

NORTEK MANUALS

Integrator's Guide

Generation 2 Vector



Table of Contents

Ch. 1	Introduction	5
1.1	Modes	7
Ch. 2	Basic interface concepts	7
2.1	Break	8
Ch. 3	Commands	9
3.1	List of Commands	11
3.2	Wake processor from sleep mode	14
3.3	Instrument main settings	14
3.4	Clock commands	16
3.5	Clock as string	16
3.6	Plan settings	17
3.7	Burst settings	18
3.8	Echosounder settings	19
3.9	Read memory usage	20
3.10	Sync settings	21
3.11	Read power use	22
3.12	Read vertical velocity range	22
3.13	Read absolute pressure	23
3.14	Echo measurement miscellaneous settings	23
3.15	Velocity measurement settings	24
3.16	Velocity measurement miscellaneous settings	25
3.17	Velocity measurement information	25
3.18	External sensor settings	26
3.19	Probe check settings	26
3.20	Instrument user settings	27
3.21	Read instrument ID	28
3.22	Frequency spectrum settings	29
3.23	Load default settings	30
3.24	Battery measurement information	30
3.25	AHRS settings	31
3.26	Velocity telemetry settings	31
3.27	Ahrs telemetry settings	32
3.28	Save settings	33
3.29	Deploy instrument	33
3.30	Start instrument	34
3.31	Enter command mode	34

3.32	Enter measurement mode	35
3.33	Power down	35
3.34	Format recorder	35
3.35	Read recorder status	36
3.36	List files	36
3.37	List configuration	36
3.38	Inquire state	37
3.39	Read instrument state	38
3.40	Read error	39
3.41	List all	39
3.42	Get the "Beam to XYZ" transformation matrix	41
3.43	Read hardware specifications	41
3.44	Read firmware version	42
3.45	List versions for all software components	43
3.46	Add license	43
3.47	Delete license	44
3.48	Lists license keys	44
3.49	Read spectrum	45
3.50	Modbus write	45
3.51	Modbus list	46
Ch. 4	Binary data formats	47
4.1	_HeaderData	48
4.2	NortekDataFormat8	49
4.3	_CommonData	61
4.4	_DF3 CurrentProfileData	64
4.5	DF3 SpectrumData	68
4.6	StringData	69
Ch. 5	Telemetry Data Formats	69

1 Introduction

The primary objective of this manual is to provide the information needed to control the Generation 2 Vector. It is aimed at system integrators and engineers with interfacing experience. The document's scope is limited to interfacing and does not address general performance issues of the instrument. For a more thorough understanding of the principles, we recommend the series of [Principles of Operation](#) documents. For more information about the difference between Generation 2 Vector and the previous generation please refer to [Release notes for the generation 2 Vector](#). For most users, it will make sense to let the supplied Nortek software do most of the hardware configuration and then let the controller limit its task to starting/stopping data collection.

The document is complete in the sense that it describes all available commands and modes of communication.

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 2025.1	03/06/2025	First version
Version 2026.1	28/06/2026	• Removed RS232 as option in SETINST (this was a copy paste error from another instrument, as we never supported RS232 on the main serial command interface) • Removed FULL as option in SETSPECTRUM (another copy paste error, this is not implemented yet)

		• Increased maximum value of burst interval to 86400 seconds

1.1 Modes

The instrument operates in distinct modes. These modes have several explicit commands used 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 Retrieval: Data download from recorder (not yet implemented)
- **Measurement:** Data collection mode
- **Confirmation:** Confirmation mode

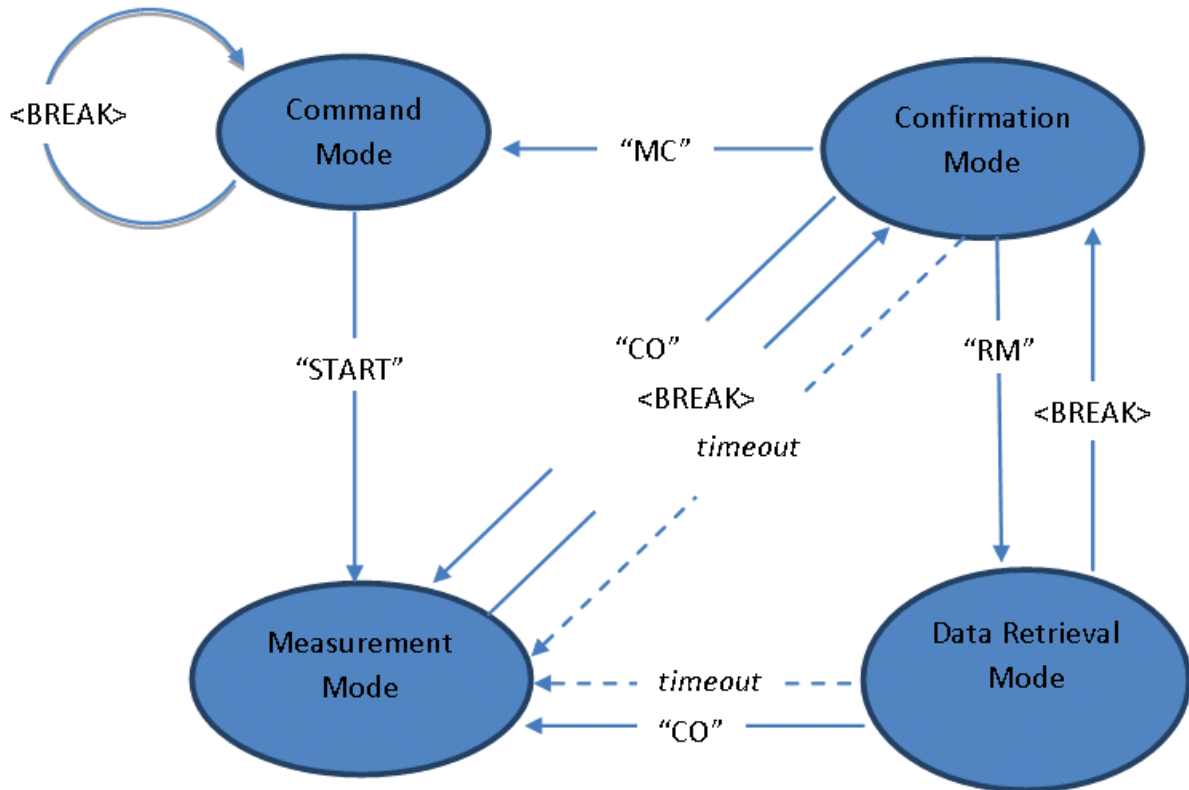


Figure 1: Instrument modes of operation

Initializing communication with the instrument is performed by sending a **<BREAK>**, which is defined in the next section. 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 Figure 1). 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, the instrument will go into sleep mode and a break or a sequence of @@@@ must be sent to wake up the processor.

2 Basic interface concepts

The command interface for the Vector 2 is ASCII based and line oriented. Before diving into the chapters covering data formats and commands, the operational modes and how to change between the modes are described. Understanding the use and constraints of the modes is important as they are used frequently when communicating with the instrument.

3 Commands

This chapter covers the commands that can be used to control an Vector generation 2 instrument. Please note that not every command is relevant for every instrument. The commands relevant for your specific instrument depends on the instrument type and licenses.

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.:

```
SETAVG, NC=10, BD=0.7  
OK
```

- Sometimes you may get an ERROR response after trying 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.

