NORTEK AS

Nortek DVL Integrator’s Guide

Table of Contents

Document Revisions

Revision	Changes
14032016	• First official release
09052016	• Update of format

1 Introduction

The Nortek DVL is based on the AD2CP hardware platform. It operates in distinct modes. These modes will have several explicit commands in order 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

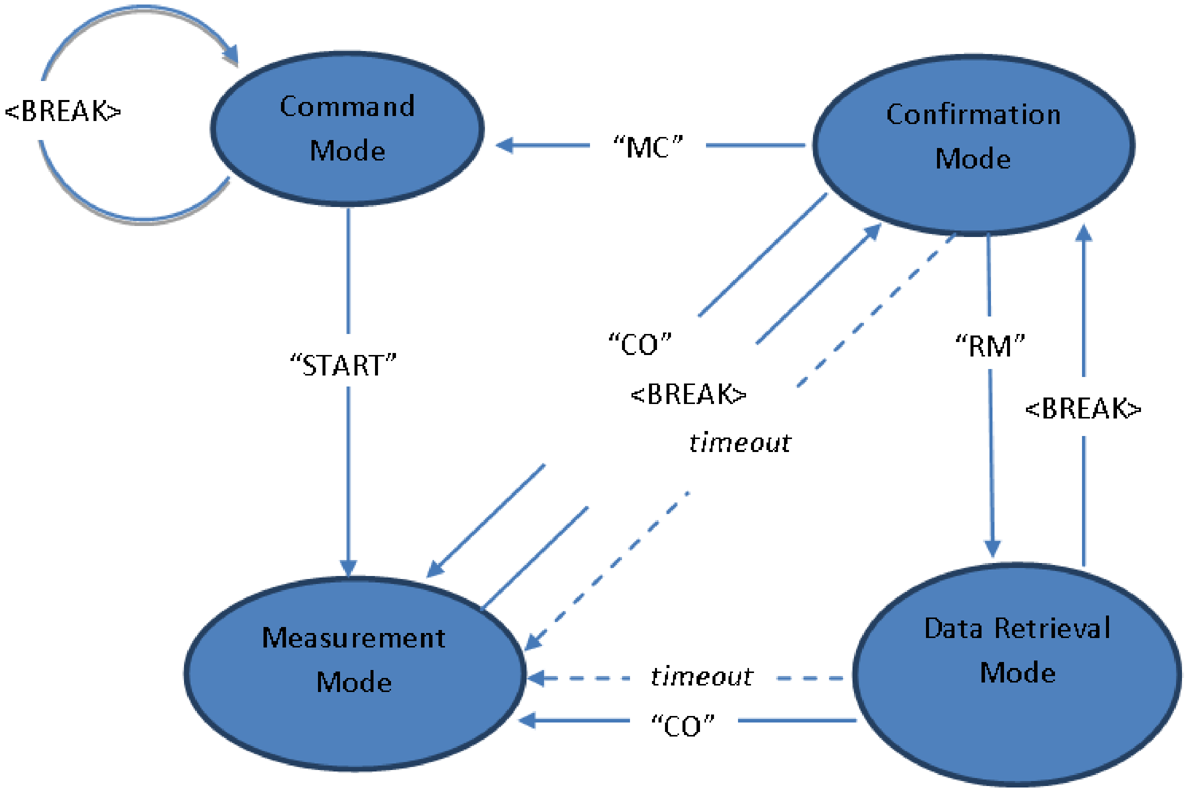


Figure 1 Instrument modes of operation

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).

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

@@@@@@ <delay 100 milliseconds> K1W%!Q <delay 300 milliseconds> K1W%!Q

The @@@@@@ are used to wake up the processor when it is in sleep mode since the instrument will only be able to monitor activity on the serial line when it sleeps. 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.

<BREAK> Timing Specification
The figure and the table below show the specified timing of the BREAK sequence.

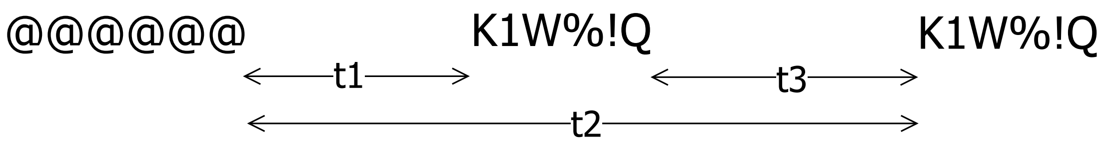


Figure 2 Break timing

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

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,ALL`) 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 GETxxxLIM 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,ALL
OK
```

1.1 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:

Command interface

The command interface makes it possible to communicate with a Signature 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)

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 data only channel (no input accepted)
- port 9004 is an ASCII data only channel (no input accepted).

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.

Commands available in measurement mode should be preceded by the command BBPWAKEUP. This ensures that the BBP is ready to process the command when it is received (see Checking instrument state over Ethernet). In measurement mode, another BBPWAKEUP must be sent when more than 2 seconds has elapsed since the previous command.

If uncertain of the active mode it is good practice to send BBPWAKEUP before sending GETSTATE or INQ.

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).

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.

FTP

The internal data recorder is accessed over Ethernet using a standard FTP (File Transfer Protocol) client. Together with the various telemetry options, the FTP data download serves as a simple way to download measured data at regular intervals if true real time operation is not required. Only the telemetry file can be deleted using FTP.

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.

For an example on how to configure the instrument to output a telemetry file and download the file to FTP, check out the next section.

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.

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 / errorprone connection, it is possible that the occasional datagram may be dropped. If every data record must be received, then TCP is recommended.

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.

PTP

Precision Time Protocol (IEEE-1588) is a standard used for distributing a high-resolution absolute time throughout an Ethernet network. The Signature series instrument 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 ~10 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 Signature series PTP implementation. Using PTP does not affect the choice of UDP or TCP for the transport of data.

List of Commands

Command	Description	Scope
START	Go into measurement mode	Command mode
MC	Go into command mode	Confirm mode
RM	Go into data retrieval mode	Confirm mode
CO	Continue in measurement mode.	Confirm mode, Data retrieval mode
INQ	Inquiry instrument state	All modes
SETINST / GETINST / GETINSTLIM	Set/Get Main Instrument Settings Get Instrument Setting Limits	Command mode
SETCLOCK/ GETCLOCK	Set/Get Real Time Clock	Command mode, Data retrieval mode
SETCLOCKSTR/ GETCLOCKSTR	Set/Get Real Time Clock using a string argument	Command mode, Data retrieval mode
SETCURPROF/ GETCURPROF/ GETCURPROFLIM	Set/Get Current Profiling Mode Settings Get Current Profiling Mode Limits	Command mode
SETUSER/GETUSER	Set/Get User Settings	Command mode
SETDEFAULT	Reload default settings.	Command mode
SAVE	Save current settings for next measurement.	Command mode
POWERDOWN	Go in power down.	Command mode
GETERRORNUM	Get last error number.	All modes
GETERRORSTR	Get last error string.	All modes
ERASE	Erase the recorder	Command mode
FORMAT	Format the recorder	Command mode
RECSTAT	Returns Recorder Statistics	Command mode, Data retrieval mode
SETBT/ GETBT/ GETBTLIM	Set Bottom Track configuration	Command mode

SETDVL/ GETDVL/ GETDVLLIM	Set DVL parameters.	Command mode
GETALL	Retrieves all configuration information from the instrument.	Command mode
READCFG	Read current configuration.	Command mode

2 Configuration examples

2.1 Example: 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:

```
SETDEFAULT,ALL
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,ALL
OK
START
OK
```

2.2 Example: 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:

```
SETDEFAULT,ALL
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,ALL
ERROR
GETERROR
261,"Invalid setting: Bottom track range invalid","SETBT,RANGE=([5.00;30.00])"
OK
SETBT,RANGE=30.0
OK
SAVE,ALL
OK
START
OK
```

3 Commands

3.1 SETINST/GETINST

Argument	Description	Default Value	Valid Range
BR	Baud Rate	9600	300 – 115000
RS	Serial protocol	232	232 or 422
LED	Enable/disable LED blink in head. When set to "ON24H" the LED will illuminate the first 24 hours of the measurement.	"ON"	"ON" "OFF" "ON24H"
ORIENT	Not used for the DVL.		
CMTOUT	Command mode timeout [seconds]. In Serial mode the system is powered down after this timeout after the last command. Not used in Ethernet mode	300	300 - 3600
DMTOUT	Data retrieval mode timeout [seconds].	60	60 - 3600
CFMTOUT	Confirmation retrieval mode timeout [seconds].	60	60 -300

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

3.2 SETCLOCK/GETCLOCK

Set or retrieve the Real Time Clock. Note that **all** parameters must be set when using the SETCLOCK command.

Argument	Description	Default Value	Valid Range
YEAR	Year		1970-
MONTH	Month		1-12
DAY	Day		1-31
HOUR	Hours (24 hour format)		0-23
MINUTE	Minutes		0-59
SECOND	Seconds		0-59

Scope:
Command mode and Data retrieval mode

3.3 SETCLOCKSTR/GETCLOCKSTR

Set or retrieve the Real Time Clock using a string. The format must be exactly as shown.

Argument	Description	Default Value	Valid Range
TIME	yyyy-mm-dd hh:mm:ss		

Scope:
Command mode and Data retrieval mode

3.4 SETDVL/GETDVL/GETDVLLIM

Argument	Description	Default Value	Valid Range
CP	Collect Current Profile every Nth ping, 0 to disable Current Profile	0	0 – 20
TRIG	Specifies trigger type	"INTSR"	"TTLRISE", "TTLFALL", "TTLEDGE", "SERIAL", "INTSR","RS485RISE", "RS485FALL","RS485EDGE"
SR	Internal sampling rate if enabled	1.0	1MHz: 1.0 to 8.0 Hz 500kHz: 1.0 to 8.0 Hz
FN	File name	""	30 characters a-z, A-Z, 0-9 . and _.
SV	Sound velocity (m/s)	0.0	1300.00-1700.00 0 will set sensor to use measured sound velocity
SA	Salinity (ppt)	35.0	0.0-50.0

3.5 SETCURPROF/GETCURPROF/GETCURPROFLIM

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.

Argument	Description	Default Value	Valid Range
NC	Number of cells		1-200
CS	Cell Size (m)		
BD	Blanking Distance (m)		
CY	Coordinate System		”XYZ“ “BEAM”
PL **VP**	Power Level [dB] Velocity Precision		-100 dB to switch off transmit -20.0 dB to 0.0 dB Not yet supported.
VR	Velocity range along beam [m/s]		1.0 – 5.0 m/s
DF	Data Format	3	3, 100, 101, 102, 150
NB	Number of beams	0	Select number of beams, 0 select all beams.
CH	Beam selection	0	Select beams, 0 selects beams in ascending order Example: 134 select the three beams 1, 3 and 4

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 3.20

3.6 SETBT/GETBT/GETBTLIM

Argument	Description	Default Value	Valid Range
RANGE	Bottom track range		
VR	Velocity range along beam		1.0 – 6.25 m/s
NB	Number of beams	0	Select number of beams, 0 select all beams.
CH	Beam selection	0	Select beams, 0 selects beams in ascending order Example: 134 select the three beams 1, 3 and 4
DF	Data Format	20	20 – Bottom Track Data Record version 1. 300 – NMEA without tags. 301 – NMEA with tags. 302 – NMEA without tags and Sensor Data. 303 – NMEA with tags and Sensor Data. 304 – NMEA PNORBT3 305 – NMEA PNORBT4 21 – Binary DVL data format.
PL	Power Level	0.0	0.0 - -20.0dB
WT	Measure Water Track velocity.	"OFF"	"OFF", ["ON")
WTDF	Water Track Data format		
BD	Blanking Distance (m)		
PLMODE	Power Level Mode	"MAX"	"MAX", "USER"

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.

3.7 SETUSER/GETUSER

Argument	Description	Default Value	Valid Range
POFF	Pressure offset [dbar] Set the offset value of the pressure sensor.	9.50	0.0-11.000
ROTXY	Alignment offset [deg]	0.0 deg	-180.0 – 180 deg
MA11	Alignment matrix	1.0	-1.0 – 1.0
MA12	Alignment matrix	0.0	
MA13	Alignment matrix	0.0	
MA21	Alignment matrix	0.0	

MA22	Alignment matrix	1.0	
MA23	Alignment matrix	0.0	
MA31	Alignment matrix	0.0	
MA32	Alignment matrix	0.0	
MA33	Alignment matrix	1.0	

The XYZ coordinate system of the system can be rotated by setting the Alignment Matrix:

$$\begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix} = \begin{bmatrix} MA_{11} & MA_{12} & MA_{13} \\ MA_{21} & MA_{22} & MA_{23} \\ MA_{31} & MA_{32} & MA_{33} \end{bmatrix} \cdot \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

All nine MAxy parameters must be set in the same command line.
An additional rotation of the XY coordinate system (about the z-axis) can be done using the ROTXY parameter. Note that the Alignment Matrix is multiplied in first so the rotation specified in ROTXY applies to the new coordinate system X'Y'Z'.

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

