NORTEK MANUALS

Integrator's Guide

Signature

NORTEK

Table of Contents

1 Introduction

The primary objective of this manual is to provide the information needed to control a Nortek product that is based on the AD2CP hardware platform. This includes all instruments in the Signature series. It is aimed at system integrators and engineers with interfacing experience, but it also includes examples on how to configure and start the instrument for more inexperienced integrators. The document's scope is limited to interfacing and does not address general performance issues of the instrument.

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

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 2015.1	2015	First version
Version 2015.2	20.10.2015	
Version 2016.1	01.03.2016	
Version 2016.2	14.03.2016	Minor updates
Version 2016.3	30.09.2016	
Version 2017.1	10.02.2017	
Version 2017.2	20.10.2017	
Version 2017.3	06.12.2017	Minor updates
Version 2018.1	03.10.2018	
Version 2021.1	27.01.2021	Major updates

Version 2021.2	17.09.2021	Major updates.
Version 2022.1	11.01.2022	Minor updates to command layouts
Version 2022.2	31.03.2022	Additions to data format chapters
Version 2022.3	23.09.2022	Data format clarification and minor updates
Version 2023.1	23.03.2023	Major data format updates and additions. Internal wave processing included. Updated contact information.
Version 2024.1	18.01.2024	Minor updates.
Version 2025.1	30.04.2025	Major review of the commands and binary data formats chapter. Description of Status and Extended status for DF3 is gathered and elaborated.
Version 2025.2	23.07.2025	Interfaces chapter updated. Data Quality chapter added. Minor updates to the command chapter.
Version 2026.1	05.03.2026	Minor updates to Interfaces, Data formats and Commands chapter
Version 2026.2	29.04.2026	Links updated

2 Basic interface concept

This chapter covers the operational modes and how to switch between them. These modes are frequently used during instrument communication, so understanding their role is important before proceeding to interfaces and commands.

2.1 Modes

The Signature instrument operates in several distinct modes, each associated with specific commands for controlling its behavior. For details on which commands are available in each mode, refer to the List of Commands section. Most commands are issued while the instrument is in Command mode.

The available operating modes are:

Command - Used for general commands and control
Data Retrieval - Used to temporarily stop a deployment and downloading data from the recorder
Measurement - Active data collection
Confirmation - Used to confirm transition between modes