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,GETAVGLIM*22
```

```
$PNOR,GETAVGLIM,NC=([1;128]),CS=([0.25;2.00]),BD=([0.10;45.00]),CY=
("BEAM"),PL=([-40.0;0.0];-
```

```
100.0),AI=([1;300]),VP=([0.000;0.100]),VR=([1.25;5.00]),DF=([0;3]),
```

```
NPING=([1;4])*46
```

```
$PNOR,OK*2B
```

Regular interface example:

```
GETPLANLIM
```

```
([1;3600]),(0;1),(),([0;2]),(),([0.0;50.0]),(0;1),([10;21600]),(),
```

```
([1300.00;1700.00];0.0),(['0';'9'];['a';'z'];['A';'Z'];'.'),(0;1)
```

```
OK
```

3.1 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
BREAK	Wakes up the processor when it is in sleep mode	ALL
SETINST	Set instrument main settings	COMMAND
GETINST	Get instrument main settings	COMMAND
GETINSTLIM	Set instrument main setting limits	COMMAND
WRITECLOCK	Set instrument clock.,Note! All parameters must be provided.	COMMAND RETRIEVAL
READCLOCK	Read instrument clock	COMMAND RETRIEVAL
WRITECLOCKSTR	Write instrument clock as string	COMMAND RETRIEVAL
READCLOCKSTR	Read instrument clock as string	COMMAND RETRIEVAL
SETPLAN	Set plan settings	COMMAND
GETPLAN	Get plan settings	COMMAND
GETPLANLIM	Get plan setting limits	COMMAND
SETBURST	Set burst settings	COMMAND
GETBURST	Get burst settings	COMMAND
GETBURSTLIM	Get burst setting limits	COMMAND
SETECHO	Set echosounder settings	COMMAND
GETECHO	Get echosounder settings	COMMAND
GETECHOLIM	Get echosounder setting limits	COMMAND
READMEM	Read recorder data memory usage	COMMAND
SETSYNC	Set sync settings	COMMAND
GETSYNC	Get sync settings	COMMAND
GETSYNCLIM	Get sync setting limits	COMMAND
READPWR	Returns the power consumption in mW for the various measurements enabled as well as the overall value	COMMAND
READVR	Read Vertical Velocity Range present configuration in m/s	COMMAND

READABSPRESSURE	Returns the absolute pressure reading from the instrument	COMMAND
SETECHOMISC	Set echo measurement miscellaneous settings	COMMAND
GETECHOMISC	Get echo measurement miscellaneous settings	COMMAND
GETECHOMISCLIM	Get echo miscellaneous limits	COMMAND
SETVEL	Set velocity measurement settings	COMMAND
GETVEL	Get velocity measurement settings	COMMAND
GETVELLIM	Get velocity limits	COMMAND
SETVELMISC	Set velocity measurement miscellaneous settings	COMMAND
GETVELMISC	Get velocity measurement miscellaneous settings	COMMAND
GETVELMISCLIM	Get velocity miscellaneous limits	COMMAND
READVEL	Read velocity measurement information	COMMAND
SETEXTSENSOR	Set external sensor settings	COMMAND
GETEXTSENSOR	Get external sensor settings	COMMAND
GETEXTSENSORLIM	Get external sensor limits	COMMAND
SETPROBE	Set probe settings	COMMAND
GETPROBE	Get probe settings	COMMAND
GETPROBELIM	Get probe limits	COMMAND
SETUSER	Set instrument user settings	COMMAND
GETUSER	Get instrument user settings	COMMAND
GETUSERLIM	Get instrument calibration parameter limits	COMMAND
READID	Get instrument Id	COMMAND CONFIRMATION MEASUREMENT RETRIEVAL
SETSPECTRUM	Set frequency spectrum settings	COMMAND
GETSPECTRUM	Get frequency spectrum settings	COMMAND
GETSPECTRUMLIM	Get frequency spectrum settings limits	COMMAND
LOADDEFAULT	Reload default settings	COMMAND
READBATTERY	Read battery measurement	COMMAND
SETAHRS	Set AHRS settings	COMMAND
GETAHRS	Get AHRS settings	COMMAND
GETAHRSLIM	Get limits for AHRS settings	COMMAND
SETTMVEL	Set velocity telemetry settings	COMMAND
GETTMVEL	Get velocity telemetry settings	COMMAND

GETTMVELLIM	Get instrument velocity telemetry limits	COMMAND
SETTMAHRS	Set AHRS telemetry settings	COMMAND
GETTMAHRS	Get AHRS telemetry settings	COMMAND
GETTMAHRSLIM	Get instrument AHRS telemetry limits	COMMAND
SAVE	Save settings for next measurement	COMMAND
DEPLOY	Deploy the instrument	COMMAND
START	Start the instrument	COMMAND
MC	Go into command mode	COMMAND CONFIRMATION
CO	Go into measurement mode	CONFIRMATION RETRIEVAL
POWERDOWN	Set instrument in sleep mode	COMMAND
FORMAT	Format the recorder	COMMAND
READRECSTATUS	Return recorder status	COMMAND RETRIEVAL
LISTFILES	Lists the files on instrument	COMMAND RETRIEVAL
LISTCFG	List current configuration	COMMAND
INQ	Inquires the instrument state	COMMAND CONFIRMATION RETRIEVAL MEASUREMENT
READSTATE	Returns information about the current operational state of the instrument	COMMAND CONFIRMATION MEASUREMENT RETRIEVAL
READERROR	Returns a full description of the last error condition to occur	COMMAND CONFIRMATION MEASUREMENT RETRIEVAL
LISTALL	Retrieves all relevant configuration information for the instrument	COMMAND
READXFVEL	Get the "Beam to XYZ" transformation matrix	COMMAND
READHW	Returns hardware revisions.	COMMAND CONFIRMATION MEASUREMENT RETRIEVAL
READFW	Read firmware version	COMMAND CONFIRMATION

		MEASUREMENT RETRIEVAL
LISTVER	List versions for all software components	COMMAND
ADDLICENSE	Add license key	COMMAND
DELETELICENSE	Delete license key	COMMAND
LISTLICENSE	Lists all license keys in instrument	COMMAND
READSPECTRUM	Reads the spectrum.	COMMAND
WRITEMODBUS	Perform a modbus write transaction (FC16). This command writes to a single Modbus address per transaction.	COMMAND
LISTMODBUS	Perform a modbus read transaction (FC03, FC04) and list the register values.	COMMAND

3.2 Wake processor from sleep mode

Command: BREAK

Command type: ACTION

Mode: ALL

Used to wake up the processor when it is in sleep mode.

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.

See more info on Break

Example:

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

3.3 Instrument main settings

Commands: SETINST, GETINST, GETINSTLIM,

Command type: CONFIGURATION

Access: User

Mode: COMMAND

Instrument main settings

Argument	Description
BR	Baud Rate 115200, 230400, 460800, 921600

	Unit: [bit/s]
RS	Serial protocol 422
LED	Enable/disable LED blink in head. When set to "ON24H" the LED will illuminate the first 24 hours of the measurement. ON, OFF, ON24H
ORIENT	Sets the instrument orientation. Not used for DVL. AUTOXUPDOWN, AUTOYUPDOWN, AUTOZUPDOWN, ZUP, ZDOWN, XUP, XDOWN, YUP, YDOWN, AUTO3D, AHRS3D
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=230400
```

OK

GETINST

Get instrument main settings

Example:

```
GETINST, BR=460800, RS=422, LED="ON", ORIENT="AUTOZUPDOWN", CMTOUT=300, DMTOUT=60, CFMTOUT=60
```

GETINSTLIM

Set instrument main setting limits

3.4 Clock commands

Commands: WRITECLOCK, READCLOCK,

Command type: CONFIGURATION

Access: User

Mode: COMMAND, RETRIEVAL

Set and read instrument clock

Argument	Description
YEAR	Year (four digits)
MONTH	Month 1-12 (Jan = 1)
DAY	Day of month (1-31)
HOURL	Hour of day (0-23)
MINUTE	Minute of hour (0-59)
SECOND	Seconds of minute (0-59)

WRITECLOCK

Set instrument clock.,Note! All parameters must be provided.

Example:

WRITECLOCK, YEAR=2024, MONTH=3, DAY=11, HOUR=8, MINUTE=16, SECOND=9

OK

READCLOCK

Read instrument clock

Example:

\$PNOR, READCLOCK*75

\$PNOR, READCLOCK, YEAR=2024, MONTH=3, DAY=11, HOUR=8, MINUTE=16, SECOND=9*
59

\$PNOR, OK*2B

3.5 Clock as string

Commands: WRITECLOCKSTR, READCLOCKSTR,

Command type: CONFIGURATION

Access: User

Mode: COMMAND, RETRIEVAL

Write and read instrument clock as string

Argument	Description
TIME	Text string on the following format: yyyy-MM-dd HH:mm:ss (use UTC)

WRITECLOCKSTR

Write instrument clock as string

Example:

```
WRITECLOCKSTR, TIME="2024-02-11 08:16:09"
```

OK

READCLOCKSTR

Read instrument clock as string

Example:

```
READCLOCKSTR
```

```
"2024-02-11 08:16:09"
```

OK

3.6 Plan settings

Commands: SETPLAN, GETPLAN, GETPLANLIM,

Command type: CONFIGURATION

Mode: COMMAND

Plan profile settings and relevant limits.

Argument	Description
SA	Salinity value.
SV	Sound velocity. When set to 0, the sound velocity is computed from water temperature, pressure and salinity.
BURST	Enable/disable burst measurements.
FN	Filename. Values: ['a'; 'z']; ['A'; 'Z']; ['0'; '9']; '_' Max Length: 30
SO	Enable/disable velocity measurement.

