";"RS485RISE";"RS485FALL";
"SERIAL")
OK
SETDVL,CP=0,TRIG="TTLRISE",SV=1500.0
OK
```

```
SETBT,RANGE=100.00,VR=2.50,NB=4,CH=0,DF=21,PL=-
2.0,WT="OFF",WTDF=22,BD=0.02,PLMODE=”MAX”
OK
SAVE,CONFIG
ERROR
GETERROR
261,"Invalid setting: Bottom track range
invalid","SETBT,RANGE=([5.00;30.00])"
OK
SETBT,RANGE=30.0
OK
SAVE,CONFIG
OK
START
OK
```

5 Operation Principles

This section covers the operational principles when configuring a DVL.

5.1 Modes

Tracking Mode: The bottom detection mode allows for four different options for detecting the bottom. The modes are as follows:

Normal mode is intended for general DVL use and is used for the DVL's full range of distances from the bottom as well as the full range of velocities. The bottom track is established and output following a valid acquisition of 8 of last 10 pings for a minimum of three beams.

Crawler mode is a dedicated mode that uses a different acoustic transmission (two short pings in sequence). The crawler mode is for aiding vehicles that intend to operate in station keeping mode or move very slowly. The advantage of the Crawler mode is that it has lower uncertainty in the velocity estimates and it will allow for closer operation to the boundary. There are limitations in range and along beam velocity and these limitations are reflected in the settings of these two parameters when the mode is changed to crawler.
The minimum detectable distance is 10 cm and the maximum is 10 m , 20 m, 30 m for 1000 kHz, 500 kHz and 333 kHz, respectively. The default velocity limit along beam is 0.15 m/s, which translates to a maximum in the horizontal, XY plane as 0.35 m/s; this may be adjusted with the bottom track velocity range. The horizontal velocity range is approximately 2.5 times greater than the beam velocity range. Bottom track estimates are not reported when the limits are exceeded. The maximum along beam velocity may be manually set to 0.4 m/s.

Auto mode is a hybrid of the Normal and Crawler modes. As the name suggests, the DVL will automatically change between Normal and Crawler modes based on distance to the bottom as well as the measured velocity. In Auto mode, the DVL will switch to Crawler mode after a consistent detection of 10 consecutive pings with an along-beam velocity of 10 cm/s and a bottom distance below 7.5 m, 12.5 m and 17.5 m for the frequencies of 1000 kHz, 500 kHz, and 333 kHz, respectively. To cope with accelerations, the DVL will switch back to Normal mode on the first detection of an along-beam velocity above the 10 cm/s threshold. It will also switch back to normal mode after four consecutive missed detections or four detections above the before mentioned range limit.

Fast_Acq mode: This mode is similar to Normal mode but it does not have an acquisition requirement to begin reporting bottom detection. It is also the bottom track legacy mode of the Nortek DVL. Each pings bottom detection uses less of the detection history for the current detection. Note that this mode exposes the DVL to a greater level of false detects.

6 Output Data Formats

This section gives an overview over the different output data formats. This chapter is divided into four part; Bottom Track data formats, Water Track data formats, Current Profile data formats, and Altimeter data formats. Each of these chapter are divided into two sections, Binary output formats and ASCII output formats.

6.1 Bottom Track Data Formats

The data format of the Bottom Track mode is controlled by the SET/GETBT command. The DF parameter of this command sets the data format

Binary:

Data format (DF)	Description
21	Nortek DVL Bottom Track data format
150	RDI PD0 – RDI documentation
154	RDI PD4 – RDI documentation
155	RDI PD5 - RDI documentation

Table 2: Available Binary Data formats for Bottom Track measurements.

ASCII:

Data format (DF)	Description
156	RDI PD6 – RDI documentation
350	NMEA $PNORBT1 including tags
351	NMEA $PNORBT0 (same as DF350 but no tags)
354	NMEA $PNORBT3 including tags
355	NMEA $PNORBT4 (same as DF354 but no tags)
356	NMEA $PNORBT6 including tags
357	NMEA $PNORBT7 (same as DF356 but no tags)
358	NMEA $PNORBT8 (sensors) including tags
359	NMEA $PNORBT9 (Sensors) (same as DF358 but no tags)

Table 2: Available ASCII Data formats for Bottom Track measurements.

6.2 Water Track Data Formats

The data format of the Water Track mode is controlled by the SET/GETBT command. The WTDF parameter of this command sets the data format

Binary:

Data format (DF)	Description
22	Nortek DVL Water Track data format
150	RDI PD0 – RDI documentation
154	RDI PD4 – RDI documentation
155	RDI PD5 - RDI documentation

Table 3: Available Binary Data formats for Water Track measurements.

ASCII:

Data format (DF)	Description
156	RDI PD 6 – RDI documentation
404	NMEA $PNORWT3 including tags
405	NMEA $PNORWT4 (same as DF404 but no tags)
406	NMEA $PNORWT6 including tags
407	NMEA $PNORWT7 (same as DF406 but no tags)
408	NMEA $PNORWT8 (sensors) including tags
409	NMEA $PNORWT9 (sensors) (same as DF408 but no tags)

Table 3: Available ASCII Data formats for Water Track measurements.

6.3 Current Profile Data Formats

The data format of the Current Profiling mode is controlled by the SET/GETCURPROF command. The DF parameter of this command sets the data format.

Binary:

Data format (DF)	Description
3	Nortek Current Profile data format.
150	RDI PD0 – RDI Documentation.

Table 4: Available Binary Data formats for Current Profile measurements.

ASCII:

Data format (DF)	Description
100	NMEA Nortek Prolog format (see Prolog documentation)
101	NMEA $PNORI1, $PNORS1, $PNORC1, No tags
102	NMEA $PNORI2, $PNORS2, $PNORC2, Including tags
103	NMEA $PNORH3, $PNORS3, $PNORC3, Including tags
104	NMEA $PNORH4, $PNORS4, $PNORC4, No tags

Table 4: Available ASCII Data formats for Current Profile measurements.

6.4 Altimeter Data Formats

The data format of the Altimeter is controlled by the SET/GETALTI command. The DF parameter of this command sets the data format.

Binary:

Data format (DF)	Description
30	Nortek Altimeter data format

Table 5: Available Binary Data formats for Altimeter measurements.

ASCII:

Data format (DF)	Description
200	NMEA $PNORA
201	NMEA $PNORA including tags
202	NMEA $SDDBT

Table 5: Available ASCII Data formats for Altimeter measurements.

7 Commands

This chapter contains an overview over all the commands, including a detailed description. Please refer to the previous chapter for examples, and how to use the commands.

7.1 List of Commands

Below is a list of all available commands with a short description and information about which mode they can be used in. For more information about each command see the following chapters. The arguments that can be used with each command are described in the respective chapter. Note that some of the commands requires at least one argument to be used.

Command	Description	Mode
SETINST	Set instrument main settings	COMMAND
GETINST	Get instrument main settings	COMMAND
GETINSTLIM	Set instrument main setting limits	COMMAND
SETCLOCK	Set instrument clock	COMMAND RETRIEVAL
GETCLOCK	Get instrument clock	COMMAND RETRIEVAL
SETCLOCKSTR	Set instrument clock as string	COMMAND RETRIEVAL
GETCLOCKSTR	Get instrument clock as string	COMMAND RETRIEVAL
GETCLOCKSTRMS	Get instrument clock with milliseconds resolution as string	COMMAND RETRIEVAL
GETCLOCKMS	Get instrument clock with milliseconds precision	COMMAND RETRIEVAL
START	Go to measurement mode.	COMMAND
SETBT	Set bottom track settings	COMMAND
GETBT	Get bottom track settings	COMMAND
GETBTLIM	Get bottom track setting limits	COMMAND
SETDVL	Set dvl main settings	COMMAND
GETDVL	Get dvl main settings	COMMAND
GETDVLLIM	Get dvl main setting limits	COMMAND
SETFASTPRESSURE	Set fast pressure settings	COMMAND
GETFASTPRESSURE	Get fast pressure settings	COMMAND
GETFASTPRESSURELIM	Get fast pressure setting limits	COMMAND

SETTRIG	Set trigger settings	COMMAND
GETTRIG	Get trigger settings	COMMAND
GETTRIGLIM	Get trigger setting limits	COMMAND
SETTRIG1	Set trigger settings for secondary plan	COMMAND
GETTRIG1	Get trigger settings for secondary plan	COMMAND
GETTRIGLIM1	Get trigger setting limits for secondary plan	COMMAND
SETCURPROF	Set current profile configuration	COMMAND
GETCURPROF	Get current profile configuration	COMMAND
GETCURPROFLIM	Get current profile configuration limits	COMMAND
SETALTI	Set altimeter configuration	COMMAND
GETALTI	Get altimeter configuration	COMMAND
GETALTILIM	Get altimeter configuration limits	COMMAND
SETUSER	Set instrument user settings	COMMAND
GETUSER	Get instrument user settings	COMMAND
GETUSERLIM	Get instrument calibration parameter limits	COMMAND
SETAODF	Set additional online data formats	COMMAND
GETAODF	Get additional online data formats	COMMAND
GETAODFLIM	Get available additional online data formats	COMMAND
ID	Get instrument Id	COMMAND
SETDEFAULT	Reload default settings	COMMAND
SAVE	Save settings for next measurement	COMMAND
MC	Go into command mode	CONFIRMATION
RM	Go into data retrieval mode	CONFIRMATION
CO	Go into measurement mode	CONFIRMATION RETRIEVAL
POWERDOWN	Set instrument in sleep mode	COMMAND
ERASE	Erase all files on the recorder	COMMAND
FORMAT	Format the recorder	COMMAND
READCFG	Read current configuration	COMMAND
INQ	Inquires the instrument state	COMMAND CONFIRMATION RETRIEVAL MEASUREMENT
GETERROR	Returns a full description of the last error condition to occur	COMMAND CONFIRMATION MEASUREMENT

		RETRIEVAL
GETERRORNUM	Returns a integer error value of the last error condition to occur	COMMAND CONFIRMATION MEASUREMENT RETRIEVAL
GETERRORSTR	Returns a string error description of the last error condition to occur	COMMAND CONFIRMATION MEASUREMENT RETRIEVAL
GETALL	Retrieves all relevant configuration information for the instrument	COMMAND
RECSTAT	Return recorder state	COMMAND RETRIEVAL
TAG	Write a Tag to output file and data output	COMMAND CONFIRMATION MEASUREMENT RETRIEVAL
PTPSET	Set precision time protocol parameters	COMMAND
PTPGET	Get precision time protocol parameters	COMMAND
GETHW	Returns hardware specifications	COMMAND CONFIRMATION MEASUREMENT RETRIEVAL
ADDLICENSE	Add license key	COMMAND
DELETELICENSE	Delete license key	COMMAND
LISTLICENSE	Lists all license keys in instrument	COMMAND

7.2 Instrument main settings

Commands: SETINST, GETINST, GETINSTLIM
Command type: CONFIGURATION
Mode: COMMAND

Instrument main settings

Argument	Description
BR	Baud Rate 300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200, 230400, 460800, 625000, 1250000 Unit: [bit/s]
RS	Serial protocol 232, 422

LED	Enable/disable LED blink in head. When set to "ON24H" the LED will illuminate the first 24 hours of the measurement. "OFF": Turn LED Off "ON": Turn LED On "ON24H": Turn LED On for 24 hours
ORIENT	Sets the instrument orientation. Not used for DVL. "AUTOXUPDOWN": Auto X up/down "AUTOYUPDOWN": Auto Y up/down "AUTOZUPDOWN": Auto Z up/down "ZUP": Z up "ZDOWN": Z down "XUP": X up "XDOWN": X down "YUP": Y up "YDOWN": Y down "AUTO3D": Auto 3D "AHRS3D": AHRS 3D
CMTOUT	Command mode timeout Unit: [s]
DMTOUT	Data retrieval mode timeout Unit: [s]
CFMTOUT	Confirmation mode timeout Unit: [s]

SETINST

Set instrument main settings

Example:

```
SETINST,BR=57600
```

GETINST

Get instrument main settings

Example:

```
GETINST,BR=460800,RS=232,LED="ON",ORIENT="AUTOZUPDOWN",CMTOUT=300,DMT
OUT=60,CFMTOUT=60
```

GETINSTLIM

Set instrument main setting limits

7.3 Clock settings

Commands: SETCLOCK, GETCLOCK
Command type: CONFIGURATION
Mode: COMMAND, RETRIEVAL

Instrument Real Time Clock specified in date parts

Argument	Description
YEAR	The year, e.g. 2020 Values: [0; 9999]
MONTH	The number of month 1-12 (Jan = 1) Values: [1; 12]
DAY	The number day of month 1-31 Values: [1; 31]
HOUR	The hour of day 0-23 Values: [0; 23]
MINUTE	The minute of hour 0-59 Values: [0; 59]
SECOND	The second of minute 0-59 Values: [0; 59]

Note: Note that all parameters must be set when using the set command

SETCLOCK

Set instrument clock

Example:
```
SETCLOCK,YEAR=2020,MONTH=11,DAY=28,HOUR=13,MINUTE:15,SECOND=45
```

GETCLOCK

Get instrument clock

Example:
```
GETCLOCK
2022,12,13,15,24,33
OK
```

7.4 Clock settings as strings

Commands: SETCLOCKSTR, GETCLOCKSTR
Command type: CONFIGURATION
Mode: COMMAND, RETRIEVAL

Set or retrieve the Real Time Clock using a string. Must use the format as shown: yyyy-MM-dd HH:mm:ss

Argument	Description
TIME	Text string with this format yyyy-MM-dd HH:mm:ss (use UTC)

SETCLOCKSTR

Set instrument clock as string

Example:
```
SETCLOCKSTR,TIME="2020-11-12 14:27:42"
```

GETCLOCKSTR

Get instrument clock as string

Example:
```
GETCLOCKSTR
GETCLOCKSTR, TIME = "2014-11-12 14:27:42"
```

7.5 Get clock as string (with ms)

Command: GETCLOCKSTRMS
Command type: CONFIGURATION
Mode: COMMAND, RETRIEVAL

Retrieve the Real Time Clock using a string. Must use the format as shown: yyyy-MM-dd HH:mm:ss.sss

Argument	Description
TIME	Text string with this format yyyy-MM-dd HH:mm:ss.sss (use UTC)

Example:
```
GETCLOCKSTRMS
"2021-05-14 19:32:11.000"
```

OK

7.6 Get clock (with ms)

Command: GETCLOCKMS
Command type: CONFIGURATION
Mode: COMMAND, RETRIEVAL

Retrieve the Real Time Clock with milliseconds resolution. The API uses the Date() object. If this command is sent directly after an instrument reboot, there may be a delay up to 1 second before the response. Otherwise, the normal latency is below 10 ms. When comparing clocks over the serial interface, the receiving side should capture its local clock when start bit of the first character in the returned message is detected. This will ensure minimum latency between comparing the local clock with the instrument clock. Over an Ethernet interface, there will be some milleseconds latency using this method. To achieve the best precision in a system using Ethernet, PTP should be used, as described in the PTP section.

Argument	Description
YEAR	The year, e.g. 2020 Values: [0; 9999]
MONTH	The number of month 1-12 (Jan = 1) Values: [1; 12]
DAY	The number day of month 1-31 Values: [1; 31]
HOUR	The hour of day 0-23 Values: [0; 23]
MINUTE	The minute of hour 0-59 Values: [0; 59]
SECOND	The second of minute 0-59 Values: [0; 59]
MSEC	Milliseconds 0-999 Values: [0; 999]

Note: Note that all parameters must be set when using the set command

Example:
GETCLOCKMS

7.7 Start measurements

Command: START
Command type: ACTION
Mode: COMMAND

Go in measurement mode.

Note that the START command will save the configuration as well as starting the measurement, as if a SAVE,CONFIG command were sent.

Example:

```
START
OK
```

7.8 Bottom track settings

Commands: SETBT, GETBT, GETBTLIM
Command type: CONFIGURATION
Mode: COMMAND **License:** BottomTrack

The BT command configures the Bottom Track measurements and the Water Track measurements. The valid range for the various arguments should be verified using the GETBTFLIM command, also for the values listed here as they may change with firmware versions and instrument frequencies.

Argument	Description
RANGE	Maximum depth Unit: [m]
VR	Max velocity expected in the water column during deployment. Velocity range spans from -VR to +VR. Unit: [m/s]
NB	Select number of beams; 0 disables all beams. Maximum number of beams equals number of transducers on sensor head.
CH	Beam selection. E.g. CH=123 will enable beams 1, 2 and 3.
DF	Data format 21: Bottom Track Data Record version 1. 150: RDI PD0 - RDI documentation 154: RDI PD4 - RDI documentation 156: RDI PD6 – RDI Documentation. 350: NMEA $PNORBT1 including tags. 351: NMEA $PNORBT0 (same as DF350 but no tags) 354: NMEA $PNORBT3 including tags. 355: NMEA $PNORBT4 (same as DF354 but no tags) 356: NMEA $PNORBT6 including tags. 357: NMEA $PNORBT7 (same as DF356 but no tags) 358: NMEA $PNORBT8 (sensors) including tags. 359: NMEA $PNORBT9 (Sensors) (same as DF358 but no tags)
PL	Power level (range -20.0 dB to 0.0 dB, -100 dB to switch off transmit). This parameter is instrument dependent and dependent on the power consumption of the deployment as a whole. Unit: [dB] | Values: [-20; 0]; -100

WT	Measure Water Track velocity ON, OFF
WTDF	Water Track Data format 22: Nortek DVL Water Track data format 150: RDI PD0 - RDI documentation 154: RDI PD4 - RDI documentation 156: RDI PD6 - RDI documentation 404: NMEA $PNORWT3 including tags 405: NMEA $PNORWT4 (same as DF404 but no tags) 406: NMEA $PNORWT6 including tags 407: NMEA $PNORWT7 (same as DF406 but no tags) 408: NMEA $PNORWT8 (sensors) including tags 409: NMEA $PNORWT9 (sensors) (same as DF408 but no tags)
BD	Blanking distance Unit: [m]
PLMODE	When set to Max the power level is always maximum. The User setting will use the value set with PL. MAX, USER

Note: Note to powerLevelMode: Lower power is sometimes desirable if there is an interest in reducing power consumption or if the DVL will only be operating close to the bottom. The maximum power level is range dependent, so the user may either let the firmware select the maximum (MAX) given the current configuration or choose a value (USER). If USER is selected, a power level of 0 dB represents maximum power output. Power is decreased by entering negative values.

SETBT

Set bottom track settings

Example:
```
SETBT,RANGE=100.0
OK
GETBT
100.00,5.00,4,0,21,0.0,"ON",22,0.02,"MAX"
OK
```

GETBT

Get bottom track settings

Example:
```
GETBT
70.00,5.00,4,0,21,0.0,"ON",22,0.02,"MAX"
OK
```

GETBTLIM

Get bottom track setting limits

Example:

```
GETBTLIM
([2.00;200.00]),([5.00;5.00]),([0;4]),([0;4321]),
(21;150;154;156;350;351;354;355;356;357;358;359),([-20.00;0.00]),
("ON";"OFF"),(22;150;154;156;404;405;406;407;408;409),([0.02;10.00]),
("USER";"MAX")
OK
```

7.9 Dvl main settings

Commands: SETDVL, GETDVL, GETDVLLIM
Command type: CONFIGURATION
Mode: COMMAND

The DVL parameters specify which type(s) of measurements that will be measured and how the measurements are initiated/trigged.

Argument	Description
CP	Licensed feature. Collect Current Profile every Nth Bottom Track ping, 0 to disable Current Profile. 0, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20 Unit: [s]
TRIG	Specifies trigger type See integrators guide for a longer description on the triggers. "INTSR": Internal Sampling is available with rates from 1 to 8 Hz. For long ranges the maximum sampling rate is reduced. "TTLEDGES": TTL trigger on both rising- and falling edge of a TTL signal "TTLRISE": TTL trigger on rising edge of a TTL signal "TTLFALL": TTL trigger on falling edge of a TTL signal "RS485EDGES": Trigger on both edges of a RS-485 signal "RS485RISE": Trigger on rising edge of a RS-485 Signal "RS485FALL": Trigger on falling edge of a RS-485 Signal "SERIAL": When the TRIG parameter of the SETDVL command is set to "SERIAL" the DVL is trigged by sending a "TRIG[CrLf]" command. The trigger time will be when the end of the [CrLf] is received.
SR	Internal sampling rate if enabled. Unit: [Hz] | Values: [0; 8]
FN	Filename Validation: 30 characters, a-z, A-Z, 0-9, . and _

SV	Sound velocity in surrounding medium 0 will set sensor to use measured sound velocity Unit: [m/s] | Values: [1400; 1600]
SA	Salinity Unit: [ppt] | Values: [0; 50]
ALTI	Licensed feature. Collect Altimeter ping every Nth Bottom Track ping. 0 to disable. 0, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20
FASTTRIG	If enabled, DVL does not sleep between pings. If disabled, there is a partial power-down between pings.
MODE	The bottom detection mode allows for four different options for detecting the bottom. For more details on choosing modes, see Modes in chapter on Operation Principles. "NORMAL": For general DVL use. "CRAWLER": For aiding vehicles that intend to operate in station keeping mode or move very slowly. "AUTO": The DVL will automatically change between Normal and Crawler modes based on distance to the bottom as well as the measured velocity "FAST_ACQ": This mode is similar to Normal mode but it does not have an acquisition requirement to begin reporting bottom detection.

SETDVL

Set dvl main settings

Example:

```
SETDVL,TRIG="TTLRISE"
OK
GETDVL
GETDVL,0,"TTLRISE",4.0,"",0.0,35.0,0
OK
```

GETDVL

Get dvl main settings

Example:

```
GETDVL
GETDVL,0,"INTSR",4.0,"",0.0,35.0,0
OK
```

GETDVLLIM

Get dvl main setting limits

7.10 Fast pressure settings

Commands: SETFASTPRESSURE, GETFASTPRESSURE, GETFASTPRESSURELIM
Command type: CONFIGURATION
Mode: COMMAND **License:** FastPressure

Enables rapid pressure reading. The Fast pressure feature is a licensed mode for enabling rapid data output from the integrated pressure sensor. Note that higher data rates does not impact the resolution of the individual pressure estimates.

Argument	Description
EN	Enable/disable fast sampling of pressure. Default: 0
SR	Sampling rate if enabled. Values may be 10, 15 or 30 Hz 10, 15, 30 Unit: [Hz]
DF	Data format. 41: Binary format. Standard 10 byte binary header with ID = 0x41. Data part is only a 4-byte floating point number (float) giving the Depth below the Surface. 203: Standard depth below surface NMEA sentence.

SETFASTPRESSURE

Set fast pressure settings

Example:

```
SETFASTPRESSURE,EN=1,SR=15.0,DF=203
OK
```

GETFASTPRESSURE

Get fast pressure settings

Example:

```
GETFASTPRESSURE
GETFASTPRESSURE,EN,SR,DF
1,15,203
OK
```

GETFASTPRESSURELIM

Get fast pressure setting limits

Example:

```
GETFASTPRESSURELIM
GETFASTPRESSURELIM,(0;1),(10.0;15.0;30.0),(41;203)
OK
```

7.11 Trigger settings

Commands: SETTRIG, GETTRIG, GETTRIGLIM
Command type: CONFIGURATION
Mode: COMMAND

The parameters and limits for Trigger. The available trigger types will depend on the harness/cable used with the instrument.

TTL Trigger. The AD2CP can trig on either Rising Edge, Falling Edge or Both Edges of a TTL Signal. The requirements for the TTL input is Vlow < 0.7V and Vhigh > 2.5V. The TTL input tolerate voltages between 0-5.5 V.

A RS-485 signal can be used to trig the AD2CP, either Rising Edge, Falling Edge or Both Edges of a RS-485 Signal. The following figure shows the polarities of the differential RS-485 signal pair for the trigger types.

When triggered the instrument will perform a complete ping (Tx and Rx) before it goes back to monitoring the trigger. Any triggers asserted during an ongoing ping will be ignored. There are no specific requirements for pulse length. The unit triggers on the edge(s) of the trigger signal and can be triggered on rising, falling or both edges. The pulse length should be minimum 1 ms.

Argument	Description
EN	Enable/disable external trigger functionality. If EN=0 the internal Sampling is available with rates from 1 to 8 Hz. For long ranges the maximum sampling rate is reduced. If EN=1 the external sampling is enabled and given with the TRIG argument.
TRIG	Specifies trigger type and are given as counting numbers. The available trigger types will depend on the harness/cable used with the instrument. "TTLEDGES": Trigger on both rising- and falling edge of a TTL signal. "TTLRISE": Trigger on rising edge of a TTL signal. "TTLFALL": Trigger on falling edge of a TTL signal. "RS485EDGES": Trigger on both edges of a RS-485 signal. "RS485RISE": Trigger on rising edge of a RS-485 signal. "RS485FALL": Trigger on falling edge of a RS-485 signal. "COMMAND": Trigger on Command. When the TRIG parameter of the SETTRIG command is set to "COMMAND" the AD2CP is trigged by

	sending a "TRIG[CrLf]" command. The trigger time will be when the end of the [CrLf] is received
TRIGOUTPUT	TRIGOUTPUT=1 enables master trig output. RS485EDGE trigger must be used with this option. This enables several instruments to be synchronized together through RS485 with one of the instruments acting as master. Only continuous measurement configurations are supported in this mode, and all synchronized AD2CP instruments must be of the same frequency and have the same deployment configuration.

SETTRIG

Set trigger settings

Example:
```
SETTRIG,EN=1,TRIG="RS485EDGES",TRIGOUTPUT=0
SETTRIG,EN=0,TRIG="INTSR",TRIGOUTPUT=1
```

GETTRIG

Get trigger settings

GETTRIGLIM

Get trigger setting limits

7.12 Trigger settings for secondary plan

Commands: SETTRIG1, GETTRIG1, GETTRIGLIM1
Command type: CONFIGURATION
Mode: COMMAND

The parameters and limits for trigger for secondary plan. The available trigger types will depend on the harness/cable used with the instrument.

Argument	Description
EN	Enable/disable external trigger functionality. If EN=0 the internal Sampling is available with rates from 1 to 8 Hz. For long ranges the maximum sampling rate is reduced. If EN=1 the external sampling is enabled and given with the TRIG argument.
TRIG	Specifies trigger type and are given as counting numbers. The available trigger types will depend on the harness/cable used with the instrument. "TTLEDGES": Trigger on both rising- and falling edge of a TTL signal.

	"TTLRISE": Trigger on rising edge of a TTL signal. "TTLFALL": Trigger on falling edge of a TTL signal. "RS485EDGES": Trigger on both edges of a RS-485 signal. "RS485RISE": Trigger on rising edge of a RS-485 signal. "RS485FALL": Trigger on falling edge of a RS-485 signal. "COMMAND": Trigger on Command. When the TRIG parameter of the SETTRIG1 command is set to "COMMAND" the AD2CP is trigged by sending a "TRIG[CrLf]" command. The trigger time will be when the end of the [CrLf] is received
TRIGOUTPUT	TRIGOUTPUT=1 enables master trig output. RS485EDGE trigger must be used with this option. This enables several instruments to be synchronized together through RS485 with one of the instruments acting as master. Only continuous measurement configurations are supported in this mode, and all synchronized AD2CP instruments must be of the same frequency and have the same deployment configuration.

SETTRIG1

Set trigger settings for secondary plan

Example:
```
SETTRIG1,EN=1,TRIG="COMMAND",TRIGOUTPUT=0
```

GETTRIG1

Get trigger settings for secondary plan

GETTRIGLIM1

Get trigger setting limits for secondary plan

7.13 Current profile settings

Commands: SETCURPROF, GETCURPROF, GETCURPROFLIM
Command type: CONFIGURATION
Mode: COMMAND **License:** CurrentProfile

The CURPROF command configures the optional Current Profile measurements.

The valid range for the various arguments should be verified using the GETCURPROFLIM command, also for the values listed here as they may change with firmware versions and instrument frequencies. Current profile always use a broad bandwidth.

The current profile functionality is licensed option and is noted in by the license field "AVERAGE MODE" in the license list when a GETALL command is issued. A single cell is made available when the end user would like to perform a functionality test of the transducers (See DVL Operations Manual).

The current profile is not the same as Water Track; Water Track is always available with each Bottom Track ping.

Argument	Description
NC	Number of cells Values: [1; 200]
CS	Cell size Unit: [m] | Values: [0.5; 4]
BD	Blanking distance Unit: [m] | Values: [0.5; 68]
CY	Co-ordinate system BEAM, XYZ
PL	Power level (range -20.0 dB to 0.0 dB, -100 dB to switch off transmit). Unit: [dB] | Values: [-20; 0]; -100
VR	Velocity range along beam Unit: [m/s] | Values: [1; 5]
DF	Data format 3: Nortek Binary format 6: ?? 100: Prolog NMEA format 101: NMEA format 1 102: NMEA format 2 103: NMEA format 3 104: NMEA format 4 150: RDI PD0, see RDI documentation
NB	Number of beams. Select number of beams. Maximum number of beams equals number of transducers on sensor head.
CH	Beam selection. E.g. CH=123 will enable beams 1, 2 and 3.

Note: The actual valid range for the various parameters for the firmware version is used can be found by using the GETCURPROFLIM command. This command has the same arguments as the SETCURPROF/GETCURPROF commands shown in the list above. The output format for limits is described in Data Limit Formats.

SETCURPROF

Set current profile configuration

GETCURPROF

Get current profile configuration

GETCURPROFLIM

Get current profile configuration limits

7.14 Altimeter settings

Commands: SETALTI, GETALTI, GETALTILIM
Command type: CONFIGURATION
Mode: COMMAND **License:** Altimeter

The altimeter is an optional hardware configuration available for select Nortek custom DVLs. The ALTI command configures the optional Altimeter measurements, and is enabled with the SETDVL command. The valid range for the various arguments should be verified using the GETALTILIM command., also for the values listed here as they may change with firmware versions and instrument frequencies.

Argument	Description
BD	Blanking distance Unit: [m] | Values: [0.5; 5]
RANGE	Range Unit: [m] | Values: [1; 175]
PL	Power level (range -20.0 dB to 0.0 dB, -100 dB to switch off transmit). Unit: [dB] | Values: [-20; 0]; -100
DF	Data format 30: Nortek altimeter data format 200: NMEA $PNORA without tags 201: NMEA $PNORA including tags 202: NMEA $SDDBT

Note: The actual valid range for the various parameters for the firmware version is used can be found by using the GETALTILIM command. This command has the same arguments as the SETALTI/GETALTI commands shown in the list above. The output format for limits is described in Data Limit Formats.

SETALTI

Set altimeter configuration

GETALTI

Get altimeter configuration

GETALTILIM

Get altimeter configuration limits

7.15 Instrument user settings

Commands: SETUSER, GETUSER, GETUSERLIM
Command type: CONFIGURATION
Mode: COMMAND

The SAVE,USER command must be sent to save changes in USER parameters.

Argument	Description
POFF	Set the offset value of the pressure sensor. Unit: [dbar] | Values: [0; 11]
ROTXY	Alignment offset. Equivalent to the Webpage Mounting Angle Unit: [deg] | Values: [-180; 180]
MA11	
MA12	
MA13	
MA21	
MA22	
MA23	
MA31	
MA32	
MA33	

Note: A rotation of the XY coordinate system (about the z-axis) can be done using the ROTXY parameter.

The new coordinate system X'Y'Z' is given by the following:

[X'Y'Z'] = [[cosd(rotxy),-sind(rotxy),0],[sind(rotxy),cosd(rotxy),0],[0,0,1]]*[X,Y,Z]

SETUSER

Set instrument user settings

Example:
```
SETUSER,ROTXY=15.0
OK
```

```
SAVE,USER
OK
```

GETUSER

Get instrument user settings

GETUSERLIM

Get instrument calibration parameter limits

7.16 Additional online data formats

Commands: SETAODF, GETAODF, GETAODFLIM
Command type: CONFIGURATION
Mode: COMMAND

The additional online data formats provide a means of producing two independent data streams with user selectable data formats that are different from those used in the main stream (e.g. the main stream could contain Nortek formatted data while the AODF streams could contain PDx formatted data). These two data streams (AODFA and AODFB) are only available when Ethernet is used for communications. The streams can be read from a client computer by either connecting to the associated instrument TCP/IP port (9010 for AODFA and 9011 for AODFB) or by configuring and enabling the associated UDP/IP data client information for the stream.

Each stream (AODFA/AODFB) can contain Bottom Track, Water Track, Current Profile and/or Altimeter data with the selected format. If UDP transmission is selected and configured, the instrument will automatically transmit UDP data records to the client address / port as it is received.

Argument	Description
DFBTA	AODFA – Data format for Additional Bottom Track Data
DFWTA	AODFA – Data format for Additional Water Track Data
DFCPA	AODFA – Data format for Additional Current Profile Data
DFALTIA	AODFA – Data format for Additional Altimeter Data
DFBTB	AODFB – Data format for Additional Bottom Track Data
DFWTB	AODFB – Data format for Additional Water Track Data
DFCPB	AODFB – Data format for Additional Current Profile Data
DFALTIB	AODFB – Data format for Additional Altimeter Data
DFBTRS	AODFRS – Data format for Additional Bottom Track Data
DFWTRS	AODFRS – Data format for Additional Water Track Data

DFCPRS	AODFRS – Data format for Additional Current Profile Data
DFALTIRS	AODFRS – Data format for Additional Altimeter Data

SETAODF

Set additional online data formats

GETAODF

Get additional online data formats

GETAODFLIM

Get available additional online data formats

7.17 Get instrument ID

Command: ID
Command type: INFO
Mode: COMMAND

Commands for accessing instrument name and serial number

Argument	Description
SN	Serial number Values: [0; 2147483647]
STR	Instrument name

Example:
```
ID
"Signature1000",900002
ID,STR
"Signature1000"
```

7.18 Reload default settings

Command: SETDEFAULT
Command type: ACTION
Mode: COMMAND

Reload default settings.

Argument	Description
CONFIG	Restore all settings below except USER and INST to default values. Legacy argument ALL acts as CONFIG.
CP	Restore Current Profiling Mode Settings (CURPROF) to default.
INST	Restore instrument main settings to default.
BT	Restore bottom track settings to default.
USER	Restore user calibration to default.
DVL	Restore DVL settings to default.

Example:
```
SETDEFAULT,CONFIG
```

7.19 Save settings

Command: SAVE
Command type: ACTION
Mode: COMMAND

Save current settings for next measurement. At least one argument must be specified for the SAVE command.

Argument	Description
CONFIG	Save all settings except INST and USER settings. Legacy argument ALL acts as CONFIG.
CP	Save Current Profiling Mode Settings (CURPROF).
INST	Save INST settings.
USER	Save user instrument settings.
BT	Save bottom track settings.
DVL	Save DVL settings.

Example:
```
SAVE,CONFIG
```

7.20 Enter command mode

Command: MC
Command type: ACTION
Mode: CONFIRMATION

Sets instrument in command mode from confirmation mode.

Example:
`MC`

7.21 Data retrieval mode

Command: RM
Command type: ACTION
Mode: CONFIRMATION

Go into data retrieval mode from confirmation mode and access the instruments recorder.

Example:
`RM`

7.22 Enter measurement mode

Command: CO
Command type: ACTION
Mode: CONFIRMATION, RETRIEVAL

Continue in measurement mode from confirmation mode or data retrieval mode. Instrument returns to collecting data according to the current configuration.

Example:
`CO`

7.23 Power down

Command: POWERDOWN
Command type: ACTION
Mode: COMMAND

Power down the instrument to set it in sleep mode.

Example:
`POWERDOWN`

7.24 Erase files on recorder

Command: ERASE
Command type: ACTION
Mode: COMMAND

Erase all files on the recorder

Argument	Description
CODE	Code should be 9999 9999

Example:
`ERASE,CODE=9999`

7.25 Format recorder

Command: FORMAT
Command type: ACTION
Mode: COMMAND

Format the recorder. Note that this can take minutes depending on the recorder size.

Argument	Description
CODE	Code should be 9999 9999

Example:
`FORMAT,CODE=9999`

7.26 Read configuration

Command: READCFG
Command type: CONFIGURATION
Mode: COMMAND

Read current configuration. If a filename parameter is given the current configuration is stored to a file. If no parameter is given the current configuration is written to console.

The output of this command can be used to configure the instrument to a known configuration.

Argument	Description
FN	Write the output to file. Values: ['a'; 'z']; ['A'; 'Z']; ['0'; '9']; '_'; '.' | Max Length: 30

Example:
```
READCFG
SETDEFAULT,CONFIG
SETDVL,CP=8,TRIG="INTSR",SR=8.0,FN="",SV=0.0,SA=35.0
SETCURPROF,NC=60,CS=0.50,BD=0.10,CY="XYZ",PL=0.0,VP=0.000,VR=2.50,DF=
3,NB=4,CH=0
```

```
SETBT,RANGE=30.00,VR=5.00,NB=4,CH=0,DF=21,PL=-
2.0,WT="ON",WTDF=22,BD=0.01,PLMODE="MAX"
SAVE,CONFIG
OK
```

7.27 Inquire state

Command: INQ
Command type: INFO
Mode: COMMAND, CONFIRMATION, RETRIEVAL, MEASUREMENT

The INQ command queries the instrument state. Note that when operating over RS232 or RS422 serial lines, it should be preceded with @@@@@@ <delay 400ms> and a flush of the input buffer in case the instrument is in power down or in a low power mode taking measurements.

Consult section "Modes" for a description of the Instrument modes.

Note: Parameters used to represent instrument mode:

0000: Bootloader/Firmware upgrade

0001: Measurement

0002: Command

0004: Data Retrieval

0005: Confirmation

0006: FTP-mode

Example:
(In command mode)
```
INQ
0002
```
(In measurement mode)
```
INQ
0001
```
(In confirmation mode)
```
INQ
0005
```
(In data retrieval mode)
```
INQ
0004
```
(In firmware upgrade mode)
```
INQ
0000
```

7.28 Get error

Command: GETERROR
Command type: INFO
Mode: COMMAND, CONFIRMATION, MEASUREMENT, RETRIEVAL

GETERROR retrieves a full description of the last error condition to occur. The error number is returned first followed by a string with the text description of the last error condition. A second string is also returned which contains information on the valid range of the failing argument.

Argument	Description
NUM	Integer error value
STR	Text description

Example:

```
SETDVL,sa=90.0
OK
SAVE,CONFIG
ERROR
GETERROR
310,"Invalid setting: DVL Salinity","GETDVLLIM,SA=([0.0;50.0])"
OK
```

7.29 Get error number

Command: GETERRORNUM
Command type: INFO
Mode: COMMAND, CONFIRMATION, MEASUREMENT, RETRIEVAL

GETERRORNUM retrieves the integer error value for the last error condition.

7.30 Get error string

Command: GETERRORSTR
Command type: INFO
Mode: COMMAND, CONFIRMATION, MEASUREMENT, RETRIEVAL

GETERRORSTR retrieves the string description for the last error condition.

7.31 Get all

Command: GETALL
Command type: INFO
Mode: COMMAND

GETALL retrieves all relevant configuration information for the instrument. This information can either be displayed on the command line or saved to a data file on the instrument. For the Nortek post-processing software to read a valid .ad2cp file it must contain both the Header and Data Record. The Header information can be obtained by using the command GETALL.

Argument	Description
FN	Write the output to this file saved on instrument

Example:

```
GETALL
GETPLAN,600,1,0,0,10,0.0,1,0,0,1500,"",1
GETAVG,20,1.00,0.30,"BEAM",-12.0,1,0.000,1.29,3,1,0,0
GETBURST,50,4,0.400,0.200,"BEAM",0.0,1,1024,4.00,0.000,0,1,0
GETUSER,0.00,0.00,0,0,0
GETINST,9600,232,1
BEAMCFGLIST,1,10.00,20.00,1000,500,1,1
BEAMCFGLIST,2,10.00,20.00,1000,500,1,2
BEAMCFGLIST,3,10.00,20.00,1000,500,1,3
BEAMCFGLIST,4,10.00,20.00,1000,500,1,4
OK
```

7.32 Get recorder state

Command: RECSTAT
Command type: INFO
Mode: COMMAND, RETRIEVAL

Returns recorder state.

Argument	Description
SS	Number of bytes in a sector. Unit: [bytes]
CS	Number of bytes in one cluster. Unit: [bytes]
FC	Number of bytes in free clusters. Unit: [bytes]
TC	Total number of bytes in clusters. Unit: [bytes]
VS	Volume Size in bytes. Unit: [bytes]

Example:

```
RECSTAT,VS
```

7.33 Write tag output

Command: TAG
Command type: ACTION
Mode: COMMAND, CONFIRMATION, MEASUREMENT, RETRIEVAL

Write a Tag to output file and data output.

The TAG command adds a tag to the both the output file and the output data, if enabled. The output is a String Record as defined in the FWRITE command. The ID of the String Record Data packet is 19dec.

Argument	Description
STR	Tag string. Maximum 200 bytes.
CLK	Add clock in tag.

Note: Remember to use the BBPWAKEUP command when sending commands to an instrument in Measurement mode, when using Ethernet.

Example:
```
TAG,STR="This is a test tag.",CLK=1
a5 0a a0 10 2f 00 42 8c 42 5d 13 32 // Binary header
30 31 37 2d 30 31 2d 32 34 20 30 38 // String Record ID = 19dec
3a 34 32 3a 35 37 2e 34 34 39 20 2d // "2017-01-24 08:42"
20 54 68 69 73 20 69 73 20 61 20 74 // ":57.449 - This i"
65 73 74 20 74 61 67 2e 00 // "s is a test tag."
OK
```

7.34 Precision time protocol

Commands: PTPSET, PTPGET
Command type: CONFIGURATION
Mode: COMMAND

Configure the precision time protocol parameters.

Argument	Description
EN	Choose a time protocol. 0: Time protocol disabled. 1: Use precision time protocol (PTP). 2: Use network time protocol (NTP).
OFFSET	Offset Unit: [µs]
CL	Time value representing the number of seconds elapsed since 00:00 hours, Jan 1, 1970 UTC Unit: [s]

IP	The numeric IP address of the NTP server to use for syncing

Note: For the case of NTP, the instrument is not capable of DNS name resolution. In order to retrieve the IP address associated with a name, use the "nslookup" tool from the computer command line

PTPSET

Set precision time protocol parameters

Example:
```
PTPSET,EN=0,OFFSET=0,CL=1609193402
OK
PTPSET,EN=2,IP="129.240.3.3"
OK
```

PTPGET

Get precision time protocol parameters

Example:
```
PTPGET,EN,OFFSET,CL
```

7.35 Get hardware specifications

Command: GETHW
Command type: INFO
Mode: COMMAND, CONFIRMATION, MEASUREMENT, RETRIEVAL

Returns Firmware versions and Board revisions.

Argument	Description
FW	Running DSP FW version
FWMINOR	Running DSP FW version (minor part)
FPGA	Running FPGA FW version
BOOT	DSP bootloader FW Version
DIGITAL	Board revision. Example: C-0
INTERFACE	Board revision. Example: C-0
ANALOG	Board revision. Example: C-0
ANALOG2	Board 2 revision. Example: C-0
SENSOR	Board revision. Example: C-0

Example:
```
GETHW,FW,MINOR
```

7.36 Add license

Command: ADDLICENSE
Command type: PRODUCTION
Access: User
Mode: COMMAND

Reads a license key and checks it against the instrument serial number before adding it to the license key flash.

Argument	Description
KEY	The license key to add

Example:
```
ADDLICENSE,KEY="9H3F5PE47HUUB"
OK
```

7.37 Delete license

Command: DELETELICENSE
Command type: PRODUCTION
Access: User
Mode: COMMAND

Deletes a license key from the license key flash.

Argument	Description
KEY	The license key to delete

Example:
```
DELETELICENSE,KEY="9H3F5PE47HUUB"
OK
```

7.38 Lists license keys

Command: LISTLICENSE
Command type: INFO
Access: User
Mode: COMMAND

Lists all license keys contained in the license key flash along with a description of the functionality enabled by the key and the variant number of the license.

Argument	Description
KEY	License key
DESC	Information about license type
TYPE	License id

Example:

```
LISTLICENSE
LISTLICENSE,"K28FDJF7RPNUB","All features",0
LISTLICENSE,"4X218TRTRPNUB","High Resolution",4
LISTLICENSE,"JKHHFNH3RPNUB","Wave Mode",6
LISTLICENSE,"WF3CJR6PRPNUB","Current Profiler",1
OK
$PNOR,LISTLICENSE*76
$PNOR,LISTLICENSE,KEY="K28FDJF7RPNUB",DESC="All features",TYPE=0*4C
$PNOR,LISTLICENSE,KEY="4X218TRTRPNUB",DESC="High
Resolution",TYPE=4*73
$PNOR,LISTLICENSE,KEY="JKHHFNH3RPNUB",DESC="Wave Mode",TYPE=6*00
$PNOR,LISTLICENSE,KEY="WF3CJR6PRPNUB",DESC="Current
Profiler",TYPE=1*1C
$PNOR,OK*2B
```

8 Data Formats

This chapter describes the Nortek DVL binary data formats for sensor output. Note that the binary data formats all use a common header that specifies how the rest of the data block should be interpreted. A data block is the data from and including one header to the next. Binary data are always sent as Little Endian.

About these chapters

Each sensor's data format is described in the following chapters. To avoid duplicating rows in the following tables, we have documented header and common data separately. This way, the chapter on one sensors data format will only contain the fields unique or this sensor. Take velocity data as an example:

In short: The data format is the sum of header data, a common part, a part that is shared for current data and the part that is unique for velocity. Se figure below.

A little longer: The header is the same for all data blocks. It is compact and quick to parse, and it contains information about the rest of the data (e.g. data type and size). This is documented separately as _HeaderData. We use the leading underscore to emphasize that this is a not a complete data format, but it is a part used by two or more data formats.

The same goes for other common data such as data format version number, offset to data and timestamp etc. This is documented separately in _CommonData. Note that for velocity data there is another current profile definition that is shared.

Last, there are the unique fields such as beam configuration, velocity data, amplitude data etc. that is given in the table in VelocityDataV3.

The table below is an illustration on how common data fields (gray for header and blue for other common's) relate to the sensor specific data fields (green).

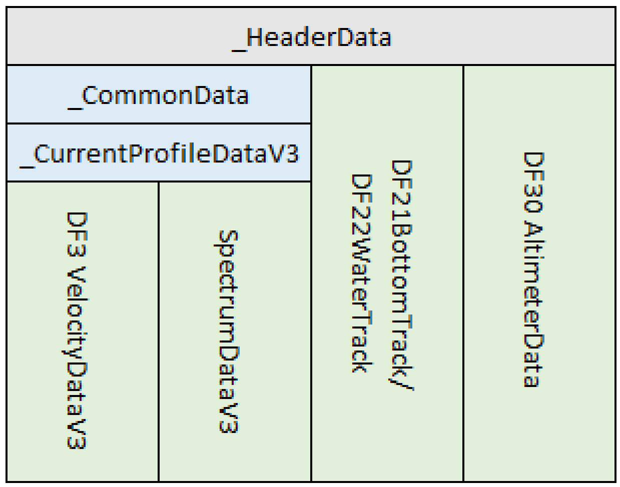


Figure: Showing how common data fields (gray for header and blue for other common's) the sensor data relate to the sensor specific data fields (green). Note that we use a leading underscore (_) to emphasize that this is not a sensor data format but is common and used by two or more data formats.

About the tables

Tables have the columns 'Field', 'Position/Size' and 'Description'. Position and size may need an explanation:

Position has the location of a field in the header or in the data that follows the header. E.g., the 'data series id' has position 2 (Note that we are counting from 0) in the header. Some positions are not fixed, but dependent on which fields are before it. In these cases, 'offset of data' (position 1 of the data - see _CommonData) can then be used to give the position of the following fields. In these cases, the position in the table will not be given as a number but as a variable name such as OFFSET. Variable descriptions are listed below the tables where they are used.

Size is the data type of field. In case of 'data series id' it is an unsigned integer of 8 bits (uint8). Note that not all fields have a specific data type but is an object using a required number of bits. E.g., the status bit masks often use 32 bits to provide 'ok'/'not ok' on several parts of the data. These object sizes and their descriptions are listed below the table where they are used.

8.1 _HeaderData

The header definition for binary data formats. Note that the header may be verified without reading the rest of the data block since it has its own checksum.

Field	Position Size	Description
Sync bit	0 uint8	Always 0xA5.
Header size	1 uint8	Number of bytes in the headers. Normally it is 10 bytes, but in a few cases it may be 12 bytes to hold data size of 32 bytes.
Data series id	2 uint8	Defines the type of the following Data Record. 0x15 - Burst Data Record. 0x16 - Average Data Record. 0x17 - Bottom Track Data Record. 0x18 - Interleaved Burst Data Record (beam 5). 0x21 - Altimeter Data Record. 0x1B - DVL Bottom Track Record. 0x1D - DVL Water Track Record. 0xA0 - String Data Record, eg. GPS NMEA data, comment from the FWRITE command.
Family id	3 uint8	Defines the Instrument Family. 0x10 is the signature Family.
Data size	4 unit16/uint32	Number of bytes in the following Data Record. If header size is 10, the data size is represented with a uint16. For large datasets, header may have 12 bytes giving room for a uint32 to represent data size.
Data checksum	6/8 uint16	Checksum of the following Data Record.
Header checksum	8/10 uint16	Checksum of all fields of the Header except the Header Checksum itself.

8.2 _CommonData

Used By: _CurrentProfileDataV3

Common data definitions for parsing SIGNATURE V3 average- and echosounder data.

Field	Position Size	Description

Version	0 uint8	Version number of the Data Record Definition. Should be 3
Offset of data	1 uint8	Number of bytes from start of the record to start of the actual data. Unit: [# bytes]
Configuration bit mask	2 16 bits	Record Configuration Bit Mask Object reference given in table below
Serial number	4 uint32	Instrument serial number from factory.
Date and time	8 uint8 *8	The date and time of the data record. Year: Is given as years from 1900. Month: January is 0. Milli seconds: Are given by the hundreds. That is as desi seconds.
Micro seconds	12 uint16	Remaining micro seconds (Date object has milliseconds resolution)
Speed of sound	16 uint16	Speed of sound used by the instrument. Raw data given as 0.1 m/s Unit: [m/s]
Temperature	18 int16	Reading from the temperature sensor. Raw data given as 0.01 °C Unit: [°C]
Pressure	20 uint32	Raw data given as 0.001 dBar Unit: [dBar]
Heading	24 uint16	Raw data given as 0.01 degrees Unit: [deg]
Pitch	26 int16	Raw data given as 0.01 degrees Unit: [deg]
Roll	28 int16	Raw data given as 0.01 degrees Unit: [deg]
Cell size	32 uint16	Size of each cell (resolution) on the beam. Raw data given as mm Unit: [m]
Nominal correlation	36 uint8	The nominal correlation for the configured combination of cell size and velocity range Unit: [%]
Battery voltage	38 uint16	Raw value given in 0.1 Volt Unit: [V]
Magnetometer.X	40 int16	X axis flux raw value in last measurement interval
Magnetometer.Y	42	Y axis flux raw value in last measurement interval

	int16	
Magnetometer.Z	44 int16	Z axis flux raw value in last measurement interval
Accelerometer.X	46 int16	Raw accelerometer X axis value in last measurement interval Raw value divided by 16384 will give vector [x,y,z] of length 1 Note: The unit of the instrument is gravity [g]. Conversion of Accelerometer unit less raw measurements to m/s^2: divide measurement by 16384, then multiply by calibrated gravity in Oslo, 9.819 m/s^2.
Accelerometer.Y	48 int16	Raw Y axis value in last measurement interval Raw value divided by 16384 will give vector [x,y,z] of length 1
Accelerometer.Z	50 int16	Raw Z axis value in last measurement interval. Raw value divided by 16384 will give vector [x,y,z] of length 1
Data set description	54/56 uint16	Data set description. 0-3 Physical beam used for 1st data set. 4-7 Physical beam used for 2nd data set. 8-11 Physical beam used for 3th data set. 12-16 Physical beam used for 4th data set.
Transmitted energy	56/58 uint16	Transmitted energy.
Velocity scaling	58/60 int8	Velocity scaling used to scale velocity data.
Power level	59/61 int8	Configured power level Unit: [dB]
Magnetometer temperature	60/62 int16	Magnetometer temperature reading. Uncalibrated Raw data in 1/1000 °C Unit: [°C]
Real time clock temperature	62/64 int16	Real Time Clock temperature reading Unit: [°C]
Error status	64/66 16 bits	Error bit mask Object reference given in table below
Status	68/70 32 bits	Status bit mask Object reference given in table below
Ensemble counter	72/74 uint32	Counts the number of ensembles in both averaged and burst data

Position and size variables:

Name	Description
54/56	The status field is at 54 or 56 depending on wether the ambiguity velocity is 16bit or 32bit long.
56/58	The status field is at 56 or 58 depending on wether the ambiguity velocity is 16bit or 32bit long.
58/60	The status field is at 56 or 58 depending on wether the ambiguity velocity is 16bit or 32bit long.
59/61	The status field is at 59 or 61 depending on wether the ambiguity velocity is 16bit or 32bit long.
60/62	The status field is at 60 or 62 depending on wether the ambiguity velocity is 16bit or 32bit long.
62/64	The status field is at 62 or 64 depending on wether the ambiguity velocity is 16bit or 32bit long.
64/66	The status field is at 64 or 66 depending on wether the ambiguity velocity is 16bit or 32bit long.
68/70	The status field is at 68 or 70 depending on wether the ambiguity velocity is 16bit or 32bit long.
72/74	The status field is at 72 or 74 depending on wether the ambiguity velocity is 16bit or 32bit long.

Object reference: Configuration bit mask

Record Configuration Bit Mask

Field	Position Size	Description
Has pressure sensor	0 bit	Pressure sensor value valid
Has temperature sensor	1 bit	Temperature sensor value valid
Has compass sensor	2 bit	Compass sensor value valid
Has tilt sensor	3 bit	Tilt sensor value valid

Object reference: Error status

Error bit mask

Field	Position	Description

	Size	
Data retrieval FIFO error	0 bit	Data retrieval FIFO error
Data retrieval overflow	1 bit	Data retrieval overflow
Data retrieval underrun	2 bit	Data retrieval Underrun
Data retrieval samples missing	3 bit	Data retrieval samples missing
Measurement not finished	4 bit	The Measurement and data storage/transmit didn't finish before next measurement started.
Sensor read failure	5 bit	Sensor read failure
Tag error beam 1 (In-phase)	8 bit	Tag error beam 1 (In-phase)
Tag error beam 1 (Quadrature-phase)	9 bit	Tag error beam 1 (Quadrature-phase)
Tag error beam 2 (In-phase)	10 bit	Tag error beam 2 (In-phase)
Tag error beam 2 (Quadrature-phase)	11 bit	Tag error beam 2 (Quadrature-phase)
Tag error beam 3 (In-phase)	12 bit	Tag error beam 3 (In-phase)
Tag error beam 3 (Quadrature-phase)	13 bit	Tag error beam 3 (Quadrature-phase)
Tag error beam 4 (In-Phase)	14 bit	Tag error beam 4 (In-phase)
Tag error beam 4 (Quadrature-phase)	15 bit	Tag Error Beam 4 (Quadrature-phase)

Object reference: Status

Status bit mask

Field	Position Size	Description
Wake up state	0	Bit 31-28: Wakeup State

	uint32	1111 0000 0000 0000 0000 0000 0000 0000 00 = bad power 01 = power applied 10 = break 11 = RTC alarm
Orientation	0 uint32	Bit 27-25: Orientation 0000 1110 0000 0000 0000 0000 0000 0000 0: "XUP" Instrument x-axis defined up, heading reference axis is Z positive 1: "XDOWN" Instrument x-axis defined down, heading reference axis is Z positive 2: "YUP" Instrument y-axis defined up, heading reference axis is Z positive 3: "YDOWN" Instrument y-axis defined down, heading reference axis is Z positive 4: "ZUP" Instrument z-axis defined up, heading reference axis is X positive 5: "ZDOWN" Instrument z-axis defined down, heading reference axis is X positive 7: "AHRS" AHRS reports orientation any way it points. Example: Z down -> Roll = 180 deg.
Auto orientation	0 uint32	Bit 24-22: autoOrientation 0000 0001 1100 0000 0000 0000 0000 0000 0: "Fixed" Fixed orientation 1: "Auto" Auto Up Down 3: "AHRS3D" AHRS3D

8.3 _CurrentProfileDataV3

Extends: _CommonData
Used By: DF3 VelocityDataV3, SpectrumDataV3

Data definitions for parsing SIGNATURE V3 average data.

Field	Position Size	Description
Temperature PressureSensor	37 uint8	Temperature of pressure sensor: T=(Val/5)-4.0 Raw value given as 0.2 °C Unit: [°C]
Blanking	34 uint16	Distance from instrument to first data point on the beam. Raw data given as cm or mm depending on status.blankingDistanceScalingInCm Unit: [m]

Ambiguity Velocity	52 uint16	Ambiguity velocity, corrected for sound velocity, scaled according to Velocity scaling. 10^(Velocity scaling) m/s Unit: [m/s]
Extended status	66 16 bits	Extended status bit mask Object reference given in table below
Status	68 32 bits	Status bit mask. Note that bits 0, 2, 3, 4 are unused. Object reference given in table below

Object reference: Extended status

Extended status bit mask

Field	**Position Size**	**Description**
Processor idles < 3%	0 bit	Indicates that the processor Idles less than 3 percent
Processor idles < 6%	1 bit	Indicates that the processor idles less than 6 percent
Processor idles < 12%	1 bit	Indicates that the processor idles less than 12 percent
Extended status should be interpreted	15 bit	If this bit is set the rest of the word/ extended status should be interpreted

Object reference: Status

Status bit mask. Note that bits 0, 2, 3, 4 are unused.

Field	**Position Size**	**Description**
Active configuration	16 bit	Bit 16: Active configuration 0: Settings for PLAN,BURST,AVG 1: Settings for PLAN1,BURST1,AVG1
Previous wakeup state	21-18 4 bits	Bit 21-18: Previous wakeup state 0000 0000 0011 1100 0000 0000 0000 0000 00 = bad power 01 = power applied 10 = break 11 = RTC alarm
Previous measurement	17 bit	Bit 17: Last measurement low voltage skip 0: normal operation

skipped due to low voltage		1: last measurement skipped due to low input voltage
Echosounder index	15-12 4 bits	Bit 15-12: Echosounder frequency index. Valid numbers are 0, 1 and 2 (or 0000, 0001 and 0010) referring to frequencies 1, 2 or 3 as used in SET-/GETECHO. 0000 0000 0000 0000 1111 0000 0000 0000
Telemetry data	11 bit	Telemetry data
Boost running	10 bit	Boost running
Echosounder frequency bin	9-5 5 bit	Bit 9-5: Echosounder frequency bin. Used only on the Signature100 which supports up to 5 packages pr frequency index. Processing is then done on the different frequency bands. 0000 0000 0000 0000 0000 0011 1110 0000
Blanking distance scaling in cm	1 bit	Bit 1: Scaling of blanking distance 0: mm scaling 1: given in cm

8.4 DF3 VelocityDataV3

Extends: _CurrentProfileDataV3
ID: 0x15, 0x16, 0x18, 0x1e, 0x1a, 0x1f

Data definitions for parsing AD2CP V3 average data.

Field	Position Size	Description
Configuration	2 16 bits	Record Configuration Bit Mask Object reference given in table below
Beams, coordinates and cells.Coordinate system	30 uint16	Bit 11-10 (2 bits): Coordinate system b00:ENU, b01:XYZ, b10:BEAM
Beams, coordinates and cells.Number of beams	30 uint16	Bit 15–12 (4 bits): Number of Beams (NB) Active beams represented as a 4 char string of 1s and 0s.
Beams, coordinates and cells.Number of cells	30 uint16	Bit 9-0 (10 bits): Number of Cells (NC).
Velocity data	OFFSET int16 *VEL_NB *VEL_NC	This field exists if the Velocity data included bit of the Config byte is set. 10^(Velocity Scaling)

		Unit: [m/s]
Amplitude data	AMP_POS int8 *AMP_NB *AMP_NC	This field exists if the amplitude data included bit of the Config byte is set 0.5 dB/count Unit: [dB]
Correlation data	CORR_POS int8 *CORR_NB *CORR_NC	This field exists if the Correlation data included bit of the Config byte is set [0 – 100 %] Unit: [%]
Altimeter data.Altimeter distance	ALTI_POS float	Distance to surface from Leading Edge algorithm Unit: [m]
Altimeter data.Altimeter quality	ALTI_POS + 4 uint16	Result of LE algorithm. When quality is deemed too low according to instrument specific limits. Distance and Quality set to 0.
Altimeter data.Altimeter status	ALTI_POS + 6 16 bits	Altimeter status bit mask
AST data.AST distance	AST_POS float	Distance to surface from Max Peak algorithm Unit: [m]
AST data.AST quality	AST_POS + 4 uint16	Amplitude at which surface is detected Raw data in steps of 0.01 dB, i.e. quality of 8000 = 80 dB Unit: [dB]
AST data.AST offset	AST_POS + 6 int16	Offset in step of measurement to velocity measurement Raw data given in 100 µs Unit: [s]
AST data.AST pressure	AST_POS + 8 float	Pressure value measured during the AST/altimeter ping Unit: [dbar]
Altimeter raw data.Num RawSamples	ALTIRAW_START + 8 uint16	Altimeter Raw Data – Number of Samples
Altimeter raw data.Samples distance	ALTIRAW_START + 10 uint16	Distance between samples Raw data given in 0.1mm Unit: [m]
Altimeter raw data.Data samples	ALTIRAW_START + 12 int16 *NRS	Altimeter Raw Data – Samples Raw data given as 16 bits Signed fract
AHRS data.Rotation matrix	AHRS_START float *3 *3	AHRS Rotation Matrix [3x3]

AHRS data.Quaternion W	AHRS_START + 36 float	W quaternion
AHRS data.Quaternion X	AHRS_START + 40 float	X quaternion
AHRS data.Quaternion Y	AHRS_START + 44 float	Y quaternion
AHRS data.Quaternion Z	AHRS_START + 48 float	Z quaternion
AHRS data.Gyro X	AHRS_START + 52 float	Gyro in X direction in degrees pr second Unit: [dps]
AHRS data.Gyro Y	AHRS_START + 56 float	Gyro in Y direction in degrees pr second Unit: [dps]
AHRS data.Gyro Z	AHRS_START + 60 float	Gyro in Z direction in degrees pr second Unit: [dps]
Percentage good data	PGD_START uint8 *PGD_LEN	Percent Good Estimate per cell These fields exist if the Percentage Good data included Unit: [%]
Standard deviation data.Pitch	SD_START int16	Standard deviation on pitch data Raw data in 0.01 degrees Unit: [deg]
Standard deviation data.Roll	SD_START + 2 int16	Standard deviation on roll data Raw data in 0.01 degrees Unit: [deg]
Standard deviation data.Heading	SD_START + 4 int16	Standard deviation on heading data Raw data in 0.01 degrees Unit: [deg]
Standard deviation data.Pressure	SD_START + 6 int16	Standard deviation on pressure data Raw data in 0.001 Bar Unit: [bar]

Position and size variables:

Name	Description
VEL_NB	Primary dimension of velocity data is number of beams. Length 0 if correlation data in configuration bit map is false.
VEL_NC	Second dimension of velocity data is number of cells pr beam. Length 0 if correlation data in configuration bit map is false.
OFFSET	Offset of data given at position 1 in this dataset. Number of bytes from start of record to start of data.

AMP_NB	Primary dimension of amplitude data is number of beams. Length 0 if correlation data in configuration bit map is false.
AMP_NC	Second dimension of amplitude data is number of cells pr beam. Length 0 if correlation data in configuration bit map is false.
AMP_POS	Correlation data starts after the amplitude data.
CORR_NB	Primary dimension of correlation data is number of beams. Length 0 if correlation data in configuration bit map is false.
CORR_NC	Second dimension of correlation data is number of cells pr beam. Length 0 if correlation data in configuration bit map is false.
CORR_POS	Correlation data starts after the amplitude data.
ALTI_POS	Altimeter data starts after the correlation data.
AST_POS	AST data starts after the altimeter data.
NRS	Number of raw samples given as first element of this object.
ALTIRAW_START	Altimeter raw data starts after the AST data.
AHRS_START	AHRS data follows the altimeter raw data.
PGD_START	Percent good data follows the AHRS data.
SD_START	The standard deviation data follows percent good data.

Object reference: Configuration

Record Configuration Bit Mask

Field	Position Size	Description
Has velocity data	5 bit	Velocity data included
Has amplitude data	6 bit	Amplitude data included
Has correlation data	7 bit	Correlation data included
Has altimeter data	8 bit	Altimeter data included
Has altimeter raw data	9 bit	Altimeter raw data included
Has AST data	10 bit	AST data included
Has AHRS data	12 bit	AHRS data included

Has percentage good data	13 bit	Percentage data included
Has standard deviation data	14 bit	Standard deviation data included

8.5 SpectrumDataV3

Extends: _CurrentProfileDataV3
ID: 0x20

Data definitions for parsing SIGNATURE V3 amplitude spectrum data.

Field	Position Size	Description
Configuration	2 16 bits	Record configuration bit mask. Object reference given in table below
Beams and bins.Number of beams	30 uint16	Bit 15–13 (3 bits) represent number of beams (NB) 1110 0000 0000 0000 Active beams represented as string of 1s and 0s.
Beams and bins.Number of bins	30 uint16	Bit 0-12 (13 bits) represents number of bins (NB) 0001 1111 1111 1111
Spectrum data.Start frequency	OFFSET float	Start frequency value Unit: [Hz]
Spectrum data.Step frequency	OFFSET + 4 float	Step frequency value Unit: [Hz]
Spectrum data.Frequency data	OFFSET + 64 int16 *BEAMS *BINS	Frequency spectrum amplitude data. There is room for 16 floating points for a spectrum header before the frequency data. Unit: [dB]

Position and size variables:

Name	Description
BEAMS	Matrix first dimension is number of beams. Eg: [[f_start, .., f_{start+step*(bins-1)}]_{beam1} [f_start, .., f_{start+step*(bins-1)}]_{beam2} .. [f_start, .., f_{start+step*(bins-1)}]_{beams}]
BINS	Per beam, frequencies are given as an array of length as number of bins. First element is the start frequency and frequencies increment by step frequency per element of the array. Eg:

	[[f_start, f_{start+step}, f_{start+step*2}, .., f_{start+step*(bins-1)}]_{beam1}, .., ..]
16+BEAMSxBINSx2	If configuration.hasSpectrumData is false, spectrum data is length 0. RAW: !this.configuration.hasSpectrumData ? 0 : this.beamsAndBins.numberOfBeams*this.beamsAndBins.numberOfBins*2 + 16*4
OFFSET	Number of bytes from start of record to start of data.

Object reference: Configuration

Record configuration bit mask.

Field	Position Size	Description
Has spectrum data	15 bit	Amplitude spectrum data included.

8.6 DF21 BottomTrack / DF22 WaterTrack

ID: 0x1b, 0x1d

Data definitions for parsing bottom track (DF21) and water track (DF22).

Field	Position Size	Description
Version	0 uint8	Version number of the Data Record Definition. Should be 3
Offset OfData	1 uint8	Number of bytes from start of the record to start of the actual data. Unit: [# bytes]
Serial Number	2 uint32	Instrument serial number from factory.
Date Time	6 uint8 *8	The date and time of the data record. Year: Is given as years from 1900. Month: January is 0. Milli seconds: Are given by the hundreds. That is as desi seconds.
Micro Seconds	12 uint16	Remaining micro seconds (Date object has milliseconds resolution)
Num Beams	14 uint16	Number of beams
Error status	16 32 bits	Error bit mask Object reference given in table below

Status	20 32 bits	Status bit mask Object reference given in table below
Sound Speed	24 float	Sound Speed Unit: [m/s]
Temperature	28 float	Temperature Unit: [°C]
Pressure	32 float	Pressure Unit: [Bar]
Velocity beam 1	36 float	Velocity beam 1 invalid estimates set to -32.768 Unit: [m/s]
Velocity beam 2	40 float	Velocity beam 2 invalid estimates set to -32.768 Unit: [m/s]
Velocity beam 3	44 float	Velocity beam 3 invalid estimates set to -32.768 Unit: [m/s]
Velocity beam 4	48 float	Velocity beam 4 invalid estimates set to -32.768 Unit: [m/s]
Distance beam 1	52 float	Distance beam 1 Vertical Distance invalid estimates set to 0.0 Unit: [m]
Distance beam 2	56 float	Distance beam 2 Vertical Distance invalid estimates set to 0.0 Unit: [m]
Distance beam 3	60 float	Distance beam 3 Vertical Distance invalid estimates set to 0.0 Unit: [m]
Distance beam 4	64 float	Distance beam 4 Vertical Distance invalid estimates set to 0.0 Unit: [m]
Uncertainty beam 1	68 float	Estimated velocity uncertainty reported as one standard deviation (Figure of merit - FOM) for beam 1. invalid estimates set to 10.0 Unit: [m/s]

Uncertainty beam 2	72 float	Estimated velocity uncertainty reported as one standard deviation (Figure of merit - FOM) for beam 2. invalid estimates set to 10.0 Unit: [m/s]
Uncertainty beam 3	76 float	Estimated velocity uncertainty reported as one standard deviation (Figure of merit - FOM) for beam 3. invalid estimates set to 10.0 Unit: [m/s]
Uncertainty beam 4	80 float	Estimated velocity uncertainty reported as one standard deviation (Figure of merit - FOM) for beam 4. invalid estimates set to 10.0 Unit: [m/s]
Dt1 beam 1	84 float	DT1 beam 1 Unit: [s]
Dt1 beam 2	88 float	DT1 beam 2 Unit: [s]
Dt1 beam 3	92 float	DT1 beam 3 Unit: [s]
Dt1 beam 4	96 float	DT1 beam 4 Unit: [s]
Dt2 beam 1	100 float	DT2 beam 1 Unit: [s]
Dt2 beam 2	104 float	DT2 beam 2 Unit: [s]
Dt2 beam 3	108 float	DT2 beam 3 Unit: [s]
Dt2 beam 4	112 float	DT2 beam 4 Unit: [s]
Time Velocity Estimate beam 1	116 float	Time Velocity Estimate beam 1 Duration of velocity estimate for each beam. Unit: [s]
Time Velocity Estimate beam 2	120 float	Time Velocity Estimate beam 2 Duration of velocity estimate for each beam. Unit: [s]
Time Velocity Estimate beam 3	124 float	Time Velocity Estimate beam 3 Duration of velocity estimate for each beam. Unit: [s]
Time Velocity Estimate beam 4	128 float	Time Velocity Estimate beam 4 Duration of velocity estimate for each beam.

		Unit: [s]
Velocity X	132 float	Velocity X Invalid estimates set to -32.768 Unit: [m/s]
Velocity Y	136 float	Velocity Y Invalid estimates set to -32.768 Unit: [m/s]
Velocity Z1	140 float	Velocity Z1 Invalid estimates set to -32.768 Unit: [m/s]
Velocity Z2	144 float	Velocity Z2 Invalid estimates set to -32.768 Unit: [m/s]
Uncertainty X	148 float	Estimated velocity uncertainty reported as one standard deviation (Figure of merit - FOM) in X dimension. Invalid estimates set to 10.0 Unit: [m/s]
Uncertainty Y	152 float	Estimated velocity uncertainty reported as one standard deviation (Figure of merit - FOM) in Y dimension. Invalid estimates set to 10.0 Unit: [m/s]
Uncertainty Z1	156 float	Estimated velocity uncertainty reported as one standard deviation (Figure of merit - FOM) in Z1 dimension. Invalid estimates set to 10.0 Unit: [m/s]
Uncertainty Z2	160 float	Estimated velocity uncertainty reported as one standard deviation (Figure of merit - FOM) in Z2 dimension. Invalid estimates set to 10.0 Unit: [m/s]
Dt1 X	164 float	DT1 X Unit: [s]
Dt1 Y	168 float	DT1 Y Unit: [s]
Dt1 Z1	172 float	DT1 Z1 Unit: [s]
Dt1 Z2	176 float	DT1 Z2 Unit: [s]
Dt2 X	180	DT2 X

	float	Unit: [s]
Dt2 Y	184 float	DT2 Y Unit: [s]
Dt2 Z1	188 float	DT2 Z1 Unit: [s]
Dt2 Z2	192 float	DT2 Z2 Unit: [s]
Time Velocity Estimate beam X	196 float	Time Velocity Estimate beam X Duration of velocity estimate for each beam. Unit: [s]
Time Velocity Estimate beam Y	200 float	Time Velocity Estimate beam Y Duration of velocity estimate for each beam. Unit: [s]
Time Velocity Estimate beam Z1	204 float	Time Velocity Estimate beam Z1 Duration of velocity estimate for each beam. Unit: [s]
Time Velocity Estimate beam Z2	208 float	Time Velocity Estimate beam Z2 Duration of velocity estimate for each beam. Unit: [s]

Object reference: Error status

Error bit mask

Field	Position Size	Description
Data retrieval FIFO error	0 bit	Data retrieval FIFO error
Data retrieval overflow	1 bit	Data retrieval overflow
Data retrieval underrun	2 bit	Data retrieval Underrun
Data retrieval samples missing	3 bit	Data retrieval samples missing
Measurement not finished	4 bit	The Measurement and data storage/transmit didn't finish before next measurement started.
Sensor read failure	5 bit	Sensor read failure
Tag error beam 1 (In-phase)	8 bit	Tag error beam 1 (In-phase)

Tag error beam 1 (Quadrature-phase)	9 bit	Tag error beam 1 (Quadrature-phase)
Tag error beam 2 (In-phase)	10 bit	Tag error beam 2 (In-phase)
Tag error beam 2 (Quadrature-phase)	11 bit	Tag error beam 2 (Quadrature-phase)
Tag error beam 3 (In-phase)	12 bit	Tag error beam 3 (In-phase)
Tag error beam 3 (Quadrature-phase)	13 bit	Tag error beam 3 (Quadrature-phase)
Tag error beam 4 (In-Phase)	14 bit	Tag error beam 4 (In-phase)
Tag error beam 4 (Quadrature-phase)	15 bit	Tag Error Beam 4 (Quadrature-phase)

Object reference: Status

Status bit mask

Field	**Position Size**	**Description**
Beam1 Velocity Valid	0 bit	Beam 1 Velocity Valid
Beam2 Velocity Valid	1 bit	Beam 2 Velocity Valid
Beam3 Velocity Valid	2 bit	Beam 3 Velocity Valid
Beam4 Velocity Valid	3 bit	Beam 4 Velocity Valid
Beam1 Distance Valid	4 bit	Beam 1 Distance Valid
Beam2 Distance Valid	5 bit	Beam 2 Distance Valid
Beam3 Distance Valid	6 bit	Beam 3 Distance Valid
Beam4 Distance Valid	7 bit	Beam 4 Distance Valid

Beam1 uncertainty Valid	8 bit	Beam 1 Figure of Merit Valid
Beam2 uncertainty Valid	9 bit	Beam 2 Figure of Merit Valid
Beam3 uncertainty Valid	12 bit	Beam 3 Figure of Merit Valid
Beam4 uncertainty Valid	11 bit	Beam 4 Figure of Merit Valid
Velocity X Valid	12 bit	X Velocity Valid
Velocity Y Valid	13 bit	Y Velocity Valid
Velocity Z1 Valid	14 bit	Z1 Velocity Valid
Velocity Z2 Valid	15 bit	Z2 Velocity Valid
Uncertainty X Valid	16 bit	X Figure of Merit Valid
Uncertainty Y Valid	17 bit	Y Figure of Merit Valid
Uncertainty Z1 Valid	18 bit	Z1 Figure of Merit Valid
Uncertainty Z2 Valid	19 bit	Z2 Figure of Merit Valid
Less 3% Capacity	20 bit	Less than 3% processing capacity left
Less 6% Capacity	21 bit	Less than 6% processing capacity left
Less 12% Capacity	22 bit	Less than 12% processing capacity left
Previous Wakeup State	28-31 4 bits	Previous wakeup State 0010 = break 0011 = RTC alarm 0000 = bad power 0001 = power applied

8.7 DF30 AltimeterData

ID: 0x21

Data format DF30 is used for DVL Altimeter measurements.

Field	Position Size	Description
Version	0 uint8	Version number of this Data Record Definition. Should be 1.
Offset of data	1 uint8	Number of bytes from start of record to start of non-common data fields. Unit: [# bytes]
Serial Number	2 uint32	Instrument serial number from factory.
Year	6 uint8	Number of years since 1900.
Month	7 uint8	Month number counting from 0 which is January.
Day	8 uint8	Day of the month
Hour	9 uint8	24 hour of the day.
Minutes	10 uint8	Minutes
Seconds	11 uint8	Seconds
Hundred micro seconds	12 uint16	Hundred micro seconds from last whole second. Unit: [µs]
Num Beams	14 uint16	Number of beams
Error status	16 32 bits	Error bit mask Object reference given in table below
Status	20 32 bits	Status bit mask Object reference given in table below
Sound Speed	24 float	Sound Speed Unit: [m/s]
Temperature	28 float	Temperature Unit: [°C]
Pressure	32 float	Pressure Unit: [Bar]
Altimeter Distance	36 float	Distance to bottom Unit: [m]
Altimeter quality	40 uint16	Quality parameter
Unused	42	34 bytes for future use

	34 bytes	

Object reference: Error status

Error bit mask

Field	Position Size	Description
Data retrieval FIFO error	0 bit	Data retrieval FIFO error
Data retrieval overflow	1 bit	Data retrieval overflow
Data retrieval underrun	2 bit	Data retrieval Underrun
Data retrieval samples missing	3 bit	Data retrieval samples missing
Measurement not finished	4 bit	The Measurement and data storage/transmit didn't finish before next measurement started.
Sensor read failure	5 bit	Sensor read failure
Tag error beam 1 (In-phase)	8 bit	Tag error beam 1 (In-phase)
Tag error beam 1 (Quadrature-phase)	9 bit	Tag error beam 1 (Quadrature-phase)
Tag error beam 2 (In-phase)	10 bit	Tag error beam 2 (In-phase)
Tag error beam 2 (Quadrature-phase)	11 bit	Tag error beam 2 (Quadrature-phase)
Tag error beam 3 (In-phase)	12 bit	Tag error beam 3 (In-phase)
Tag error beam 3 (Quadrature-phase)	13 bit	Tag error beam 3 (Quadrature-phase)
Tag error beam 4 (In-Phase)	14 bit	Tag error beam 4 (In-phase)
Tag error beam 4 (Quadrature-phase)	15 bit	Tag Error Beam 4 (Quadrature-phase)

Object reference: Status

Status bit mask

Field	Position Size	Description
Less 3% Capacity	20 bit	Less than 3% processing capacity left
Less 6% Capacity	21 bit	Less than 6% processing capacity left
Less 12% Capacity	22 bit	Less than 12% processing capacity left
Previous Wakeup State	28-31 4 bits	Previous wakeup State 0010 = break 0011 = RTC alarm 0000 = bad power 0001 = power applied

9 ASCII Data Formats

9.1 DVL Bottom Track ASCII formats

Invalid estimates of Velocity are set to set to -32.768.
Invalid estimates of Range are set to 0.0.
Invalid estimates of FOM are set to 10.0.

9.1.1 DF350/DF351 – NMEA $PNORBT1/$PNORBT0

The NMEA formats 350/351 have the following fields:

Field/TAG	Description	Data format	Example
BEAM	Beam number	n	BEAM=2
DATE	Date	DDMMYY	DATE=110916
TIME	Time	hhmmss.ssss	TIME=112034.0346
DT1	Time from the trigger to the centre of the bottom echo. [ms]	s.sss	DT1=55.717
DT2	Time from the start of the NMEA output message to the centre of the bottom echo. [ms]	s.sss	DT2=-157.912
BV	Beam Velocity	f.fffff	BV=0.15630
FM	Figure of Merit [m/s]	f.fffff	FM=0.00146
DIST	Vertical Distance to bottom. [m]	f.ff	DIST=26.92

Field/TAG	Description	Data format	Example
STAT	Status (see Table 7)	0xHHHHHHHH	STAT=0x000FFFFF

There is one text line output per beam so a four beam system will output four lines for each bottom track sample.

The DT1 parameter is the time from the trigger to the centre of the bottom echo that estimates the bottom track velocity. The DT2 parameter is the time from the start of the NMEA output message to the centre of the bottom echo. This will thus be a negative value.

Example (DF=350):
```
$PNORBT1,BEAM=1,DATE=110916,TIME=112034.0346,DT1=55.717,DT2=-
157.789,BV=0.15633,FM=0.00066,DIST=26.92,STAT=0x000FFFFF*2A
$PNORBT1,BEAM=2,DATE=110916,TIME=112034.0346,DT1=55.717,DT2=-
157.912,BV=0.15630,FM=0.00146,DIST=26.92,STAT=0x000FFFFF*25
$PNORBT1,BEAM=3,DATE=110916,TIME=112034.0346,DT1=55.717,DT2=-
158.034,BV=-0.14928,FM=0.00165,DIST=26.92,STAT=0x000FFFFF*0D
$PNORBT1,BEAM=4,DATE=110916,TIME=112034.0346,DT1=54.892,DT2=-
158.981,BV=-0.14925,FM=0.00359,DIST=26.92,STAT=0x000FFFFF*0E
```

9.1.2 DF354/DF355 – NMEA $PNORBT3/$PNORBT4

The NMEA formats 354/355 have the following fields:

Field/TAG	Description	Data format	Example
DT1	Time from the trigger to the centre of the bottom echo. [ms]	s.sss	DT1=1.234
DT2	Time from the start of the NMEA output message to the centre of the bottom echo. [ms]	s.sss	DT2=-1.234
SP	Speed over ground [m/s]	f.fff	SP=1.234
DIR	Direction [deg]	f.f	DIR=23.4
FOM	Figure of Merit [m/s]	f.fffff	FOM=12.34567
D	Vertical Distance to bottom. [m]	ff.f	D=12.3

DIR is the angle calculated by atan2(velY,velX) and is referenced to the X-axis.

DF354 outputs the tags. DF355 minimizes the number of character to be transmitted by discarding the tags in the outputted sentence.

Example (DF=354):
```
$PNORBT3,DT1=1.234,DT2=-
1.234,SP=1.234,DIR=23.4,FOM=12.34567,D=12.3*65
```

Example (DF=355):