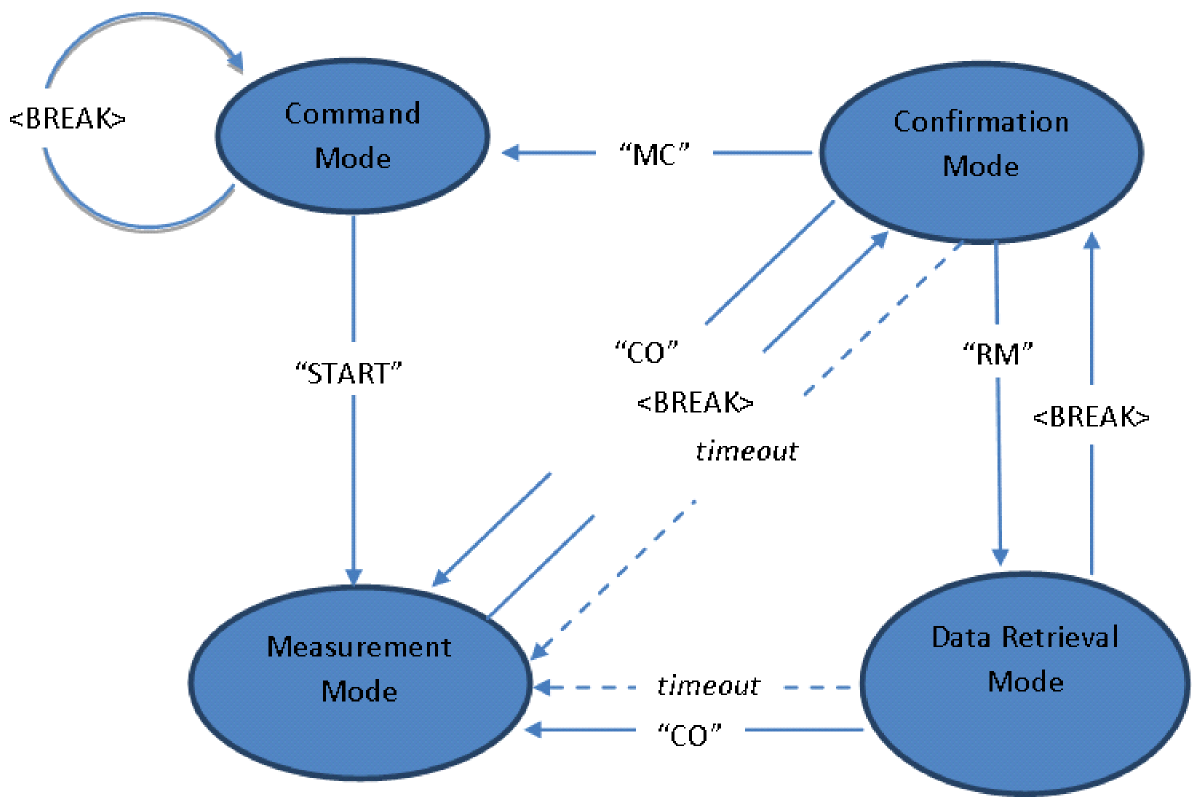


Figure: Instrument modes of operation

Communication with the instrument is initialized by sending a **<BREAK>** signal, as defined in the next section. Depending on the current mode, the **<BREAK>** either places the instrument into Confirmation Mode or restarts Command Mode. The available options for switching modes depend on the instrument's current mode (see the diagram above for details). If no commands are received within a given mode, a timeout occurs, prompting the instrument to resume operation automatically.

Default timeout durations are as follows:

- Confirmation Mode: 60 seconds
- Data Retrieval Mode: 60 seconds
- Command Mode (Serial interface only): 5 minutes

In Command Mode when operating over the serial interface, if no commands are received for 5 minutes, the processor must be reawakened. This can be done by sending either a **<BREAK>** or a sequence of @@@@@@.

All the time out durations listed above can be altered using the **SETINST** command.

2.2 Break

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

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

A sequence of **@@@@@@** characters is used to wake the processor from sleep mode, since the instrument is not able to monitor serial line activity while asleep. A second sequence of actual break characters is included to ensure that a break is reliably detected—even if the instrument wakes up due to another cause, such as a real-time clock alarm. This ensures that the processor correctly interprets the command that follows.

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

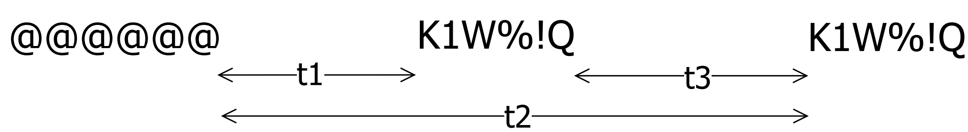


Figure: Break timing

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

2.3 Dual Processor

The Signature instrument features a dual-processor (DSP) design: one processor is dedicated to Doppler processing (BBP), while the other handles interface operations (SEC). The primary interface is Ethernet, meaning the interface processor is only powered when external power is supplied through the Ethernet cable. Supplying power via Ethernet also energizes the remaining electronic components. Because Ethernet is the primary interface, it takes priority - if both Ethernet and the Serial interface are powered simultaneously, communication and data output will occur exclusively over Ethernet.

3 Interfaces

This chapter provides an overview of the available communication methods for the Signature instrument. By default, the instrument supports Ethernet communication, which serves as the primary interface. Ethernet communication is handled through the 6-pin connector. In addition, serial communication is available if the instrument is equipped with an endbell featuring an 8-pin connector, which can be selected during ordering or obtained later as a spare part.

Note: The Ethernet cable must be powered for the interface to function. When powered, the Ethernet connection takes priority over the serial connection, meaning the serial interface will be unavailable while Ethernet is active.

The command structure and data output formats are consistent across both Ethernet and serial communication. In most cases, when the option "Serial output" is enabled, the instrument transmits data over both interfaces simultaneously. You can use either interface to receive the data and issue commands, there is no need to explicitly configure the preferred interface.

3.1 Serial Operation

The Signature uses the serial protocol configuration of eight data bits, no parity, and one stop bit (8N1).

If the Signature instrument has been ordered with, or later upgraded to include, an 8-pin serial connector, it can communicate over a serial interface. The instrument supports both RS232 and RS422 protocols. The serial cable provided with the instrument is compatible with both protocols, and you can switch between them using the SETINST command. To change between RS232 and RS422, you must connect via Ethernet. For step-by-step instructions, refer to the FAQ: How can I change the communication protocol from RS232 to RS422 on my Signature?

Serial communication is command-line based only; the graphical user interface (GUI) in the Signature Deployment software cannot be used over a serial connection. If you connect via serial using Signature Deployment, a terminal window will open. You can also use any other terminal program that supports standard serial communication.