SETPLAN

Set plan settings

GETPLAN

Get plan settings

GETPLANLIM

Get plan setting limits

3.7 Burst settings

Commands: SETBURST, GETBURST, GETBURSTLIM,

Command type: CONFIGURATION

Mode: COMMAND

Burst profile settings and relevant limits.

Argument	Description
MI	Measurement interval. Unit: [s]
DURATION	Measurement duration. Unit: [sec]
VEL	Enable/disable velocity measurement.
ECHO	Enable/disable echo measurement.
SPECTRUM	Enable/disable spectrum measurement.
EXTSENS	Enable/disable external sensor .
AHRS	Enable/disable AHRS.
PROBE	Enable/disable probe check.

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

SETBURST

Set burst settings

GETBURST

Get burst settings

GETBURSTLIM

Get burst setting limits

3.8 Echosounder settings

Commands: SETECHO, GETECHO, GETECHOLIM,

Command type: CONFIGURATION

Mode: COMMAND

Echosounder settings and relevant limits

Argument	Description
SR	Sampling rate. Unit: [Hz]
NBINS	Number of echsounder bins
BINSIZE	Bin size in mm. Min. value is 2.0 mm Unit: [mm]
BD	Blanking distance Unit: [m]
DF	Dataformat. 8: DF8 Binary
FREQ1	Enable and set frequency 1 of echogram. Unit: [kHz]
XMIT1	Transmission length on frequency 1. Unit: [mm]
PL1	Power level on frequency 1 (range -13.0 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]
PULSECOMP1	Enable/disable pulse compression on frequency 1.
CH1	Frequency 1 channel beam selection. E.g. CH=123 will enable beams 1, 2 and 3.
RAW1	Frequency 1 raw echo data recording interval.
FREQ2	Enable and set frequency 2 of echogram. Unit: [kHz]
XMIT2	Transmission length on frequency 2. Unit: [mm]
PL2	Power level on frequency 2 (range -13.0 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]

PULSECOMP2	Enable/disable pulse compression on frequency 2.
CH2	Frequency 2 channel beam selection. E.g. CH=123 will enable beams 1, 2 and 3.
RAW2	Frequency 2 raw echo data recording interval.

Note: Note that Pulse Compression may only be enabled for one echogram.

SETECHO

Set echosounder settings

Example:

SETECHO, SR=10, BINSIZE=2

OK

GETECHO

Get echosounder settings

GETECHOLIM

Get echosounder setting limits

3.9 Read memory usage

Command: READMEM

Command type: INFO

Access: User

Mode: COMMAND

Recorder data memory usage in MB/day for the current configuration and sub configurations.

Argument	Description
TOTAL	Total memory usage Unit: [MB/day]
VEL	Velocity memory usage Unit: [MB/day]
ECHO	Echo memory usage Unit: [MB/day]
EXTSENS	External sensors memory usage Unit: [MB/day]
AHRS	AHRS sensors memory usage

	Unit: [MB/day]
--	----------------

3.10 Sync settings

Commands: SETSYNC, GETSYNC, GETSYNCLIM,

Command type: CONFIGURATION

Access: User

Mode: COMMAND

With EN=1 and master for Vector is set we shall output a 1.95 ms (1/512 Hz) pulse on the sync output which shall be driven

through the RS485 sync interface. It shall start with rising or falling edge as configured. When sync is set to being

a master for other, the same sync pulse shall be output but now at the middle of the velocity pings for each sample.

Argument	Description
EN	Enable/disable external sync functionality.
MODE	Specifies sync type and are given as counting numbers. The available sync types will depend on the harness/cable used with the instrument. "MASTER_VECTOR": A 1.95 ms (1/512 Hz) pulse on the sync output which shall be driven through the RS485 sync interface. "MASTER_OTHER": A 1.95 ms (1/512 Hz) pulse on the sync output which shall be driven through the RS485 sync interface. At the middle of the velocity pings for each sample "SLAVE_PPS": Slave mode.
TYPE	Specifies sync type and are given as counting numbers. The available sync types will depend on the harness/cable used with the instrument. "RISING": The instrument will trigger on the rising edge of the signal. "FALLING": The instrument will trigger on the falling edge of the signal. "EDGE": The instrument will trigger on both edges of the signal.

SETSYNC

Set sync settings

Example:

```
SETSYNC, EN=1, MODE="MASTER_VECTOR", TYPE="RISING"
```

OK

GETSYNC

Get sync settings

GETSYNCLIM

Get sync setting limits

3.11 Read power use

Command: READPWR

Command type: INFO

Access: User

Mode: COMMAND

Power usage.

Argument	Description
TOTAL	Total power usage Unit: [mW]
VEL	Velocity power usage Unit: [mW]
ECHO	Echo power usage Unit: [mW]
EXTSENS	External sensors power usage Unit: [mW]
AHRS	AHRS sensors power usage Unit: [mW]

3.12 Read vertical velocity range

Command: READVR

Command type: INFO

Access: User

Mode: COMMAND

Read actual Velocity Ranges for present configuration in m/s.

Argument	Description
VERT	Vertical velocity range Unit: [m/s]
HORZ	Horizontal velocity range Unit: [m/s]

3.13 Read absolute pressure

Command: READABSPRESSURE

Command type: INFO

Access: User

Mode: COMMAND

Returns the current absolute pressure measurement from the instrument in dBar.

Argument	Description
PRESSURE	Absolute pressure value returned by the instrument. Unit: [dBar]

Example:

READABSPRESSURE

10.812

OK

3.14 Echo measurement miscellaneous settings

Commands: SETECHOMISC, GETECHOMISC, GETECHOMISCLIM,

Command type: CONFIGURATION

Mode: COMMAND

Echo measurement miscellaneous settings.

Argument	Description
EN	Enable echo extended blanking
BD	Echo blanking distance. Unit: [m]

SETECHOMISC

Set echo measurement miscellaneous settings

GETECHOMISC

Get echo measurement miscellaneous settings

GETECHOMISCLIM

Get echo miscellaneous limits

3.15 Velocity measurement settings

Commands: SETVEL, GETVEL, GETVELLIM,

Command type: CONFIGURATION

Mode: COMMAND

Velocity measurement settings.

Argument	Description
SR	Sampling rate. 1, 2, 4, 8, 16, 32, 64 Unit: [Hz]
PL	Power level. Unit: [dB]
DF	Data format. 8: DF8 Binary
XMIT	Transmit length. Unit: [mm]
RECV	Receive length. Unit: [mm]
VR	Velocity range. Unit: [m/s]
CY	Coordinate system ENU, XYZ, BEAM
NPINGS	Number of pings for averaging
BW	Bandwidth NARROW, BROAD, ULTRA

SETVEL

Set velocity measurement settings

GETVEL

Get velocity measurement settings

GETVELLIM

Get velocity limits

3.16 Velocity measurement miscellaneous settings

Commands: SETVELMISC, GETVELMISC, GETVELMISCLIM,

Command type: CONFIGURATION

Mode: COMMAND

Velocity measurement miscellaneous settings.

Argument	Description
EN	Enable extended velocity range
VRFINE	Fine velocity range. Unit: [m/s]
EVR	Extended velocity range [0-6]. Values: [0; 6]

SETVELMISC

Set velocity measurement miscellaneous settings

GETVELMISC

Get velocity measurement miscellaneous settings

GETVELMISCLIM

Get velocity miscellaneous limits

3.17 Velocity measurement information

Command: READVEL

Command type: INFO

Access: User

Mode: COMMAND

Velocity measurement information.

Argument	Description
SVOL	Sampling volume. Unit: [mm]
VAMBBEAM	Ambiguity velocity. Unit: [m/s]

3.18 External sensor settings

Commands: SETTEXTSENSOR, GETTEXTSENSOR, GETTEXTSENSORALIM,

Command type: CONFIGURATION

Access: User

Mode: COMMAND

External sensor configuration settings.

If enabled, the configured sensor will be sampled as close as possible to the configured sampling rate.

The sensor output will be included in velocity data formats.

Argument	Description
EN	Enable/disable external sensor.
ID	Modbus ID of external sensor.
TYPE	Type of external sensor. "NOSENSOR": No sensor "AQUAPHOX": PyroScience AquapHOx-Tx oxygen sensor "RINKO_EC": RINKO-EC oxygen sensor "SEAPOINT_STM_S": Seapoint STM-S turbidity sensor
SR	Sampling rate. Set to 0 to sample sensor together with velocity measurements. Note that the actual maximum sampling rate depends on sensor type. This will be checked when the configuration is saved. Unit: [Hz]

SETTEXTSENSOR

Set external sensor settings

GETTEXTSENSOR

Get external sensor settings

GETTEXTSENSORALIM

Get external sensor limits

3.19 Probe check settings

Commands: SETPROBE, GETPROBE, GETPROBELIM,

Command type: CONFIGURATION

Mode: COMMAND

Probe check settings.