```
$PNORBT4,1.234,-1.234,1.234,23.4,12.34567,12.3*09
```

9.1.3 DF356/DF357 – NMEA $PNORBT6/$PNORBT7

The NMEA formats 356/357 have the following fields:

Field/TAG	Description	Data format	Example
TIME	Ping time [POSIX time]	s.ssss	TIME=1452244916.7508
DT1	Time from the trigger to the center of the bottom echo. [ms]	s.sss	DT1=1.234
DT2	Time from the start of the NMEA output message to the center of the bottom echo. [ms]	s.sss	DT2=-1.234
VX	Speed in X direction [m/s]	f.ffff	VX=0.1234
VY	Speed in Y direction [m/s]	f.ffff	VY=0.1234
VZ	Speed in Z direction [m/s]	f.ffff	VZ=0.1234
FOM	Figure of Merit [m/s]	f.fffff	FOM=12.34567
D1	Beam 1: Vertical Distance to bottom. [m]	f.ff	D1=23.45
D2	Beam 2: Vertical Distance to bottom. [m]	f.ff	D2=23.45
D3	Beam 3: Vertical Distance to bottom. [m]	f.ff	D3=23.45
D4	Beam 4: Vertical Distance to bottom. [m]	f.ff	D4=23.45

DF356 outputs the tags. DF357 minimizes the number of character to be transmitted by discarding the tags in the outputted sentence.

Example (DF=356):
```
$PNORBT6,TIME=1452244916.7508,DT1=1.234,DT2=-
1.234,VX=0.1234,VY=0.1234,VZ=0.1234,FOM=12.34567,D1=23.45,D2=23.45,D3
=23.45,D4=23.45*6A
```

Example (DF=357):
```
$PNORBT7,1452244916.7508,1.234,-
1.234,0.1234,0.1234,0.1234,12.34,23.45,23.45,23.45,23.45*39
```

9.1.4 DF358/DF359 – NMEA $PNORBT8/$PNORBT9

The NMEA formats 358/359 have the following fields:

Field/TAG	Description	Data format	Example
TIME	Ping time [POSIX time]	s.ssss	TIME=1452244916.7508
DT1	Time from the trigger to the centre of the bottom echo. [ms]	s.sss	DT1=1.234
DT2	Time from the start of the NMEA output message to the centre of the bottom echo. [ms]	s.sss	DT2=-1.234
VX	Speed in X direction [m/s]	f.ffff	VX=0.1234
VY	Speed in Y direction [m/s]	f.ffff	VY=0.1234
VZ	Speed in Z direction [m/s]	f.ffff	VZ=0.1234
FOM	Figure of Merit [m/s]	f.ff	FOM=12.34
D1	Beam 1: Vertical Distance to bottom. [m]	f.ff	D1=23.45
D2	Beam 2: Vertical Distance to bottom. [m]	f.ff	D2=23.45
D3	Beam 3: Vertical Distance to bottom. [m]	f.ff	D3=23.45
D4	Beam 4: Vertical Distance to bottom. [m]	f.ff	D4=23.45
BATT	Battery Voltage [V]	f.f	BATT=23.4
SS	Speed of sound in Water [m/s]	f.f	SS=1567.8
PRESS	Pressure [dBar]	f.f	PRESS=1.2
TEMP	Water temperature [deg C]	f.f	TEMP=12.3
STAT	Status (see Table 7)	0xHHHHHHHH	STAT=0x000FFFFF

DF358 outputs the tags. DF359 minimizes the number of character to be transmitted by discarding the tags in the outputted sentence.

Example (DF=358):