3.2 Ethernet Operation

The Signature instrument uses TCP (Transmission Control Protocol) for both command processing and data transmission. In TCP/IP networks, each communication channel is uniquely identified by a combination of the IP address and port number. On the Signature, different ports are used for different types of communication:

- **Port 9000** is a Telnet-style 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 different ports are described in more detail in the following chapters.

If password protection is disabled via the web interface, the system will accept any input, including an empty password.

When the instrument is in Measurement Mode, any command sent should be preceded by the BBPWAKEUP command. This ensures that the BBP (Baseband Processor) is active and ready to process incoming instructions. If more than 2 seconds have passed since the previous command, another BBPWAKEUP must be issued. If you're unsure of the current instrument mode, it's good practice to send BBPWAKEUP before commands like GETSTATE or INQ. For an example, refer to the section Checking instrument state over Ethernet.

3.2.1 Telnet Connection (Port 9000)

The Telnet interface, accessible via TCP/IP port 9000, is used for direct user interaction with the instrument. This dedicated port accepts ASCII-based commands and returns human-readable responses. The list of supported commands is provided in the Commands section.

To connect using the Windows Telnet client, enter the following command in the terminal:

```
telnet <ip_address> 9000
```

You will be prompted for a username and password, these are by default:

```
Username: nortek
Password: (leave blank - password protection is disabled by default)
```

Password protection can be enabled through the instrument’s web interface.

When logging in you will see:

```
Nortek Signature Command Interface
```

This interface is very similar to the direct serial interface (RS232/RS422), with some additions to simplify interaction. Most notably, you can send a <BREAK> command to the Doppler DSP by pressing Ctrl-C (ASCII 0x03). The internal application handles waking up the Doppler processor and ensures proper timing of the break signal.

The Telnet server does not echo typed characters by default. If you wish to see or edit commands before sending them, you must enable local echo and local line editing in your Telnet client. A client such as PuTTY supports these features.

Port 9000 is dedicated solely to ASCII communication. Do not use a Telnet client to connect to other instrument ports that transmit binary data (see next section for details).

To terminate the Telnet session, press Ctrl-X (ASCII 0x18).

3.2.2 Raw Connections (Ports 9001, 9002, and 9004)

A port can be understood as a address point between two communicating parts. When first connecting to a data listening port, the following identification string is sent by the instrument:

```
\r\nNortek <name> Data Interface\r\n
```

Here, <name> is replaced by the instrument's host name, allowing clients to verify which device responded.

The Signature instrument uses the following TCP ports for data streaming and control:

Port 9001 - Binary data and command interface

- Used for machine-driven communication and control
- Requires username/password authentication
- Provides access to binary data generated in Measurement Mode
- Translates serial port data directly into TCP/IP over Ethernet, and the commands and data formats are identical to those used over the serial port.
- The instrument handles correct internal timing when breaks are sent. Break signal can be sent in two ways:
  - Sending the string **K1W%!Q<CR><LF>** as described in the Break chapter
  - Send **Ctrl-C (**ASCII 0x03) as a standalone character (not embedded in a command)
- Standard streaming record delineation techniques must be used to ensure the received data is correctly synchronized for decoding

Port 9002 - Data Output Channel

- Data-only channel with no command input
- Can be used by real-time display applications while configuration is handled by a separate system.

Port 9004 - ASCII Data Output Channel

- Outputs ASCII-formatted telemetry data with no command input
- Only active when the instrument is configured to generate telemetry output in ASCII format

A Telnet client should **not** be used to access ports 9001, 9002, or 9004. Telnet introduces its own control sequences and binary interpretation rules, which are not compatible with the raw data these ports provide. Using a Telnet client on these ports will result in unintended control characters being sent and binary characters being misinterpreted.

3.2.3 FTP

The Signature instrument’s internal data recorder can be accessed over Ethernet using any standard FTP (File Transfer Protocol) client. When combined with telemetry options, FTP offers a convenient method for downloading measured data at regular intervals, especially in applications where true real-time data access is not required. Only the telemetry file (`telemetryfile.bin`) can be deleted via FTP. Other data files are read-only.

When an FTP connection is active, the internal state of the instrument is changed so that commands are no longer processed. Any commands sent during an active FTP session will return an error. If a break is sent during an active FTP transfer, the FTP connection will be forcibly terminated.