Argument	Description
SR	Sampling rate. 1, 2, 4, 8 Unit: [Hz]
DF	Dataformat. 8: DF8 Binary
NC	Number of cells
BD	Blanking distance Unit: [m]
PL	Powerlevel. Unit: [dB]
NPINGS	Number of pings for averaging Unit: [-]

SETPROBE

Set probe settings

GETPROBE

Get probe settings

GETPROBELIM

Get probe limits

3.20 Instrument user settings

Commands: SETUSER, GETUSER, GETUSERLIM,

Command type: CONFIGURATION

Access: User

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]
DECL	Magnetic declination. Unit: [deg]

HX	Hard iron x-component. Unit: [Gauss]
HY	Hard iron y-component. Unit: [Gauss]
HZ	Hard iron z-component. Unit: [Gauss]

SETUSER

Set instrument user settings

Example:

```
SETUSER,POFF=4.5,DECL=1.4,HX=0.1,HY=0.1,HZ=0.1  
OK  
SAVE,USER  
OK
```

GETUSER

Get instrument user settings

GETUSERLIM

Get instrument calibration parameter limits

3.21 Read instrument ID

Command: READID

Command type: INFO

Access: User

Mode: COMMAND, CONFIRMATION, MEASUREMENT, RETRIEVAL

Commands for accessing instrument name and serial number

Argument	Description
STR	Instrument name
SN	Serial number

Example:

```
READID  
"Vector 2",500146  
READID,STR
```

"Vector 2"

3.22 Frequency spectrum settings

Commands: SETSPECTRUM, GETSPECTRUM, GETSPECTRUMLIM,

Command type: CONFIGURATION

Mode: COMMAND

Configures the spectrum analyzer tool to identify external noise sources or interference.

Argument	Description
SR	Sampling rate. Unit: [Hz]
BW	Bandwidth NARROW, BROAD, ULTRA
DEC	Decimate raw data.
NFFT	Number of FFT bins. 256, 512, 1024, 2048
NB	Number of beams.
CHFULL	Selects the channel (beam) to use when bandwidth is FULL.
CH	Selects the channel (beam) to use when bandwidth is not FULL. Example: 123 will enable beams 1, 2 and 3. 1, 2, 3, 12, 13, 23, 123
DF	Data format. 3: DF3 Binary
NPINGS	Number of pings for averaging
BD	Blanking distance Unit: [m]

SETSPECTRUM

Set frequency spectrum settings

Example:

SETSPECTRUM, SR=1, BW="NARROW", NFFT=1024, CH=123

OK

GETSPECTRUM

Get frequency spectrum settings

GETSPECTRUMLIM

Get frequency spectrum settings limits

3.23 Load default settings

Command: LOADDEFAULT

Command type: ACTION

Access: User

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.
INST	Restore instrument main settings to default.
USER	Restore user calibration to default.

Example:

LOADDEFAULT, CONFIG

OK

3.24 Battery measurement information

Command: READBATTERY

Command type: INFO

Access: User

Mode: COMMAND

Battery measurement information.

Argument	Description
VOLTAGE	Voltage Unit: [V]
CURRENT	Current Unit: [A]
POWER	Power Unit: [W]

Example:

READBATTERY

16.0000, 0.1500, 2.4000

OK

```
READBATTERY, VOLTAGE  
16.0000  
OK
```

3.25 AHRS settings

Commands: SETAHRS, GETAHRS, GETAHRS LIM,

Command type: CONFIGURATION

Access: User

Mode: COMMAND

AHRS sensor settings

Argument	Description
SR	Output frequency. Set to 0 to sample together with velocity measurements. 0, 1, 2, 4, 8, 16, 32, 64 Unit: [Hz]
DF	Data format. 8: DF8 Binary
ALG	Which AHRS algorithm to use. "STATIC": Static compass at 1Hz sampling rate. Based on magnetometer and accelerometer. "DYNAMIC": Dynamic AHRS. Requires AHRS license. "IMU": Delta angles? tbd.

SETAHRS

Set AHRS settings

GETAHRS

Get AHRS settings

GETAHRS LIM

Get limits for AHRS settings

3.26 Velocity telemetry settings

Commands: SETTMVEL, GETTMVEL, GETTMVEL LIM,

Command type: CONFIGURATION

Mode: COMMAND

The instrument velocity telemetry settings and relevant limits.

Argument	Description
EN	Enable velocity telemetry
NS	Number of velocity samples to output. If set to 0, all samples will be outputted.
CY	Co-ordinate System ENU, XYZ, BEAM
FO	Enable file output
SO	Enable serial output
DF	Telemetry data format. 700

SETTMVEL

Set velocity telemetry settings

Example:

SETTMVEL, EN=1, NS=1

GETTMVEL

Get velocity telemetry settings

GETTMVELLIM

Get instrument velocity telemetry limits

3.27 Ahrs telemetry settings

Commands: SETTMAHRS, GETTMAHRS, GETTMAHRSLIM,

Command type: CONFIGURATION

Mode: COMMAND

The instrument AHRS telemetry settings and relevant limits.

Argument	Description
EN	Enable AHRS telemetry
FO	Enable file output
SO	Enable serial output
DF	Telemetry data format.

	750
--	-----

SETTMAHRS

Set AHRS telemetry settings

GETTMAHRS

Get AHRS telemetry settings

GETTMAHRSLIM

Get instrument AHRS telemetry limits

3.28 Save settings

Command: SAVE

Command type: ACTION

Access: User

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.
INST	Save INST settings.
USER	Save user instrument settings.

Example:

SAVE, CONFIG

OK

3.29 Deploy instrument

Command: DEPLOY

Command type: ACTION

Access: User

Mode: COMMAND

Deploy the instrument. Start the measurement at the time specified in the string argument. The format must be exactly as shown. If no time value is passed, the deployment will start immediately.

The number of seconds to the specified deployment time is returned.

The DEPLOY command will save the configuration as well as deploying the instrument, as if a SAVE,CONFIG command were sent.

Argument	Description
TIME	yyyy-MM-dd HH:mm:ss

Note: The difference between DEPLOY and START is that the latter will immediately start a measurement any time the instrument state returns to Measurement mode such as by applying power or timeout from Data Retrieval Mode. If DEPLOY is used, be aware that if the deployment time has passed when the battery is connected, the instrument will resynchronize its data sampling according to the deployment time and the instrument configuration. This means you may have to wait for one average measurement interval or one burst measurement interval before the instrument starts to ping.

Example:

```
DEPLOY,TIME="2020-12-18 15:30:00"  
592  
OK
```

3.30 Start instrument

Command: START

Command type: ACTION

Mode: COMMAND

Start the instrument (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.

3.31 Enter command mode

Command: MC

Command type: ACTION

Access: User

Mode: COMMAND, CONFIRMATION

Sets instrument in command mode from confirmation mode.

Example:

```
MC  
OK
```

3.32 Enter measurement mode

Command: CO

Command type: ACTION

Access: User

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

OK

3.33 Power down

Command: POWERDOWN

Command type: ACTION

Access: User

Mode: COMMAND

Power down the instrument to set it in sleep mode.

Example:

POWERDOWN

OK

3.34 Format recorder

Command: FORMAT

Command type: ACTION

Access: User

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

OK

3.35 Read recorder status

Command: READRECSTATUS

Command type: INFO

Access: User

Mode: COMMAND, RETRIEVAL

Returns recorder status.

Argument	Description
VS	Volume Size in bytes. Unit: [bytes]
FS	Available size in bytes. Unit: [bytes]
TF	Total number of files.
FF	Number of free files.

3.36 List files

Command: LISTFILES

Command type: INFO

Access: User

Mode: COMMAND, RETRIEVAL

Lists the files stored on the instrument recorder.

Argument	Description
OPT	Amount of file information to list. "normal": Normal file info. "la": Lists extended information.