```
SETUSER,MA11=1.0,MA12=0.0,MA13=0.0,MA21=0.0,MA22=-1.0,MA23=0.0,MA31=0.0,MA32=0.0,MA33=-1.0
SETUSER,ROTXY=135.0
SAVE,USER
```

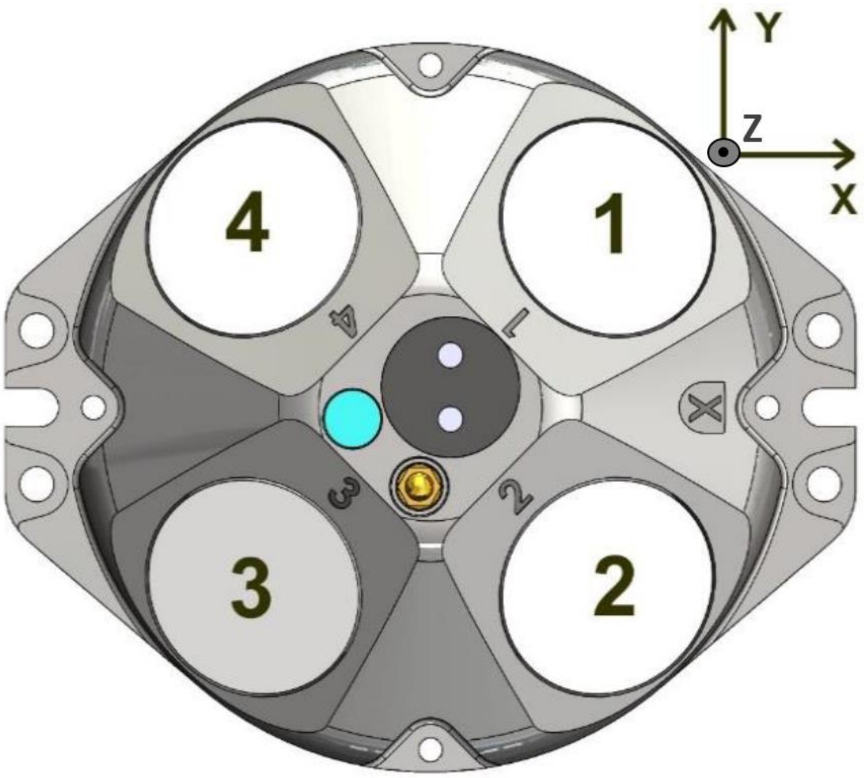


Figure 3 DVL beam numbering and Axes convention. Note that positive Z is pointing out of the page.

3.8 SETDEFAULT

Argument	Description	Default Value	Valid Range
ALL	Restore all settings below **except** USER and INST to default values.	No value	No value
CP	Restore CURPROF default.	No value	No value
INST	Restore INST default.	No value	No value
BT	Restore BT default.	No value	No value
USER	Restore USER default.	No value	No value
DVL	Restore DVL default.	No value	No value

3.9 SAVE

Argument	Description	Default Value	Valid Range
ALL	Save all settings.	No value.	No value.
CP	Save CURPROF parameters.	No value	No value
INST	Save INST parameters.	No value	No value
USER	Save USER parameters.	No value	No value
BT	Save BT parameters.	No value	No value
DVL	Save DVL setting parameters.	No value	No value

At least one argument must be specified for the SAVE command.

3.10 POWERDOWN

Power down the instrument to set it in sleep mode.

3.11 ERASE

Argument	Description	Default Value	Valid Range
CODE	Code should be 9999	No value.	9999

Erase all files on the recorder.

3.12 FORMAT

Argument	Description	Default Value	Valid Range
CODE	Code should be 9999	No value.	9999

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

3.13 INQ

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

Example (in command mode) :

```
08:43:31   INQ<CR><LF>
08:43:31   0002<CR><LF>
```

Example (in measurement mode) :

```
08:43:31   INQ<CR><LF>
08:43:31   0001<CR><LF>
```

Example (in confirmation mode) :

```
08:43:31   INQ<CR><LF>
08:43:31   0005<CR><LF>
```

Example (in data retrieval mode) :

```
08:43:31   INQ<CR><LF>
08:43:31   0004<CR><LF>
```

Example (in firmware upgrade mode) :

```
08:43:31   INQ<CR><LF>
08:43:31   0000<CR><LF>
```

3.14 GETERRORNUM

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

Argument	Description	Default Value	Valid Range
NUM	Integer error value	-	-

3.15 GETERRORSTR

GETERRORSTR retrieves a text description of the last error condition.

Argument	Description	Default Value	Valid Range
STR	Text description	-	-

3.16 GETERROR

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, see example below.

Argument	Description	Default Value	Valid Range
NUM	Integer error value		
STR	Text description		

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

3.17 GETALL

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.

Argument	Description	Default Value	Valid Range
FN	Write the output to this file		

Example :