```
$PNORBT8,TIME=1452244916.7508,DT1=1.234,DT2=-
1.234,VX=0.1234,VY=0.1234,VZ=0.1234,FOM=12.34,D1=23.45,D2=23.45,D3=23
.45,D4=23.45,BATT=23.4,SS=1567.8,PRESS=1.2,TEMP=12.3,STAT=0x000FFFFF*
1E
```

Example (DF=359):
```
$PNORBT9,1452244916.7508,1.234,-
1.234,0.1234,0.1234,0.1234,12.34,23.45,23.45,23.45,23.45,23.4,1567.8,
1.2,12.3,0x000FFFFF*1E
```

9.2 DVL Water Track ASCII formats

Invalid estimates of Velocity are set to set to -32.768.
Invalid estimates of Range are set to 0.0.
Invalid estimates of FOM are set to 10.0.

9.2.1 DF404/DF405 – NMEA $PNORWT3/$PNORWT4

The NMEA formats 404/405 have the following fields:

Field/TAG	Description	Data format	Example
DT1	Time from the trigger to the centre of the water track cell. [ms]	s.sss	DT1=1.234
DT2	Time from the start of the NMEA output message to the centre of the water track cell. [ms]	s.sss	DT2=-1.234
SP	Speed [m/s]	f.fff	SP=1.234
DIR	Direction [deg]	f.f	DIR=23.4
FOM	Figure of Merit [m/s]	f.ff	FOM=12.34
D	Vertical Distance to water track cell. [m]	ff.f	D=12.3

DIR is the angle calculated by atan2(velY,velX) and is referenced to the X-axis.

DF404 outputs the tags. DF405 minimizes the number of character to be transmitted by discarding the tags in the outputted sentence.

Example (DF=404):

```
$PNORWT3,DT1=1.2345,DT2=-1.2345,SP=1.234,DIR=23.4,FOM=12.34,D=12.3*44
```

Example (DF=405):

```
$PNORWT4,1.2345,-1.2345,1.234,23.4,12.34,12.3*1C
```

9.2.2 DF406/DF407 – NMEA $PNORWT6/$PNORWT7

The NMEA formats 406/407 have the following fields:

Field/TAG	Description	Data format	Example
TIME	Ping time [POSIX time]	s.ssss	TIME=1452244916.7508
DT1	Time from the trigger to the centre of water track cell. [ms]	s.sss	DT1=1.234
DT2	Time from the start of the NMEA output message to the centre of the water track cell. [ms]	s.sss	DT2=-1.234
VX	Speed in X direction [m/s]	f.ffff	VX=0.1234
VY	Speed in Y direction [m/s]	f.ffff	VY=0.1234
VZ	Speed in Z direction [m/s]	f.ffff	VZ=0.1234
FOM	Figure of Merit [m/s]	f.ff	FOM=12.34
D1	Beam 1: Vertical Distance to water track cell. [m]	f.ff	D1=23.45
D2	Beam 2: Vertical Distance to water track cell. [m]	f.ff	D2=23.45
D3	Beam 3: Vertical Distance to water track cell. [m]	f.ff	D3=23.45
D4	Beam 4: Vertical Distance to water track cell. [m]	f.ff	D4=23.45

DF406 outputs the tags. DF407 minimizes the number of character to be transmitted by discarding the tags in the outputted sentence.

Example (DF=406):
```
$PNORWT6,TIME=1452244916.7508,DT1=1.234,DT2=-
1.234,VX=0.1234,VY=0.1234,VZ=0.1234,FOM=12.34,D1=23.45,D2=23.45,D3=23
.45,D4=23.45*4B
```

Example (DF407):
```
$PNORWT7,1452244916.7508,1.234,-
1.234,0.1234,0.1234,0.1234,12.34,23.45,23.45,23.45,23.45*2C
```

9.2.3 DF408/DF409 – NMEA $PNORWT8/$PNORWT9

The NMEA formats 408/409 have the following fields:

Field/TAG	Description	Data format	Example
TIME	Ping time [POSIX GMT time]	s.ssss	TIME=1452244916.7508
DT1	Time from the trigger to the centre of the water track cell. [ms]	s.sss	DT1=1.234
DT2	Time from the start of the NMEA output message to the centre of the water track cell. [ms]	s.sss	DT2=-1.234
VX	Speed in X direction [m/s]	f.ffff	VX=0.1234
VY	Speed in Y direction [m/s]	f.ffff	VY=0.1234
VZ	Speed in Z direction [m/s]	f.ffff	VZ=0.1234
FOM	Figure of Merit [m/s]	f.ff	FOM=12.34
D1	Beam 1: Vertical Distance to water track cell. [m]	f.ff	D1=23.45
D2	Beam 2: Vertical Distance to water track cell. [m]	f.ff	D2=23.45
D3	Beam 3: Vertical Distance to water track cell. [m]	f.ff	D3=23.45
D4	Beam 4: Vertical Distance to water track cell. [m]	f.ff	D4=23.45
BATT	Battery Voltage [V]	f.f	BATT=23.4
SS	Speed of sound in Water [m/s]	f.f	SS=1567.8
PRESS	Pressure [dBar]	f.f	PRESS=1.2
TEMP	Water temperature [deg C]	f.f	TEMP=12.3
STAT	Status (see Table 7)	0xHHHHHHHH	STAT=0x000FFFFF

DF408 outputs the tags. DF409 minimizes the number of character to be transmitted by discarding the tags in the outputted sentence.

Example (DF=408):