3.37 List configuration

Command: LISTCFG

Command type: CONFIGURATION

Access: User

Mode: COMMAND

List 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:`LISTCFG``LOADDEFAULT, CONFIG``SETPLAN, SA=35.00, SV=0.00, BURST=1, FN="Data", SO=1``SETBURST, MI=300, DURATION=300, VEL=1, ECHO=0, SPECTRUM=0, ALTI=0, EXTSENS=0, AHRS=1, SENS=0, PROBE=0``SETVEL, SR=8, PL=0.00, DF=8, XMIT=4.00, RECV=8.00, VR=0.30, CY="ENU", NPINGS=2, BW="BROAD"``SETAHRS, SR=0, DF=8, ALG="STATIC"``SAVE, CONFIG``OK`

3.38 Inquire state

Command: INQ**Command type:** INFO**Access:** User**Mode:** COMMAND, CONFIRMATION, RETRIEVAL, MEASUREMENT

The INQ command queries the instrument state. Note that when operating over RS422 serial line, 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

3.39 Read instrument state

Command: READSTATE**Command type:** INFO**Access:** User**Mode:** COMMAND, CONFIRMATION, MEASUREMENT, RETRIEVAL

Returns information about the current operational state of the instrument

Argument	Description
MODE	Current instrument state. 1: Measurement mode (START command received). 2: Command mode. 3: DEPLOY command received and deployment running. 4: Data retrieval mode. 5: Confirmation mode. 8: DEPLOY command received, but deployment has not yet started. 9: Confirmation in measurement mode. 10: Confirmation in deploy mode. 11: Confirmation in pre-deploy mode.
DEPTIME	Seconds since deployment: 0 – DEPLOY command has not been received. < 0 – Number of seconds until deployment starts. > 0 – Number of seconds that deployment has been running. Unit: [s]
MEASTIME	Seconds with measurements: 0 – START command has not been received. > 0 – Number of seconds that measurement has been running. Unit: [s]
CURRTIME	The current instrument time. Time format is "YYYY-MM-DD HH:MM:SS" Unit: [Time]
WAKEUP	Reason for instrument wakeup. 0: Last startup/reboot caused by loss/low voltage. 1: Last startup/reboot caused by power on. 2: Last startup/reboot caused by wakeup by real time clock. 3: Last startup/reboot caused by wakeup by serial activity. 4: Last startup/reboot caused by BREAK command.

	5: Last startup/reboot caused by BREAK timeout. 6: Last startup/reboot caused by watchDog
INTPROC	Internal processing active

Example:

```
READSTATE, WAKEUP, CURRTIME
```

3.40 Read error

Command: READERROR**Command type:** INFO**Access:** User**Mode:** COMMAND, CONFIRMATION, MEASUREMENT, RETRIEVAL

READERROR 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
LIM	Valid limits as text

Example:

```
SETAVG, CS=2.5
```

```
OK
```

```
SAVE, CONFIG
```

```
ERROR
```

```
READERROR
```

```
40, "Invalid setting: Avg Cell Size", "GETAVGLIM, CS=([0.20;2.00]) "
```

```
OK
```

3.41 List all

Command: LISTALL**Command type:** INFO**Access:** User**Mode:** COMMAND

LISTALL 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 LISTTALL.

Argument	Description
----------	-------------

FN

Write the output to this file saved on instrument

Example:**LISTALL**

```

READCLOCKSTR, TIME="2025-04-30 10:23:48"
READID, STR="Vector Current Meter", SN=550199
READHW, DIGITAL="C-1 (Vector) ", ANALOG="C-1 (Vector) ", SOM="1.1 (SOM) "
LISTVER, COMPONENT="FW", STR="0.4.4-
20250430", MAJOR=0, MINOR=4, PATCH=4, HASH="b87f59cc-dirty"
LISTVER, COMPONENT="SW", STR="0.1.0-
internal", MAJOR=0, MINOR=1, PATCH=0, HASH="e5830149"
LISTVER, COMPONENT="PMC", STR="1.14", MAJOR=1, MINOR=14, PATCH=0, HASH="d
3079e0"
LISTVER, COMPONENT="BOOT", STR="0.1.0-
internal", MAJOR=0, MINOR=1, PATCH=0, HASH="gcbd7eb1"
LISTVER, COMPONENT="FPGA", STR="0.44", MAJOR=0, MINOR=44, PATCH=0, HASH="
9a5cbbd"
GETCALPROBE, SDIST=150
GETCALACCLALIGN, M11=1, M12=0, M13=0, M21=0, M22=1, M23=0, M31=0, M32=0, M33
=1
GETCALROTACCL, M11=1.00000, M12=0.00000, M13=0.00000, M21=0.00000, M22=1
.00000, M23=0.00000, M31=0.00000, M32=0.00000, M33=1.00000
GETCALGYROALIGN, M11=1, M12=0, M13=0, M21=0, M22=1, M23=0, M31=0, M32=0, M33
=1
GETCALROTGYRO, M11=1.00000, M12=0.00000, M13=0.00000, M21=0.00000, M22=1
.00000, M23=0.00000, M31=0.00000, M32=0.00000, M33=1.00000
GETCALMAGNALIGN, M11=1, M12=0, M13=0, M21=0, M22=1, M23=0, M31=0, M32=0, M33
=1
GETCALROTMAGN, M11=1.00000, M12=0.00000, M13=0.00000, M21=0.00000, M22=1
.00000, M23=0.00000, M31=0.00000, M32=0.00000, M33=1.00000
GETPRODINSTRUMENT, STR="Vector Current
Meter", SN=550199, NB=3, FREQ=6000000.00, SENSORMASK=31
GETINST, BR=115200, RS="422", LED="ON", ORIENT="AUTOXUPDOWN", CMTOUT=360
0, DMTOUT=60, CFMTOUT=60
GETUSER, POFF=9.50, DECL=0.00, HX=0.0000, HY=0.0000, HZ=0.0000
READSTATE, MODE=2, DEPTIME=0, MEASTIME=0, CURRTIME="2025-04-30
10:23:48", WAKEUP=3, INTPROC=0
READRECSTATUS, VS=15523119104, FS=15254355968, TF=4095, FF=4095
READVEL, SVOL=12.0, VAMBBEAM=0.46
READXFVEL, ROWS=4, COLS=4, M11=1.0000, M12=0.0000, M13=0.0000, M14=0.0000
, M21=0.0000, M22=1.0000, M23=0.0000, M24=0.0000, M31=0.0000, M32=0.0000,
M33=1.0000, M34=0.0000, M41=0.0000, M42=0.0000, M43=0.0000, M44=0.0000
LISTPRODBEAM, BEAM=1, THETA=0.00, PHI=0.00, FREQ=6000000.00, BW=12.50, HW
BEAM=1, ZNOM=1000.00
LISTPRODBEAM, BEAM=2, THETA=0.00, PHI=0.00, FREQ=6000000.00, BW=12.50, HW
BEAM=2, ZNOM=1000.00
LISTPRODBEAM, BEAM=3, THETA=0.00, PHI=0.00, FREQ=6000000.00, BW=12.50, HW
BEAM=3, ZNOM=1000.00
GETPLAN, SA=35.00, SV=0.00, BURST=1, FN="Data", SO=1

```

```
GETBURST,MI=300,DURATION=300,VEL=1,ECHO=0,SPECTRUM=0,ALTI=0,EXTSENS=0,AHRS=1,SENS=0,PROBE=0
GETVEL,SR=8,PL=0.00,DF=8,XMIT=4.00,RCV=8.00,VR=0.30,CY="ENU",NPINGS=2,BW="BROAD"
GETAHRS,SR=0,DF=8,ALG="STATIC"
OK
```

3.42 Get the "Beam to XYZ" transformation matrix

Command: READXFVEL

Command type: INFO

Access: User

Mode: COMMAND

Argument	Description
ROWS	Number of rows
COLS	Number of columns
M11	
M12	
M13	
M14	
M21	
M22	
M23	
M24	
M31	
M32	
M33	
M34	
M41	
M42	
M43	
M44	

3.43 Read hardware specifications

Command: READHW

Command type: INFO

Access: User

Mode: COMMAND, CONFIRMATION, MEASUREMENT, RETRIEVAL

Returns hardware board revisions. Digital and analog board revisions contain the board revision, bill of material (BOM) and type. The SOM uses major.minor for its hardware revision. The devicetree uses board revision and board type.

Argument	Description
DIGITAL	Get digital board revision.
ANALOG	Get analog board revision.
SOM	Get system on module (SOM) board revision.
DEVICETREE	Get devicetree revision.

Example:

READHW

"B-0 (vector2) ", "A-1 (vector2) ", "1.1 (SOM) ", "B (vector2) "

OK

3.44 Read firmware version

Command: READFW

Command type: INFO

Access: User

Mode: COMMAND, CONFIRMATION, MEASUREMENT, RETRIEVAL

This command returns version information for the main firmware in the instrument.

Use LISTVER to get version information for all software components.

Argument	Description
STR	Firmware version string
MAJOR	Major part of the firmware version
MINOR	Minor part of the firmware version
PATCH	Patch part of the firmware version
EXTRA	Extra description of the firmware version
HASH	Unique hash of the firmware source code

Example:

READFW, STR

"1.0.3"

OK

3.45 List versions for all software components

Command: LISTVER

Command type: INFO

Access: User

Mode: COMMAND

The instrument contains multiple software and firmware components. This command lists the version number for each component.