```
GETALL
GETCLOCKSTR,"2016-03-11 08:31:13"
ID,"NortekDVL",200012
GETHW,4033,159,"H-0","D-0","C-4","D-0",19,1
BOARDSENSGET,3,4,1000,2.0000,0.0000
GETDVL,0,"INTSR",8.0,"",0.0,35.0
GETBT,30.00,5.00,4,0,21,-2.0,"OFF",22,0.01,"MAX"
GETXFBT,4,4,0.836579,0.836579,-0.836579,-0.836579,0.836579,-0.836579,-
0.836579,0.836579,0.551689,0.000000,0.551689,0.000000,0.000000,0.551689,0.000000,0.551689
GETUSER,9.50,0.000,1.000,0.000,0.000,0.000,1.000,0.000,0.000,0.000,1.000
GETINST,9600,232,"ON","AUTOZUPDOWN",300,60,60
GETCOMPASSCAL,199,-33,68,29794,-512,-215,637,29296,79,7,-136,32767
BEAMCFGLIST,1,25.00,45.00,1000,25,1,1,60.00
BEAMCFGLIST,2,25.00,-45.00,1000,25,1,2,60.00
BEAMCFGLIST,3,25.00,-135.00,1000,25,1,3,60.00
BEAMCFGLIST,4,25.00,135.00,1000,25,1,4,60.00
BEAMIMPLIST,1,1.00000e+02,0.00000e+00,0.00000e+00,0.00000e+00,0.00000e+00,0.00000e+00
BEAMIMPLIST,2,1.00000e+02,0.00000e+00,0.00000e+00,0.00000e+00,0.00000e+00,0.00000e+00
BEAMIMPLIST,3,1.00000e+02,0.00000e+00,0.00000e+00,0.00000e+00,0.00000e+00,0.00000e+00
BEAMIMPLIST,4,1.00000e+02,0.00000e+00,0.00000e+00,0.00000e+00,0.00000e+00,0.00000e+00
LISTLICENSE,"LR9UFYGLM518D","Averaging Mode",1
LISTLICENSE,"KKEXGJ74M518D","Burst Mode",2
LISTLICENSE,"YGD79C6Z38Y0B","Bottom Track",3
OK
```

3.18 RECSTAT

Return Recorder Statistics

Argument	Description	Description
SS	SectorSize	# of Bytes in a Sector.
CS	ClusterSize	# of Bytes in one Cluster
FC	Free Clusters	# of Bytes in Free Clusters
TC	Total Clusters	Total # of bytes in Clusters
VS	Volume Size	Volume Size in bytes

3.19 READCFG

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	Description
FN	Filename	

Example :

```
$PNOR,READCFG*7F
$PNOR,SETDEFAULT,ALL*4B
$PNOR,SETDVL,CP=8,TRIG="INTSR",SR=8.0,FN="",SV=0.0,SA=35.0*4C
$PNOR,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*4F
$PNOR,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"*58
$PNOR,SAVE,ALL*43
$PNOR,OK*2B
```

3.20 Data 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]),CY=("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";"RS485FALL"
;"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 Output Data Formats

This section describes the output data formats. This chapter is divided into three part; Bottom Track data formats, Water Track data formats and Current Profile data formats. Each of these chapter are divided into two chapters, Binary output formats and ASCII output formats.

4.1 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

4.1.1 Binary

Data format (DF)	Description
21	Nortek DVL Bottom Track data format. See chapter 5.3.
150	RDI PD0
154	RDI PD4
156	RDI PD6

Table 1 Available Binary Data formats for Bottom Track measurements.

4.1.2 ASCII

Data format (DF)	Description
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.

4.2 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

4.2.1 Binary

Data format (DF)	Description
22	Nortek DVL Water Track data format. See chapter 5.3.
150	RDI PD0
154	RDI PD4

Table 3 Available Binary Data formats for Water Track measurements.

4.2.2 ASCII

Data format (DF)	Description
404	NMEA $PNORWT3 including tags.
405	NMEA $PNORWT4 (same as DF354 but no tags)
406	NMEA $PNORWT6 including tags.
407	NMEA $PNORWT7 (same as DF356 but no tags)
408	NMEA $PNORWT8 (sensors) including tags.
409	NMEA $PNORWT9 (Sensors) (same as DF358 but no tags)

Table 4 Available ASCII Data formats for Water Track measurements.

4.3 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.

4.3.1 Binary

Data format (DF)	Description
3	Nortek Current Profile data format. See chapter 5.2.
150	RDI PD0

Table 5 Available Binary Data formats for Current Profile measurements.

4.3.2 ASCII

Data format (DF)	Description
100	NMEA Nortek Prolog format
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 6 Available ASCII Data formats for Current Profile measurements.

5 Binary Data formats

Note: All binary data of the DVL interface are stored/sent as Little Endian.

Each output data packet sent/stored by the AD2CP consists of a Header part and a Data Record part:

Header Synchronization, ID, length and Checksums.
Data Record Data

The following chapters describe the format of the Header and the different variants of the Data Record.

5.1 Header Definition

The Header consists of the following fields:

Field	Size	Description
Sync	8 bits	Always **0xA5**
Header Size	8 bits (unsigned)	Size (number of bytes) of the Header.
ID	8 bits	Defines 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). **0x1B** –DVL Bottom Track **0x1D** –DVL Water Track **0xA0** - String Data Record, eg. GPS NMEA data, comment from the FWRITE command.
Family	8 bits	Defines the Instrument Family. **0x10** – AD2CP Family
Data Size	16 bits (unsigned)	Size (number of bytes) of the following Data Record.
Data Checksum	16 bits	Checksum of the following Data Record.
Header Checksum	16 bits	Checksum of all fields of the Header (excepts the Header Checksum itself).

5.1.1 C-style Header Struct Definition

```
typedef struct
{
        unsigned char  sync;
        unsigned char  hdrSize;
        unsigned char  ID;
        unsigned char  family;
        unsigned short dataSize;
        unsigned short dataChecksum;
        unsigned short hdrChecksum;
} CommandHeader_t;
```

5.1.2 Checksum Definition

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;
}
```