If an FTP connection is done while the instrument is in Measurement Mode (see Modes diagram), the FTP connection is made through Data Retrieval Mode. Upon terminating the FTP session, the instrument returns to its previous mode. An FTP session will terminate automatically if no FTP commands or data transfers occur for 120 seconds.

The telemetry option implemented in the Doppler processor enables system integrators to regularly offload subsets of the data by using FTP. When the network processor receives an incoming FTP request, it will interrupt the Doppler processor by entering Data Retrieval Mode and mount the file system of the recorder. The telemetry file can be deleted after it has been downloaded, which is particularly suitable for event driven data downloads. If the instrument was started using the DEPLOY command, it will resynchronize to its measurement time base once the FTP session ends. For a step-by-step example of how to configure telemetry output and retrieve files via FTP, refer to the User Cases section.

3.2.4 HTTP

The Signature instrument includes a built-in HTTP interface for downloading individual data files. This feature is accessible from the instrument's web page by clicking the "Data Download (HTTP)" link on the main page. HTTP download is useful in environments where firewall or security settings restrict the use of FTP.

3.2.5 UDP

The Signature instrument also supports UDP (User Datagram Protocol) for real-time data transmission. Unlike TCP, UDP does not require a connection to be established; the data collection software simply waits for incoming data from the instrument.

UDP is useful in scenarios where instrument power may be intermittently interrupted, as the data stream resumes automatically after a power cycle. However, UDP does not guarantee delivery—on unreliable or noisy networks, some datagrams may be lost. If reliable delivery is critical, TCP is recommended.

To enable UDP in a power-resilient configuration use the instrument's web interface to configure the IP address and port of the data collection software. The IP address specifies the target client; the port (range: 9000–9500) can be used to distinguish instruments. The same port may be reused across instruments if the software uses the source IP address to identify them. Once configured, the instrument's Ethernet processor will begin automatically transmitting real-time data to the specified address/port. After a power cycle, an instrument originally in Measurement Mode re-enters Measurement Mode and resumes transmission without needing to re-establish communication.

3.2.6 PTP/NTP

The Signature instrument supports time synchronization via PTP (Precision Time Protocol) and NTP (Network Time Protocol), both of which can be enabled through the instrument's web interface.

Precision Time Protocol (PTP – IEEE 1588)

PTP enables distribution of high-resolution absolute time across an Ethernet network. The Signature can be configured to operate as a PTP slave, synchronized to a customer-supplied PTP master clock located on the same 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. Using PTP does not affect the choice of UDP or TCP for the transport of data.

Network Time Proyocol (NTP)

NTP provides time synchronization accuracy of approximately ±1 ms on a local area network (LAN) and ±10 ms across a wide area network (WAN).

3.3 Software

The primary way to communicate with your Signature instrument is through the instrument software called Signature Deployment. This software is free of charge and can be accessed via our software web page or downloaded directly from the Microsoft Store. Please note that the graphical user interface (GUI) of the software is only accessible when connecting via Ethernet. If you connect using serial communication, the built-in terminal will open instead.

The Signature Deployment software offers the quickest and easiest way to perform tasks such as:

1. Configuring and starting a new deployment
2. Stopping a deployment and downloading data from the recorder
3. Performing a function test of the instrument
4. Checking for and updating the instrument firmware

Most standard deployment setups can be completed using the Deployment Wizard included in the software. For more advanced configurations, beyond what the wizard allows, you can customize your setup using the commands described in the Commands chapter. For guidance, please refer to the FAQ: How do I change my Signature configuration (.deploy file) outside of the deployment software?. We recommend using the wizard whenever possible, as it covers most common cases. If you do need to add custom commands, start by creating a .deploy file that is as close to your desired configuration as possible. Then, use the Customize option and only add the commands that are necessary. There is also a built-in terminal in the software which can be used to communicate directly with the instrument by commands.

3.4 Web Browser

When connected over Ethernet, you can access the instrument web page through your web browser. The site can be accessed directly by typing in the IP address or through the Signature Deployment software for example by right clicking on the instrument on the discovery page.

The instrument web page allows you to for example

1. Check for updates and update firmware
2. Set your desired Network Configuration, for example enabling UDP data transmission
3. Retrieve support file
4. Enable password protection
5. Download data

3.5 Command interface

The command interface enables direct communication with the instrument via either a serial or Ethernet connection. Commands are described in the Commands chapter and can be used both to interact with the instrument through a terminal and to customize the .deploy file used by the Signature Deployment software. The .deploy file is command-based and can be read directly into the command interface.

Each command accepts a defined set of arguments, which are detailed in the relevant chapters. Some commands require at least one argument in order to execute properly.