```
$PNORWT8,TIME=1452244916.7508,DT1=1.234,DT2=-
1.234,VX=0.1234,VY=0.1234,VZ=0.1234,FOM=12.34,D1=23.45,D2=23.45,D3=23
.45,D4=23.45,BATT=23.4,SS=1567.8,PRESS=1.2,TEMP=12.3,STAT=0x000FFFFF*
0B
```

Example (DF=409):

```
$PNORWT9,1452244916.7508,1.234,-
1.234,0.1234,0.1234,0.1234,12.34,23.45,23.45,23.45,23.45,23.4,1567.8,
1.2,12.3,0x000FFFFF*0B
```

9.3 DVL Current Profile ASCII formats

9.3.1 DF100 - Prolog NMEA Format

See Prolog Data Format Description.

9.3.2 DF101/DF102 - NMEA Format 1 and 2

Information Data:
Identifier:
PNORI1 for DF = 101
PNORI2 for DF = 102

Field	Description	TAG	Data format	Example
Instrument type	4 = Signature75	IT	N	IT=4
Head ID		SN	N	SN=123456
Number of Beams		NB	N	NB=3
Number of Cells		NC	N	NC=30
Blanking Distance	[m]	BD	dd.dd	BD=1.00
Cell Size	[m]	CS	dd.dd	CS=5.00
Coordinate System		CY	ENU,BEAM,XYZ	CY=BEAM

Example (DF=101):
```
$PNORI1,4,123456,3,30,1.00,5.00,BEAM*5B
```

Example (DF=102):
```
$PNORI2,IT=4,SN=123456,NB=3,NC=30,BD=1.00,CS=5.00,CY=BEAM*68
```

Sensors Data:
Identifier:
PNORS1 for DF = 101
PNORS2 for DF = 102

Field	Description	TAG	Data format	Example
Date		DATE	MMDDYY	DATE=083013
Time		TIME	hhmmss	TIME=132455
Error Code		EC	N	EC=0
Status Code		SC	hhhhhhhh	SC=34000034
Battery Voltage	[V]	BV	dd.d	BV=23.9
Sound Speed	[m/s]	SS	dddd.d	SS=1500.0
Heading	[deg] [unused]	H	ddd.d	H=123.4

Field	Description	TAG	Data format	Example
Heading Std.Dev	[deg] [unused]	HSD	dd.dd	HSD=0.02
Pitch	[deg] [unused]	PI	dd.d	PI=45.6
Pitch Std.Dev	[deg] [unused]	PISD	dd.dd	PISD=0.02
Roll	[deg] [unused]	R	dd.d	R=23.4
Roll Std.Dev.	[deg] [unused]	RSD	dd.dd	RSD=0.02
Pressure	[dBar]	P	ddd.ddd	P=123.456
Pressure Std.Dev	[dBar]	PSD	dd.dd	PSD=0.02
Temperature	[deg C]	T	dd.dd	T=24.56

Example (DF=101):

```
$PNORS1,083013,132455,0,34000034,23.9,1500.0,123.4,0.02,45.6,0.02,R=2
3.4,0.02,123.456,0.02,24.56*39
```

Example (DF=102):

```
$PNORS2,DATE=083013,TIME=132455,EC=0,SC=34000034,BV=23.9,SS=1500.0,H=
123.4,HSD=0.02,PI=45.6,PISD=0.02,R=23.4,RSD=0.02,P=123.456,PSD=0.02,T
=24.56*3F
```

Current Data:
Identifier:
PNORC1 for DF = 101
PNORC2 for DF = 102

The current data is output for each measurement cell.

Field	Description	TAG	Data format	Example
Date	Date	DAT E	MMDDYY	DATE=083013
Time	Time	TIME	hhmmss	TIME=132455
Cell Number	#	CN	dd	CN=3
Cell Position	[m]	CP	dd.d	CP=11.0
Velocity East	[m/s] Only if CY=ENU	VE	dd.ddd	VE=0.332
Velocity North	[m/s] Only if CY=ENU	VN	dd.ddd	VN=0.332
Velocity Up	[m/s] Only if CY=ENU and #beams >= 3	VU	dd.ddd	VU=0.332
Velocity Up2	[m/s]	VU2	dd.ddd	VU2=0.332

Field	Description	TAG	Data format	Example
	Only if CY=ENU and #beams = 4			
Velocity X	[m/s] Only if CY=XYZ	VX	dd.ddd	VX=0.332
Velocity Y	[m/s] Only if CY=XYZ	VY	dd.ddd	VY=0.332
Velocity Z	[m/s] Only if CY=XYZ and #beams >= 3	VZ	dd.ddd	VZ=0.332
Velocity Z2	[m/s] Only if CY=XYZ and #beams = 4	VZ2	dd.ddd	VZ2=0.332
Velocity Beam 1	[m/s] Only if CY=BEAM	V1	dd.ddd	V1=0.332
Velocity Beam 2	[m/s] Only if CY=BEAM and #beams >=2	V2	dd.ddd	V2=0.332
Velocity Beam 3	[m/s] Only if CY=BEAM and #beams >=3	V3	dd.ddd	V3=-0.332
Velocity Beam 4	[m/s] Only if CY=BEAM and #beams =4	V4	dd.ddd	V4=-0.332
Amplitude Beam 1	[dB]	A1	ddd.d	A1=78.9
Amplitude Beam 2	[dB] Only if #beams >=2	A2	ddd.d	A2=78.9
Amplitude Beam 3	[dB] Only if #beams >=3	A3	ddd.d	A3=78.9
Amplitude Beam 4	[dB] Only if #beams =4	A4	ddd.d	A4=78.9
Correlation Beam 1	[%]	C1	dd	C1=78
Correlation Beam 2	[%] Only if #beams >=2	C2	dd	C2=78
Correlation Beam 3	[%] Only if #beams >=3	C3	dd	C3=78
Correlation Beam 4	[%] Only if #beams =4	C4	dd	C4=78

Example (DF=101 (ENU, 3 beams):

```
$PNORC1,083013,132455,3,11.0,0.332,0.332,0.332,78.9,78.9,78.9,78,78,7
8*46
```

Example (DF=102 (ENU, 3 beams):

`$PNORC2,DATE=083013,TIME=132455,CN=3,CP=11.0,VE=0.332,VN=0.332,VU=0.3
32,A1=78.9,A2=78.9,A3=78.9,C1=78,C2=78,C3=78*6D`

Example (DF=102 (BEAM, 4 beams):
`$PNORC2,DATE=083013,TIME=132455,CN=3,CP=11.0,V1=0.332,V2=0.332,V3=-
0.332,V4=-
0.332,A1=78.9,A2=78.9,A3=78.9,A4=78.9,C1=78,C2=78,C3=78,C4=78*49`

9.3.3 DF103/DF104

Header Data:
Identifier:
PNORH3 for DF = 103
PNORH4 for DF = 104

Field	Description	TAG	Data format	Example
DATE	Date	DATE	YYMMDD	DATE=161109
TIME	Time	TIME	HHMMSS	TIME=143459
EC	Error Code	EC	D	EC=0
SC	Status Code (hex)	SC	HHHHHHHH	SC=204C0002

Example (DF=103):
`$PNORH3,DATE=161109,TIME=143459,EC=0,SC=204C0002*28`

Example (DF=104):
`$PNORH4,161109,143459,0,204C0002*38`

Sensors Data:
Identifier:
PNORS3 for DF = 103
PNORS4 for DF = 104

Field	Description	TAG	Data format	Example
Battery Voltage	[Volts]	BV	ff.f	BV=23.6
Speed of Sound	[m/s]	SS	ffff.f	SS=1530.2
Heading	[deg] [unused]	H	f.f	H=56.7
Pitch	[deg] [unused]	PI	f.f	PI=3.4
Roll	[deg] [unused]	R	f.f	R=-3.4
Pressure	[dBar]	P	f.fff	P=6.789
Temperature	[deg C]	T	f.ff	T=23.30

Example (DF=103):
`$PNORS3,BV=23.6,SS=1530.2,H=0.0,PI=0.0,R=0.0,P=0.000,T=23.30*64`

Example (DF=104):
```
$PNORS4,23.6,1530.2,0.0,0.0,0.0,0.000,23.30*66
```

Current Data:
Identifier:
PNORC3 for DF = 103
PNORC4 for DF = 104

The current data is output for each measurement cell.

Field	Description	TAG	Data format	Example
Cell Position	[meters]	CP	f.f	CP=1.5
Speed	[m/s]	SP	f.fff	SP=1.395
Direction	[deg]	DIR	f.f	DIR=227.1
Correlation	[%]	AC	u	AC=32
Amplitude	[dB]	AA	u	AA=32

Direction is the angle calculated by atan2(velY,velX) and is referenced to the X-axis.

Example (DF=103):
```
$PNORC3,CP=1.5,SP=1.395,DIR=227.1,AC=32,AA=32*0D
$PNORC3,CP=2.5,SP=1.275,DIR=228.1,AC=35,AA=32*09
$PNORC3,CP=3.5,SP=1.256,DIR=240.9,AC=35,AA=32*0F
```

Example (DF=104):
```
$PNORC4,1.5,1.395,227.1,32,32*7A
```

9.4 DVL Altimeter ASCII Formats

9.4.1 DF200/DF201 - NMEA Format 200 and 201

Altimeter Data:

PNORA for DF = 200 and 201.
DF201 includes tags.

<table>
<tr><th>Field</th><th colspan="2">Description</th><th>TAG</th><th>Data format</th><th>Example</th></tr>
<tr><td>DATE</td><td colspan="2">Date</td><td>DATE</td><td>YYMMDD</td><td>161206</td></tr>
<tr><td>TIME</td><td colspan="2">Time</td><td>TIME</td><td>HHMMSS</td><td>094717</td></tr>
<tr><td>P</td><td colspan="2">Pressure [dBar]</td><td>P</td><td>f.fff</td><td>1.234</td></tr>
<tr><td>A</td><td colspan="2">Altimeter [m]</td><td>A</td><td>f.fff</td><td>49.401</td></tr>
<tr><td>Q</td><td colspan="2">Quality</td><td>Q</td><td>U</td><td>12345</td></tr>
<tr><td rowspan="6">ST</td><td colspan="2">Status, Hexadecimal</td><td rowspan="6">ST</td><td rowspan="6">HH</td><td rowspan="6">08 - one beam</td></tr>
<tr><td>Bit #</td><td>Description</td></tr>
<tr><td>0</td><td>Pitch or Roll > 5 deg</td></tr>
<tr><td>1</td><td>Pitch or Roll > 10 deg</td></tr>
<tr><td>2</td><td>NA</td></tr>
<tr><td>6-3</td><td># beams</td></tr>
</table>

Example (DF=200):
```
$PNORA,161206,094717,0.000,49.401,17081,08*43
```

Example (DF=201):
```
$PNORA,DATE=161206,TIME=094737,P=0.000,A=49.404,Q=14447,ST=08*0F
```

9.4.2 DF202 - NMEA Format 202

Standard “Depth below Transducer” NMEA sentence.

```
         1   2 3   4 5   6 7
         |   | |   | |   | |
$--DBT,x.x,f,x.x,M,x.x,F*hh<CR><LF>
```

Field Number:
1. Depth, feet
2. f = feet
3. Depth, meters
4. M = meters
5. Depth, Fathoms
6. F = Fathoms
7. Checksum

Example (DF=202):
`$SDDBT,162.01,f,49.38,M,27.00,F*31`

9.4.3 DF203 - NMEA Format 203

Standard “Depth below Surface” NMEA sentence. This is output when the fast pressure function is enabled.

```
         1   2 3   4 5   6 7
         |   | |   | |   | |
$--DBS,x.x,f,x.x,M,x.x,F*hh<CR><LF>
```

Field Number:
1. Depth, feet
2. f = feet
3. Depth, meters
4. M = meters
5. Depth, Fathoms
6. F = Fathoms
7. Checksum

Example (DF=203):
`$SDDBS,162.01,f,49.38,M,27.00,F*31`

10 Appendices

10.1 Checksum Definitions

The Checksum is defined as a 16-bits unsigned sum of the data (16 bits). The sum shall be initialized to the value of 0xB58C before the checksum is calculated.

C-code for Checksum calculations:

```
unsigned short calculateChecksum(unsigned short *pData, unsigned short size)
{
    unsigned short checksum = 0xB58C;
    unsigned short nbshorts = (size >> 1);
    int i;
    for (i = 0; i < nbshorts; i++)
    {
        checksum += *pData;
        size  -= 2;
        pData++;
    }
    if (size > 0)
    {
        checksum += ((unsigned short)(*pData)) << 8;
    }
    return checksum;
}
```