5.2 DF3 - Current Profile Data Record Definition

Field	Size	Format	Resolution/ Unit	Description	
Version	8 bits			Version number of the Data Record Definition. (3)	
offsetOfData	8 bits	Unsigned	#Bytes	Number of bytes from start of record to start of data (velocity/amplitude/correlation)	
Configuration	16 bits			Record Configuration Bit Mask	
				Bit 0	Pressure sensor value valid.
				Bit 1	Temperature sensor value valid.
				Bit 2	Compass sensor values valid.
				Bit 3	Tilt sensor values valid.
				Bit 4	-
				Bit 5	Velocity data included
				Bit 6	Amplitude data included
				Bit 7	Correlation data included.
				Bit 8-15	Unused
Serial Number	32 bits	Unsigned			
Year	8 bits	Unsigned	1 Year	Years since 1900 (see struct tm definition)	
Month	8 bits	Unsigned	1 Month	Jan =0, Feb= 1, etc.(see struct tm definition)	
Day	8 bits	Unsigned	1 Day	(see struct tm definition)	
Hour	8 bits	Unsigned	1 Hour	(see struct tm definition)	
Minute	8 bits	Unsigned	1 Minute	(see struct tm definition)	
Seconds	8 bits	Unsigned	1 Second	(see struct tm definition)	
Microsec100	16 bits	Unsigned	100 µsec		
Speed of Sound	16 bits	Unsigned	0.1 m/s		
Temperature	16 bits	Signed	0.01 Degree Celsius		
Pressure	32 bits	Unsigned	0.001 dBar		
Heading	16 bits	Unsigned	0.01 Deg		
Pitch	16 bits	Signed	0.01 Deg		
Roll	16 bits	Signed	0.01 Deg		
#Beams & Coordinate system & #Cells	16 bits			Definition:	
				Bit 9 - 0	Number of Cells (NC)
				Bit 11 - 10	Coordinate system, b01 : XYZ b10 : BEAM b11 : -
				Bit 15 – 12	Number of Beams (NB)
Cell Size	16 bits	Unsigned	1 mm		
Blanking	16 bits	Unsigned	1 mm		
Nominal Correlation	8 bits	Unsigned	%	The nominal correlation for the configured combination of cell size and velocity range.	
Temperature Pressure Sensor	8 bits	Unsigned	0.2 Deg Celsius	Temperature of Pressure sensor: T = (Val/5) - 4.0	
Battery Voltage	16 bits	Unsigned	0.1 Volt		

<table>
<tr><td>Magnetometer Raw(X-axis)</td><td>16 bits</td><td>Signed</td><td></td><td colspan="3">Magnetometer Raw, X axis value in last measurement interval.</td></tr>
<tr><td>Magnetometer Raw(Y-axis)</td><td>16 bits</td><td>Signed</td><td></td><td colspan="3">Magnetometer Raw, Y axis value in last measurement interval.</td></tr>
<tr><td>Magnetometer Raw(Z-axis)</td><td>16 bits</td><td>Signed</td><td></td><td colspan="3">Magnetometer Raw, Z axis value in last measurement interval.</td></tr>
<tr><td>Accelerometer Raw (X-axis)</td><td>16 bits</td><td>Signed</td><td></td><td colspan="3">Accelerometer Raw X axis value in last measurement interval. (16384 = 1.0)</td></tr>
<tr><td>Accelerometer Raw (Y-axis)</td><td>16 bits</td><td>Signed</td><td></td><td colspan="3">Accelerometer Raw Y axis value in last measurement interval. (16384 = 1.0)</td></tr>
<tr><td>Accelerometer Raw (Z-axis)</td><td>16 bits</td><td>Signed</td><td></td><td colspan="3">Accelerometer Raw Z axis value in last measurement interval. (16384 = 1.0)</td></tr>
<tr><td>Ambiguity Velocity</td><td>16 bits</td><td>Unsigned</td><td>10^(Velocity Scaling) m/s</td><td colspan="3">Ambiguity velocity, corrected for sound velocity, scaled according to Velocity Scaling</td></tr>
<tr><td rowspan="5">Data Set Description</td><td rowspan="5">16 bits</td><td rowspan="5"></td><td rowspan="5"></td><td>Bits</td><td colspan="2">Description</td></tr>
<tr><td>0-3</td><td colspan="2">Physical beam used for 1st data set.</td></tr>
<tr><td>4-7</td><td colspan="2">Physical beam used for 2nd data set.</td></tr>
<tr><td>8-11</td><td colspan="2">Physical beam used for 3th data set.</td></tr>
<tr><td>12-16</td><td colspan="2">Physical beam used for 4th data set.</td></tr>
<tr><td>Transmit Energy</td><td>16 bits</td><td>Unsigned</td><td></td><td colspan="3"></td></tr>
<tr><td>Velocity Scaling</td><td>8 bits</td><td>Signed</td><td></td><td colspan="3">Used to scale velocity data.</td></tr>
<tr><td>Power level</td><td>8 bits</td><td>Signed</td><td>dB</td><td colspan="3">Configured power level</td></tr>
<tr><td>Magnetometer Temperature</td><td>16 bits</td><td>Signed</td><td>Uncalibrated</td><td colspan="3">Magnetometer temperature reading</td></tr>
<tr><td>Real Time Clock Temperature</td><td>16 bits</td><td>Signed</td><td>0.01 Degree Celsius</td><td colspan="3">Real time clock temperature reading</td></tr>
<tr><td>Error</td><td>32 bits</td><td></td><td></td><td colspan="3">See Error Description (version 1)</td></tr>
<tr><td rowspan="4">Status</td><td rowspan="4">32 bits</td><td rowspan="4"></td><td rowspan="4"></td><td>Bit 31-28</td><td>Wakeup State</td><td>10 = break
11 = RTC alarm
00 = bad power
01 = power applied</td></tr>
<tr><td>Bit 27-25</td><td>Orientation</td><td>See Table 7.</td></tr>
<tr><td>Bit 24-22</td><td>Auto orientation</td><td>See Table 8.</td></tr>
<tr><td>Bit 21-18</td><td>Previous Wakeup State</td><td>10 = break
11 = RTC alarm
00 = bad power
01 = power applied</td></tr>
<tr><td>Ensemble counter</td><td>32 bits</td><td>Unsigned</td><td></td><td colspan="3">Counts the number of ensembles in both averaged data and burst data</td></tr>
<tr><td>Velocity data</td><td>NB*NC* 16 bits</td><td>Signed</td><td>10^(Velocity Scaling) m/s</td><td colspan="3">This field exists if the Velocity data included bit of the Config byte is set.</td></tr>
<tr><td>Amplitude data</td><td>NB*NC* 8 bits</td><td>Unsigned</td><td>1 Count</td><td colspan="3">This field exists if the Amplitude data included bit of the Config byte is set.</td></tr>
<tr><td>Correlation data</td><td>NB*NC* 8 bits</td><td>Unsigned</td><td>[0 – 100]</td><td colspan="3">This field exists if the Correlation data included bit of the Config byte is set.</td></tr>
</table>