The "FW" component is considered the "main firmware" in the instrument, and is the one that defines the version of the embedded software bundle (the .swu file). This information can also be read using the READFW command, which also contains some additional version information.

Argument	Description
COMPONENT	Software component
STR	Version string for software component
MAJOR	Major part of the component version
MINOR	Minor part of the component version
PATCH	Patch part of the component version
HASH	Unique hash of the component source code

Example:

```
$PNOR, LISTVER*6C
$PNOR, LISTVER, COMPONENT="FW", STR="1.0.3", MAJOR=1, MINOR=0, PATCH=3, HASH="a7f9b64e"*1D
$PNOR, LISTVER, COMPONENT="SW", STR="1.1.2", MAJOR=1, MINOR=1, PATCH=2, HASH="7b11c4e5"*70
$PNOR, LISTVER, COMPONENT="PMC", STR="0.25", MAJOR=0, MINOR=25, PATCH=0, HASH="c89700b8"*38
$PNOR, LISTVER, COMPONENT="BOOT", STR="1.2.0", MAJOR=1, MINOR=2, PATCH=0, HASH="gcbd7eb1"*33
$PNOR, LISTVER, COMPONENT="FPGA", STR="18", MAJOR=18, MINOR=0, PATCH=0, HASH=""*25
$PNOR, OK*2B
```

3.46 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
```

3.47 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
```

3.48 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="WF3CJR6RPNUB",DESC="Current
Profiler",TYPE=1*1C
$PNOR,OK*2B
```

3.49 Read spectrum

Command: READSPECTRUM

Command type: INFO

Access: User

Mode: COMMAND

Reads the spectrum.

Argument	Description
FSTART	Spectrum start frequency.
FSTEP	Spectrum frequency resolution.
NB	Number of beams.
CH	channels used for NARROW, BROAD and ULTRA modes.

3.50 Modbus write

Command: WRITEMODBUS

Command type: CONFIGURATION

Access: User

Mode: COMMAND

Perform a Modbus write transaction (FC16).

This command writes to a single Modbus address per transaction.

All arguments are required and, except for PWROUT, must be unsigned integers in base-10 (decimal) format.

Argument	Description
PWROUT	Modbus output voltage. 0V, 5V, 12V Unit: [V]
ID	Target device modbus ID (0 - 247). Note: ID = 0 is broadcast.
REG	Modbus register. Note: Must be a holding register
VAL	Write value.

Example:

```
WRITEMODBUS, PWROUT="12V", ID=1, REG=2, VAL=10  
OK
```

3.51 Modbus list

Command: LISTMODBUS

Command type: CONFIGURATION

Access: User

Mode: COMMAND

Perform a Modbus read transaction (FC03, FC04) and list register values.

All arguments, except for PWROUT and TYPE, must be unsigned integers in base-10 (decimal) format.

Register values in the output are also represented in base-10.

All arguments must be provided unless explicitly marked as optional.

Argument	Description
PWROUT	Modbus output voltage. 5V, 12V Unit: [V]
ID	Target device modbus ID (1 - 247).
REG	Modbus register.
NWORDS	Number of 16-bit words to read. Optional, defaults to 1.
TYPE	Register type to read. Optional, defaults to "INPUT". "INPUT": Input register "HOLDING": Holding register

Example:

```
LISTMODBUS, PWROUT="12V", ID=1, REG=1  
0001  
OK  
LISTMODBUS, PWROUT="12V", ID=1, REG=1, NWORDS=4  
0001  
0002  
0003  
0004  
OK  
LISTMODBUS, PWROUT="12V", ID=1, REG=1, NWORDS=4, TYPE="HOLDING"  
0010  
0020  
0030
```

0040
OK

4 Binary data formats

This chapter describes the Vector generation 2 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 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 data format will only contain the fields unique for this data format. Take DF3 velocity data as an example:

In short: The data format is the sum of header data, two parts that are shared with other types of data blocks, and the part that is unique for velocity. See 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 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 DF3 velocity data there is also another part 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 DF3 velocity data.

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).

_HeaderData			
_CommonData		DF8 NortekDataformat8	StringData
_DF3 CurrentProfileData			
DF3 VelocityData	DF3 SpectrumData		

Figure: Showing how common data fields (gray for header and blue for other common's) 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.

4.1 `_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 byte	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. 0xC8 - Vector 2 data as DF8. 0xA0 - String Data Record, eg. output from listAll 0x20 - Spectrum data as DF3.
Family id	3 uint8	Defines the instrument family. 0x30 is the latest Aquadopp range of products.
Data size	4 unit16/unit32	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	Checksum of the following Data Record.

	uint16	
Header checksum	8/10 uint16	Checksum of all fields of the Header except the Header Checksum itself.

4.2 NortekDataFormat8

ID: 0xc8

DF8 - Nortek Data Format 8

Field	Position Size	Description
Version	0 uint16	Version number of this Data Record Definition.
Offset of data	2 uint16	Number of bytes from start of record to start of non-common data fields. Unit: [# bytes]
Types of data included	4 4 * 8 bits	Types of data included. Object reference given in table below
Unix time	8 uint32	Seconds since 1970-01-01 00:00:00 UTC. Based on struct timespec
Nano seconds	12 uint32	Nano seconds from last whole second. Based on struct timespec
Ensemble counter	16 uint32	Ensemble counter.
Water temperature	20 float	Reading from the temperature sensor. Unit: [°C]
Speed of sound	24 float	Configured or measured sound velocity. Unit: [m/s]
Battery voltage	28 float	Battery voltage
Current drain	32 float	Unit: [Ampere]
Dew point temperature	36 float	Unit: [°C]
CPU temperature	40 float	Unit: [°C]
FPGA temperature	44 float	Unit: [°C]
RTC temperature	48 float	Unit: [°C]
Status	52	Status bit mask.

	4 * 8 bits	Object reference given in table below
Error TBD	56 4 * 8 bits	Error bit mask. TDB- DUMMY MASK Object reference given in table below
reserved	60 float * 5	5 reserved floats
Velocity data	OFFSET VEL_SIZE bytes	This object exists if the Velocity data included bit of the config byte is set. Object reference given in table below
Pressure data	OFFSET + VEL_SIZE PRESS_SIZE bytes	This object exists if the pressure data included bit of the config byte is set. Object reference given in table below
Euler angles data	OFFSET + VEL_SIZE + PRESS_SIZE EULER_SIZE bytes	This object exists if the Euler angles data included bit of the config byte is set. Object reference given in table below
Direction Cosine Matrix data	OFFSET + VEL_SIZE + PRESS_SIZE + EULER_SIZE float * DIM1 SIZE * DIM2 SIZE	This object exists if the Direction Cosine Matrix data included bit of the config byte is set. dcm00, dcm01, dcm02, dcm10...dcm22,
Quaternion data	OFFSET + VEL_SIZE + PRESS_SIZE + EULER_SIZE + DCM_SIZE QUAT_SIZE bytes	This object exists if the Quaternion data included bit of the config byte is set. Object reference given in table below
Raw Compass data	OFFSET + VEL_SIZE + PRESS_SIZE + EULER_SIZE + DCM_SIZE + QUAT_SIZE COMPASS_SIZE bytes	This object exists if the raw Compass data included bit of the config byte is set. Object reference given in table below
Raw AHRS data	OFFSET + VEL_SIZE + PRESS_SIZE + EULER_SIZE + DCM_SIZE + QUAT_SIZE + COMPASS_SIZE AHRS_SIZE bytes	This object exists if the raw AHRS data included bit of the config byte is set. Object reference given in table below
PyroScience AquapHOx-TX sensor data	OFFSET + VEL_SIZE + PRESS_SIZE + EULER_SIZE + DCM_SIZE + QUAT_SIZE + COMPASS_SIZE + AHRS_SIZE PYROSCIENCE_AQUAP HOX_SIZE bytes	This object exists if the PyroScience AquapHOx-TX included bit of the config byte is set. Object reference given in table below

Probe check data	OFFSET + VEL_SIZE + PRESS_SIZE + EULER_SIZE + DCM_SIZE + QUAT_SIZE + COMPASS_SIZE + AHRS_SIZE + PYROSCIENCE_AQUAP HOX_SIZE PROBE_SIZE bytes	This object exists if the probe check data included bit of the config byte is set. Object reference given in table below
Echosounder data	OFFSET + VEL_SIZE + PRESS_SIZE + EULER_SIZE + DCM_SIZE + QUAT_SIZE + COMPASS_SIZE + AHRS_SIZE + PYROSCIENCE_AQUAP HOX_SIZE + PROBE_SIZE ECHO_SIZE bytes	This object exists if the echosounder data included bit of the config byte is set. Object reference given in table below
RINKO-EC sensor data	OFFSET + VEL_SIZE + PRESS_SIZE + EULER_SIZE + DCM_SIZE + QUAT_SIZE + COMPASS_SIZE + AHRS_SIZE + PYROSCIENCE_AQUAP HOX_SIZE + PROBE_SIZE RINKO_EC_SIZE bytes	This object exists if the RINKO-EC included bit of the config byte is set. Object reference given in table below
Seapoint STM-S sensor data	OFFSET + VEL_SIZE + PRESS_SIZE + EULER_SIZE + DCM_SIZE + QUAT_SIZE + COMPASS_SIZE + AHRS_SIZE + PYROSCIENCE_AQUAP HOX_SIZE + PROBE_SIZE + RINKO_EC_SIZE SEAPPOINT_STM_S_SIZE bytes	This object exists if the Seapoint STM-S included bit of the config byte is set. Object reference given in table below

Position and size variables:

Name	Description
------	-------------

VEL_NB	Primary dimension of velocity data is number of beams. Length 0 if velocity data in includedData bit map is false.
VEL_NB	Primary dimension of amplitude data is number of beams. Length 0 if velocity data in includedData bit map is false.
CORR_NB	Primary dimension of correlation data is number of beams. Length 0 if velocity data in includedData bit map is false.
VEL_SIZE	Size of velocity data. 28 bytes if the Velocity data included bit of the config byte is set. Else 0.
OFFSET	Number of bytes from start of record to start of data.
PRESS_SIZE	Size of pressure data. 8 bytes if the pressure data included bit of the config byte is set. Else 0.
PYROSCIENCE_AQUAPHOX_SIZE	Size of PyroScience AquaphOX-TX data. 16 bytes if the included bit of the config byte is set. Else 0.
NB	Primary dimension of amplitude data is 3 which is the number beams.
PROBE_NS	Second dimension of amplitude data is number of samples pr beam.
PROBE_SIZE	Size of probe check data. If the Probe check data included bit of the included byte is set, the size is $10 + 2 * \text{number of beams} * \text{number of samples pr beam}$. Else 0.
CB	Primary dimension of amplitude data is 1 which is the center beam.
ECHO_NS	Second dimension of amplitude data is number of samples pr beam.
ECHO_SIZE	Size of echosounder data. If the Echosounder data included bit of the included byte is set, the size is $14 + 2 * \text{number of samples pr beam}$. Else 0.
RINKO_EC_SIZE	Size of RINKO-EC data. 12 bytes if the included bit of the config byte is set. Else 0.
SEAPOINT_STM_S_SIZE	Size of Seapoint STM-S data. 12 bytes if the included bit of the config byte is set. Else 0.

Object reference: Types of data included

Types of data included.

Field	Position Size	Description
Velocity included	0 bit	Velocity, correlation and amplitude data is included
Pressure included	1 bit	Pressure data is included
Euler angles included	2 bit	Euler angles are included

Direction cosine matrix included	3 bit	Direction cosine matrix is included
Quaternions included	4 bit	Quaternions are included
Raw compass data included	5 bit	Raw compass data is included
Raw AHRS data included	6 bit	Raw AHRS data is included
PyroScience AquapHOx-TX included	7 bit	PyroScience AquapHOx-TX data is included
Probe check included	8 bit	Probe check data is included
Echo sounder included	9 bit	Echo sounder data is included
RINKO-EC included	10 bit	RINKO-EC data is included
Seapoint STM-S included	11 bit	Seapoint STM-S data is included

Object reference: Status

Status bit mask.

Field	Position Size	Description
Wake up state	0-3 4 bits	Bit 0-3: Wakeup State 0000 = Bad Power 0001 = Power Applied 0010 = RTC Alarm 0011 = Wakeup RS422 0100 = BREAK 0101 = BREAK (timeout) 0110 = WatchDog
Previous wakeup state	4-7 4 bits	Bit 4-7: Previous Wakeup State 0000 = Bad Power 0001 = Power Applied 0010 = RTC Alarm 0011 = Wakeup RS422 0100 = BREAK 0101 = BREAK (timeout) 0110 = WatchDog

Coordinate system	8-9 2 bits	Bit 8-9: Coordinate system 00 = ENU (East North Up) 01 = XYZ 10 = BEAM 11 = na
Number of beams	10-12 3 bits	Bit 10-12: Number of beams
Last Measure Voltage Skip	13 bit	Last measure voltage skip
Active Configuration	14 bit	True if active configuration

Object reference: Error TBD

Error bit mask. TDB- DUMMY 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	Measurement not finished The Measurement and data storage/transmit didn't finish before next measurement started.
Sensor read failure	5 bit	Sensor read failure
FPGA clock PLL Lock failure	6 bit	FPGA clock PLL Lock 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)
Error Reading Pressure Sensor	16 bit	Error Reading Pressure Sensor
Error Reading Water temperature Sensor	17 bit	Error Reading Water temperature Sensor
Error Reading Battery Sensor	18 bit	Error Reading Battery Sensor
Error Reading Humidity Sensor	19 bit	Error Reading Humidity Sensor
Error Reading IMU	20 bit	Error Reading IMU
Error Reading Magnetometer	21 bit	Error Reading Magnetometer

Object reference: Velocity data

This object exists if the Velocity data included bit of the config byte is set.

Field	Position Size	Description
Ambiguity velocity	0 float	Ambiguity velocity Required bit: Velocity included Unit: [m/s]
Velocity data	4 float * VEL_NB * 1	Velocity data Required bit: Velocity included Unit: [m/s]
Amplitude data	16 uint16 * VEL_NB * 1	Amplitude data, raw data given as 0.01 % Required bit: Velocity included Unit: [dB]
Correlation data	22 uint16 * CORR_NB * 1	Correlation data [0 - 100 %], raw data given as 0.01 % Required bit: Velocity included Unit: [%]

Object reference: Pressure data

This object exists if the pressure data included bit of the config byte is set.

Field	Position Size	Description
Pressure	0 float	Pressure measurement Required bit: Pressure included Unit: [Bar]
Pressure sensor temperature	4 float	Temperature measured by the pressure sensor Required bit: Pressure included Unit: [°C]

Object reference: Euler angles data

This object exists if the Euler angles data included bit of the config byte is set.

Field	Position Size	Description
Heading	0 float	Heading angle. Unit: [deg]
Pitch	4 float	Pitch angle. Unit: [deg]
Roll	8 float	Roll angle. Unit: [deg]

Object reference: Quaternion data

This object exists if the Quaternion data included bit of the config byte is set.