Value	Instrument Vertical Definition	Description
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
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

Table 7 Orientation Description

Value		Description
0		Fixed orientation
1		Auto Up Down

Table 8 Automatic Orientation Detection Status

5.2.1 DF3 VelocityData Record Struct Definition (C99 standard)

```
typedef struct
{
   unsigned short beamData1      : 4;
   unsigned short beamData2      : 4;
   unsigned short beamData3      : 4;
   unsigned short beamData4      : 4;
} t_DataSetDescription4Bit;

typedef struct
{
   unsigned long _empty1         : 21;
   unsigned long prevWakeUpState : 1;
   unsigned long autoOrient      : 3;
   unsigned long orientation     : 3;
   unsigned long wakeupstate     : 4;
} t_status;


#define VERSION_DATA_STRUCT_3   3

/* Data field */
typedef struct
{
   unsigned char  version; // 3
   unsigned char  offsetOfData; // offsetof(BurstData3_t, data)
   struct  {
      unsigned short pressure     : 1; // 0
      unsigned short temp         : 1; // 1
      unsigned short compass      : 1; // 2
      unsigned short tilt         : 1; // 3
      unsigned short _empty       : 1; // 4
      unsigned short velIncluded  : 1; // 5
      unsigned short ampIncluded  : 1; // 6
      unsigned short corrIncluded : 1; // 7
      unsigned short _unused      : 8;
   } headconfig;
```

```
   unsigned long  serialNumber;

   unsigned char  year;
   unsigned char  month;
   unsigned char  day;
   unsigned char  hour;
   unsigned char  minute;
   unsigned char  seconds;
   unsigned short microSeconds100;
   unsigned short soundSpeed;     /* resolution: 0.1 m/s */
   short          temperature;    /* resolution: 0.01 degree Celsius */
   unsigned long  pressure;
   unsigned short heading;
   short          pitch;
   short          roll;
   unsigned short beams_cy_cells; /* bit 15-12: Number of beams, bit 11-10: coordinate system,
                                     bit 9-0: number of cells. */
   unsigned short cellSize;
   unsigned short blanking;
   unsigned char  nominalCorrelation;
   unsigned char  pressTemperature;
   unsigned short battery;
   short          magnHxHyHz[3];  /* Magnetometer Min data */
   short          accl3D[3];      /* Accelerometer Data  */
   unsigned short ambVelocity;
   t_DataSetDescription4Bit DataSetDescription4bit; /* unsigned short */
   unsigned short transmitEnergy;
   char           velocityScaling;
   char           powerlevel;
   short          magnTemperature;
   short          rtcTemperature;
   unsigned long  error;
   t_status       status;        /* Unsigned long */
   unsigned long  ensembleCounter;

   unsigned char data[SIZE_VAR_DATA];

   /* Actual size of the following =
      int16_t hVel[nBeams][nCells];   // velocity
      uint8_t cAmp[nBeams][nCells];   // amplitude
      uint8_t cCorr[nBeams][nCells];  // correlation  (0-100)
   */
} OutputData3_t;
```

5.3 DF21/DF22 - DVL Bottom Track & Water Track Data Record Definitions

Data format DF21 is used for Bottom Track measurements. Data format DF22 is used for Water Track measurements. The binary definition of the data record is equal. The ID of the record header is used to differ.

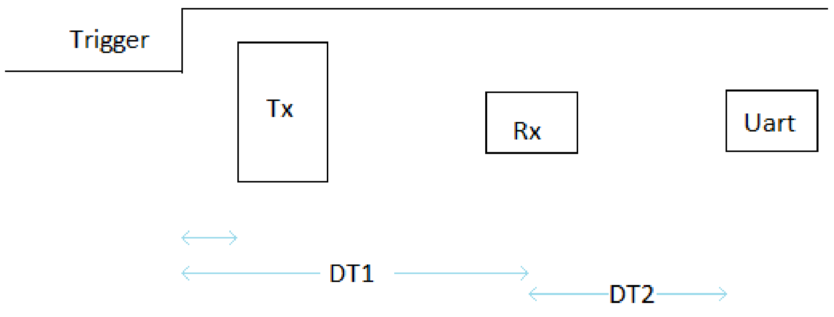
Field	Size	Format	Resolution/ Unit	Description
Version	8 bits	Unsigned		1
offsetOfData	8 bits	Unsigned	#Bytes	Number of bytes from start of record to start of data (velBeam[0])
Serial number	32 bits	Unsigned		
Year	8 bits	Unsigned	1 Year	Years since 1900 (see struct tm definition)
Month	8 bits	Unsigned	1 Month	Jan =0, Feb= 1, etc.(see struct tm definition)
Day	8 bits	Unsigned	1 Day	(see struct tm definition)
Hour	8 bits	Unsigned	1 Hour	(see struct tm definition)
Minute	8 bits	Unsigned	1 Minute	(see struct tm definition)
Seconds	8 bits	Unsigned	1 Second	(see struct tm definition)
Microsec100	16 bits	Unsigned	100 µsec	
#beams	16 bits	Unsigned		Number of beams
Error	32 bits	Unsigned		
Status	32 bits	Unsigned		See Table 9 DVL Status Bit Description.
Sound Speed	32 bits	Float	m/s	
Temperature	32 bits	Float	°Celsius	
Pressure	32 bits	Float	Bar	
Velocity Beam 0	32 bits	Float	m/s	
Velocity Beam 1	32 bits	Float	m/s	
Velocity Beam 2	32 bits	Float	m/s	
Velocity Beam 3	32 bits	Float	m/s	
Distance Beam 0	32 bits	Float	m	Vertical Distance
Distance Beam 1	32 bits	Float	m	Vertical Distance
Distance Beam 2	32 bits	Float	m	Vertical Distance
Distance Beam 3	32 bits	Float	m	Vertical Distance
FOM beam 0	32 bits	Float		Figure of Merit
FOM beam 1	32 bits	Float		Figure of Merit
FOM beam 2	32 bits	Float		Figure of Merit
FOM beam 3	32 bits	Float		Figure of Merit
DT1 Beam 0	32 bits	Float	s	
DT1 Beam 1	32 bits	Float	s	
DT1 Beam 2	32 bits	Float	s	
DT1 Beam 3	32 bits	Float	s	
DT2 Beam 0	32 bits	Float	s	
DT2 Beam 1	32 bits	Float	s	
DT2 Beam 2	32 bits	Float	s	
DT2 Beam 3	32 bits	Float	s	
Time Vel Est Beam 0	32 bits	Float	s	Duration of velocity estimate for each beam.

Field	Size	Format	Resolution/ Unit	Description
Time Vel Est Beam 1	32 bits	Float	s	Duration of velocity estimate for each beam.
Time Vel Est Beam 2	32 bits	Float	s	Duration of velocity estimate for each beam.
Time Vel Est Beam 3	32 bits	Float	s	Duration of velocity estimate for each beam.
Velocity X	32 bits	Float	m/s	
Velocity Y	32 bits	Float	m/s	
Velocity Z1	32 bits	Float	m/s	
Velocity Z2	32 bits	Float	m/s	
FOM X	32 bits	Float		Figure of Merit
FOM Y	32 bits	Float		Figure of Merit
FOM Z1	32 bits	Float		Figure of Merit
FOM Z2	32 bits	Float		Figure of Merit
DT1 X	32 bits	Float	s	
DT1 Y	32 bits	Float	s	
DT1 Z1	32 bits	Float	s	
DT1 Z2	32 bits	Float	s	
DT2 X	32 bits	Float	s	
DT2 Y	32 bits	Float	s	
DT2 Z1	32 bits	Float	s	
DT2 Z2	32 bits	Float	s	
Time Vel Est X	32 bits	Float	s	Duration of velocity estimate for each component.
Time Vel Est Y	32 bits	Float	s	Duration of velocity estimate for each component.
Time Vel Est Z1	32 bits	Float	s	Duration of velocity estimate for each component.
Time Vel Est Z2	32 bits	Float	s	Duration of velocity estimate for each component.

Bit #	Description
0	Beam 1 Velocity Valid
1	Beam 2 Velocity Valid
2	Beam 3 Velocity Valid
3	Beam 4 Velocity Valid
4	Beam 1 Distance Valid
5	Beam 2 Distance Valid
6	Beam 3 Distance Valid
7	Beam 4 Distance Valid
8	Beam 1 Figure of Merit Valid
9	Beam 2 Figure of Merit Valid
10	Beam 3 Figure of Merit Valid
11	Beam 4 Figure of Merit Valid
12	X Velocity Valid
13	Z Velocity Valid
14	Z1 Velocity Valid
15	Z2 Velocity Valid
16	X Figure of Merit Valid
17	Y Figure of Merit Valid
18	Z1 Figure of Merit Valid
19	Z2 Figure of Merit Valid
20	Less than 3% processing capacity left
21	Less than 6% processing capacity left
22	Less than 12% processing capacity left
28-31	Wakeup State: 0010 = break 0011 = RTC alarm 0000 = bad power 0001 = power applied

Table 9 DVL Status Bit Description

Timing information



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.

5.3.1 DF21/22 Bottom Track/Water Track Record Struct Definition (C99 standard)

```
typedef struct
{
   unsigned char  version;
   unsigned char  offsetOfData;
   unsigned long  serialNumber;
   unsigned char  year;            ///< Trigger time
   unsigned char  month;
   unsigned char  day;
   unsigned char  hour;
   unsigned char  minute;
   unsigned char  seconds;
   unsigned short microSeconds100;
   unsigned short nBeams;
   unsigned long  error;
   t_DVLstatus    status;          /* Unsigned long */
   float          soundSpeed;      ///< [m/s]
   float          temperature;     ///< [Celsius]
   float          pressure;        ///< [Bar]

   /* Beam data */
   float          velBeam[4];      ///< Velocities for each beam. [m/s]
   float          distBeam[4];     ///< Distances for each beam.  [m  ]
   float          fomBeam[4];      ///< FOM for each beam.
   float          timeDiff1Beam[4]; ///< DT1 for each beam.       [s  ]
   float          timeDiff2Beam[4]; ///< DT2 for each beam.       [s  ]
   float          timeVelEstBeam[4];///< Duration of velocity estimate for each beam. [s  ]

   /* XYZ data */
   float          velX;            ///< Velocity X.               [m/s]
   float          velY;            ///< Velocity Y.               [m/s]
   float          velZ1;           ///< Velocity Z1.              [m/s]
   float          velZ2;           ///< Velocity Z2.              [m/s]
   float          fomX;            ///< FOM X.
   float          fomY;            ///< FOM Y.
   float          fomZ1;           ///< FOM Z1.
   float          fomZ2;           ///< FOM Z2.
   float          timeDiff1X;      ///< DT1 X.                    [s  ]
   float          timeDiff1Y;      ///< DT1 Y.                    [s  ]
   float          timeDiff1Z1;     ///< DT1 Z1.                   [s  ]
   float          timeDiff1Z2;     ///< DT1 Z2.                   [s  ]
   float          timeDiff2X;      ///< DT2 X.                    [s  ]
   float          timeDiff2Y;      ///< DT2 Y.                    [s  ]
   float          timeDiff2Z1;     ///< DT2 Z1.                   [s  ]
   float          timeDiff2Z2;     ///< DT2 Z2.                   [s  ]
   float          timeVelEstX;     ///< Duration of velocity estimate for each component. [s  ]
   float          timeVelEstY;     ///< Duration of velocity estimate for each component. [s  ]
   float          timeVelEstZ1;    ///< Duration of velocity estimate for each component. [s  ]
   float          timeVelEstZ2;    ///< Duration of velocity estimate for each component. [s  ]
} DVLformat21_t;
```

```
typedef struct
{
   unsigned long beam1VelValid  : 1;  // BIT( 0)
   unsigned long beam2VelValid  : 1;  // BIT( 1)
   unsigned long beam3VelValid  : 1;  // BIT( 2)
   unsigned long beam4VelValid  : 1;  // BIT( 3)
   unsigned long beam1DistValid : 1;  // BIT( 4)
   unsigned long beam2DistValid : 1;  // BIT( 5)
   unsigned long beam3DistValid : 1;  // BIT( 6)
   unsigned long beam4DistValid : 1;  // BIT( 7)
   unsigned long beam1FOMValid  : 1;  // BIT( 8)
   unsigned long beam2FOMValid  : 1;  // BIT( 9)
   unsigned long beam3FOMValid  : 1;  // BIT(10)
   unsigned long beam4FOMValid  : 1;  // BIT(11)
   unsigned long xVelValid      : 1;  // BIT(12)
   unsigned long yVelValid      : 1;  // BIT(13)
   unsigned long z1VelValid     : 1;  // BIT(14)
   unsigned long z2VelValid     : 1;  // BIT(15)
   unsigned long xFOMValid      : 1;  // BIT(16)
   unsigned long yFOMValid      : 1;  // BIT(17)
   unsigned long z1FOMValid     : 1;  // BIT(18)
   unsigned long z2FOMValid     : 1;  // BIT(19)
   unsigned long procIdle3      : 1;
   unsigned long procIdle6      : 1;
   unsigned long procIdle12     : 1;
   unsigned long _empty1        : 5;
   unsigned long wakeupstate    : 4;  // BIT(28-31)
} t_DVLstatus;
```