Field	Position Size	Description
W	0 float	W component of the quaternion
X	4 float	X component of the quaternion
Y	8 float	Y component of the quaternion
Z	12 float	Z component of the quaternion

Object reference: Raw Compass data

This object exists if the raw Compass data included bit of the config byte is set.

Field	Position Size	Description
Magnetometer.X	0 float	Raw X axis value in last measurement interval
Magnetometer.Y	4 float	Raw Y axis value in last measurement interval
Magnetometer.Z	8 float	Raw Z axis value in last measurement interval
Accelerometer.X	12 float	Raw X axis value in last measurement interval
Accelerometer.Y	16 float	Raw Y axis value in last measurement interval
Accelerometer.Z	20 float	Raw Z axis value in last measurement interval

Object reference: Raw AHRS data

This object exists if the raw AHRS data included bit of the config byte is set.

Field	Position Size	Description
Gyro.X	0 float	Raw X axis value in last measurement interval
Gyro.Y	4 float	Raw Y axis value in last measurement interval
Gyro.Z	8 float	Raw Z axis value in last measurement interval
AHRS Figure of merit	12 float	Figure of merit describing data quality.
IMU temperature	16 float	Temperature measured by the IMU sensor. Unit: [°C]

Object reference: PyroScience AquapHOx-TX sensor data

This object exists if the PyroScience AquapHOx-TX included bit of the config byte is set.

Field	Position Size	Description
Status	0 int16	ERRNO driver status code from fetching the sample. A value of 0 indicates that everything is OK.
Validity	2 2 * 8 bits	Validity bit mask of the sensor values. Object reference given in table below
Dissolved oxygen concentration	4 float	Dissolved oxygen concentration given in umolar Unit: [umolar]
Oxygen volume fraction	8 float	Oxygen volume fraction given as a percentage Unit: [percentage]
Sample temperature	12 float	Sample temperature given in Celsius Unit: [°C]

Object reference: Probe check data

This object exists if the probe check data included bit of the config byte is set.

Field	Position Size	Description
Blanking	0 float	Blanking length Required bit: Probe check included Unit: [m]
Sample distance	4 float	Sample distance. Required bit: Probe check included Unit: [m]
Number of samples	8 uint16	Number of samples per beam. Required bit: Probe check included
Probe check amplitude data	10 int16 * NB * PROBE_NS	Probe check amplitude data. This field exists if the Probe check data included bit of the included byte is set Required bit: Probe check included Unit: [dB]

Object reference: Echosounder data

This object exists if the echosounder data included bit of the config byte is set.

Field	Position Size	Description
Blanking	0 float	Blanking length Required bit: Echosounder included Unit: [m]
Binsize	4 float	Binsize. Required bit: Echosounder included Unit: [m]

Frequency	8 float	Frequency. Required bit: Echosounder included Unit: [Hz]
Number of samples	12 uint16	Number of samples per beam. Only the center beam is used. Required bit: Echosounder included
Echosounder amplitude data	14 int16 * CB * ECHO_NS	Echosounder amplitude data. This field exists if the Echosounder data included bit of the included byte is set Required bit: Echosounder included Unit: [dB]

Object reference: RINKO-EC sensor data

This object exists if the RINKO-EC included bit of the config byte is set.

Field	Position Size	Description
Status	0 int16	ERRNO driver status code from fetching the sample. A value of 0 indicates that everything is OK.
Validity	2 2 * 8 bits	Validity bit mask of the sensor values. Object reference given in table below
Dissolved oxygen concentration	4 float	Dissolved oxygen concentration given in umolar Unit: [umolar]
Sample temperature	8 float	Sample temperature given in Celsius Unit: [°C]

Object reference: Seapoint STM-S sensor data

This object exists if the Seapoint STM-S included bit of the config byte is set.

Field	Position Size	Description
Status	0 int16	ERRNO driver status code from fetching the sample. A value of 0 indicates that everything is OK.
Validity	2 2 * 8 bits	Validity bit mask of the sensor values. Object reference given in table below
S signal	4 float	S signal given in FTU (Formazin Turbidity Units) Unit: [FTU]
R signal	8 float	R signal given in NTU (Nephelometric Turbidity Units) Unit: [NTU]

Object reference: Validity

Validity bit mask of the sensor values.

Field	Position Size	Description
Dissolved oxygen concentration validity	0 bit	True if it is is valid
Oxygen volume fraction validity	1 bit	True if it is is valid
Sample temperature validity	2 bit	True if it is is valid

Object reference: Validity

Validity bit mask of the sensor values.

Field	Position Size	Description
Dissolved oxygen concentration validity	0 bit	True if it is is valid
Sample temperature validity	2 bit	True if it is is valid

Object reference: Validity

Validity bit mask of the sensor values.

Field	Position Size	Description
S signal validity	0 bit	True if it is is valid
R signal validity	1 bit	True if it is is valid

4.3 _CommonData

Used By: _DF3 CurrentProfileData

Common data definitions for parsing Nortek data formats.

Field	Position Size	Description
Version	0 uint8	Version number of the Data Record Definition.
Offset of data	1 uint8	Number of bytes from start of the record to start of the actual data. Unit: [# bytes]
Serial number	4 uint32	Instrument serial number from factory.
Year	8 uint8	Number of years since 1900.
Month	9 uint8	Month number counting from 0 which is January.
Day	10 uint8	Day of the month
Hour	11 uint8	24 hour of the day
Minutes	12 uint8	Minutes.
Seconds	13 uint8	Seconds.
Hundred micro seconds	14 uint16	Hundred micro seconds from last whole second. Unit: [100 μ s]
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	Raw data given as 0.01 degrees

	int16	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 int16	Y axis flux raw value in last measurement interval
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 2 * 8 bits	Error 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 whether the ambiguity velocity (in DF3 CurrentProfileData) is 16bit or 32bit long.
56/58	The status field is at 56 or 58 depending on whether the ambiguity velocity (in DF3 CurrentProfileData) is 16bit or 32bit long.
58/60	The status field is at 56 or 58 depending on whether the ambiguity velocity (in DF3 CurrentProfileData) is 16bit or 32bit long.
59/61	The status field is at 59 or 61 depending on whether the ambiguity velocity (in DF3 CurrentProfileData) is 16bit or 32bit long.
60/62	The status field is at 60 or 62 depending on whether the ambiguity velocity (in DF3 CurrentProfileData) is 16bit or 32bit long.
62/64	The status field is at 62 or 64 depending on whether the ambiguity velocity (in DF3 CurrentProfileData) is 16bit or 32bit long.
64/66	The status field is at 64 or 66 depending on whether the ambiguity velocity (in DF3 CurrentProfileData) is 16bit or 32bit long.
72/74	The status field is at 72 or 74 depending on whether the ambiguity velocity (in DF3 CurrentProfileData) is 16bit or 32bit long.

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)

4.4 _DF3 CurrentProfileData

Extends: _CommonData

Used By: DF3 SpectrumData

Common data definitions for Nortek data format 3 (DF3).

Field	Position Size	Description
Configuration bit mask	2 2 * 8 bits	Record Configuration Bit Mask Object reference given in table below
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]
Temperature PressureSensor	37 uint8	Temperature of pressure sensor: $T=(Val/5)-4.0$ Raw value given as 0.2 °C Unit: [°C]
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 2 * 8 bits	Extended status bit mask Object reference given in table below
Status	68 4 * 8 bits	Status bit mask. Note that bits 0, 2, 3, 4 are unused. Object reference given in table below

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
Has external sensor	4 bit	External sensor value valid
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 echosounder data	11 bit	Echosounder 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
Has spectrum data	15 bit	Amplitude spectrum data included.

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%	2 bit	Indicates that the processor idles less than 12 percent
External sound velocity probe	3 bit	Sound velocity probe velocity received
External heading, pitch, roll, and position	4 bit	External heading, pitch, roll and position received from NMEA NTKNAV
External heading	5 bit	External heading received from NMEA input
External pitch and roll	6 bit	External pitch and roll received from NMEA input
Ethernet powered system	7 bit	Indicates if the system is powered via Ethernet
File system flush	13 bit	File system flush in progress
Internal processing	14 bit	Internal processing in progress (e.g. wave processing)

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
Blanking distance scaling in cm	1 bit	Bit 1: Scaling of blanking distance 0: mm scaling 1: given in cm
Echosounder frequency bin	5-9 5 bit	Used only on the Signature100 which supports up to 5 packages pr frequency index. Processing is then done on the different frequency bands.
Boost running	10 bit	Boost running
Telemetry data	11 bit	Telemetry data
Echosounder index	12-15 4 bits	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.
Active configuration	16 bit	Bit 16: Active configuration 0: Settings for PLAN,BURST,AVG 1: Settings for PLAN1,BURST1,AVG1
Previous measurement skipped due to low voltage	17 bit	Bit 17: Last measurement low voltage skip 0: normal operation 1: last measurement skipped due to low input voltage
Previous wakeup state	18-21 4 bits	0 - Bad power 1 - Power on 2 - Break 3 - RTC 4 - Watchdog 5 - Low voltage 6 - Filesystem error
Auto orientation	22-24 3 bits	0: "Fixed" Fixed orientation 1: "Auto" Auto Up Down 3: "AHRS3D" AHRS3D
Orientation	25-27 3 bits	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.
Wake up state	28-31 4 bits	0 - Bad power 1 - Power on 2 - Break 3 - RTC 4 - Watchdog 5 - Low voltage 6 - Filesystem error

4.5 DF3 SpectrumData

Extends: _DF3 CurrentProfileData

ID: 0x20

Data definitions for parsing DF3 amplitude spectrum data.

Field	Position Size	Description
Beams and bins	30 2 bytes	Number of bins in the frequency spectrum. Object reference given in table below
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: Beams and bins

Number of bins in the frequency spectrum.

Field	Position Size	Description
Number of beams	15-13 3 bits	Number of active beams.
Number of bins	12-0 13 bits	Number of bins.

4.6 StringData

ID: 0xa0

String Data Record, eg. GPS NMEA data, comment from the FWRITE command.

Field	Position Size	Description
String	0 Size of record bytes	String data record.

5 Telemetry Data Formats

The telemetry of the average current data is controlled by the **SET/GETTMAVG** command. The DF parameter of this command sets the data format, this can also be set in the deployment software.

Data format (DF)	Description
------------------	-------------

700

Legacy binary data format, same as used by the previous generation of the Vector current meter. Described in the [Integrators Guide - Classic](#)

Table of Figures

Figure 1: Instrument modes of operation	7
---	---