6 ASCII Data Formats

6.1 DVL Bottom Track ASCII formats

6.1.1 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	f.ff	FOM=12.34
D	Vertical Distance to bottom. [m]	ff.f	D=12.3

Table 10 PNORBT3/4 NMEA sentence parameter description.

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=12.34,D=12.3*51
```

Example (DF=355):
```
$PNORBT4,1.234,-1.234,1.234,23.4,12.34,12.3*09
```

6.1.2 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 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	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

Table 11 PNORBT6/7 NMEA sentence parameter description.

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=12.34,D1=23.45,D2=23.45,D3=23.45,D4=2
3.45*5E
```

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
```

6.1.3 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	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 9)	0xHHHHHHHH	STAT=0x000FFFFF

Table 12 PNORBT8/9 NMEA sentence parameter description.

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=2
3.45,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
```

6.2 DVL Water Track ASCII formats

6.2.1 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	f.ff	FOM=12.34
D	Vertical Distance to water track cell. [m]	ff.f	D=12.3

Table 13 PNORWT3/4 NMEA sentence parameter description.

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
```

6.2.1 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	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

Table 14 PNORWT6/7 NMEA sentence parameter description.

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=2
3.45*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
```

6.2.1 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	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 9)	0xHHHHHHHH	STAT=0x000FFFFF

Table 15 PNORBWT8/9 NMEA sentence parameter description.

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=2
3.45,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
```

6.3 DVL Current Profile ASCII formats

6.3.1 DF100 - Prolog NMEA Format

See Prolog Data Format Description.

6.3.2 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

Table 16 PNORI1/2 NMEA sentence parameter description

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]	H	ddd.d	H=123.4
Heading Std.Dev.	[deg]	HSD	dd.dd	HSD=0.02
Pitch	[deg]	PI	dd.d	PI=45.6
Pitch Std.Dev	[deg]	PISD	dd.dd	PISD=0.02
Roll	[deg]	R	dd.d	R=23.4
Roll Std.Dev.	[deg]	RSD	dd.dd	RSD=0.02
Pressure	[dBar]	P	ddd.ddd	P=123.456
Pressure StdDev	[dBar]	PSD	dd.dd	PSD=0.02
Temperature	[deg C]	T	dd.dd	T=24.56

Table 17 PNORS1/2 NMEA sentence parameter description

Example (DF=101):

```
$PNORS1,083013,132455,0,34000034,23.9,1500.0,123.4,0.02,45.6,0.02,R=23.4,
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	DATE	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] Only if CY=ENU and #beams = 4	VU2	dd.ddd	VU2=0.332
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

Field	Description	TAG	Data format	Example
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

Table 18 PNORC1/2 NMEA sentence parameter description

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

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

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

```
$PNORC2,DATE=083013,TIME=132455,CN=3,CP=11.0,VE=0.332,VN=0.332,VU=0.332,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
```

6.3.1 DF103/DF104

6.4 DVL Bottom Track

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

The NMEA formats 300/301 have the following fields:

Field/TAG	Description	Data format	Example
BEAM	Beam number	n	BEAM=3
DATE	Date	MMDDYY	DATE=112813
TIME	Time	hhmmss.ssss	TIME=072228.2345
DT1	Diff. time 1 [ms]	s.ssss	DT1=0.1234
DT2	Diff. time 2 [ms]	s.ssss	DT2=0.1234
BV, VY,VZ	Bottom Velocity [m/s]	f.fffff	BV=1.11111
FM	Figure of Merit	f.f	FM=122.2
DIST	Distance [meter]	f.ff	DIST=36.66
WV	Water Velocity [m/s]	f.fffff	WV=2.22222
STAT	Status	hh	STAT=F7

Table 19 PNORBT NMEA sentence parameter description.

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=300):

```
$PNORBT,3,112813,072228.2345,0.1234,0.1234,1.11111,122.2,36.66,2.22222,F7*7A
```

Example (DF=301):

```
$PNORBT,BEAM=3,DATE=112813,TIME=072228.2345,DT1=0.1234,DT2=0.1234,BV=1.11111,
FM=122.2,DIST=36.66,WV=2.22222,STAT=F7*75
```