The command interface is ASCII-based and line-oriented. Each command must be terminated with a carriage return and line feed (CR/LF). For enhanced data integrity, commands may also be used in NMEA format, which includes a prefix and checksum. When using NMEA-style commands, the instrument will return argument names in the response.

3.5.1 Error messages and validation

Comprehensive validation and error handling are built into the command interface. When you send the **SAVE** command, the current configuration is immediately verified. If you skip **SAVE**, the deployment plan will instead be validated when issuing the **START** or **DEPLOY** command.

If the deployment plan contains invalid elements, such as parameters with values outside their permitted range, an **ERROR** will be returned. To diagnose the issue, use the **GETERROR** command. This will return a detailed error message, including the parameter and its valid range. Each validation error must be corrected before proceeding, and additional errors may be revealed as each one is resolved. Below is an example of how to use the **GETERROR** command.

```
% Setting the measurement interval to 10000 seconds on a Signature 1000 using the SETPLAN
command and receiving an OK
SETPLAN,MIAVG=10000
$PNOR,OK*2B

% Saving the configuration and receiving an ERROR
SAVE,CONFIG
$PNOR,ERROR*77

% Following up with the GETERROR command to investigate
GETERROR
$PNOR,GETERROR,NUM=227,STR="Invalid setting: Plan Profile
Interval",LIM="GETPLANLIM,MIAVG=([1;7200])"*56
```

The response to the **GETERROR** command indicates that the measurement interval must be within 1:7200 seconds and you have to go back and reconfigure this before you can proceed. Note that the limits will change depending on the rest of your configuration

3.5.2 Limits

The valid range for the various arguments can be retrieved by using the **GETxxxLIM** commands where **xxx** is the command you want the limits for. For example, if you need the limits for all arguments associated with the **SETAVG** command, you send the **GETAVGLIM** command. You can also retrieve the limits for one specific argument by including this when sending the **GETxxxLIM** command. As in the example above when you are configuring the measurement interval for the average measurements, you can get the valid range beforehand by sending the command **GETPLANLIM,MIAVG**. The same goes for arguments associated with all types of applications, the valid ranges for burst measurements can be retrieved by the command **GETBURSTLIM** etc**.**

3.5.3 Starting the measurements

There are two commands that can be used to start the instrument: **DEPLOY** and **START**. While both initiate measurement, they differ significantly in how the instrument behaves after transitions between modes.

When the instrument is started using the **START** command, it will begin measuring immediately each time it enters Measurement Mode—such as after applying power or following a timeout from Data Retrieval Mode.

In contrast, the **DEPLOY** command synchronizes the instrument's measurements to a fixed time schedule. Even if the instrument has been out of Measurement Mode, it will align its operation with the original timing. This means that if a scheduled measurement time has already passed when the instrument powers on, it will wait until the next scheduled interval before beginning to ping.

The **START** command is useful in integration scenarios where immediate measurement is required. However, it may result in inconsistent time stamps in the recorded data. Use **DEPLOY** if synchronized, regular sampling intervals are critical.

3.6 Telemetry

In the Signature instrument, the term telemetry refers to both the transfer of data, via serial or Ethernet connection, and the storage of selected data subsets. When telemetry is enabled, you can choose to activate file output, serial output, or both.

Telemetry data can be formatted in various ways, including NMEA and binary formats. For a complete list of supported formats, refer to the Telemetry Data Formats chapter. Note that different data types use distinct telemetry data formats.Telemetry for each data type must be configured individually using its specific command. For example does SETTMAVG configure telemetry of average current data while SETTMWAVE configure telemetry of processed wave data.

The instrument stores each individual ping to the raw data file on its internal recorder. Telemetry output (file or serial) can optionally include internally averaged velocity data, which provides a lower-bandwidth, processed data product suitable for real-time or near-real-time applications. Whenever averaging is enabled, the data undergoes an internal quality control prior to the averaging. For more details about the quality control, please refer to the Telemetry Quality Control chapter. After deployment, the raw data can be retrieved and post-processed as needed.

File output

When file output is enabled, a subset of the data is saved to the recorder in the configured data format. The file is always named `telemetryfile.bin`, and all telemetry data is written to this file. The file output option is typically used when real-time, continuous output is not feasible, and data must be stored for later retrieval. The file can be downloaded either in chunks or as a complete file. A checksum or CRC can be applied during download to ensure data integrity. System integrators can offload the telemetry file regularly using FTP or the command interface, and the file can be deleted after retrieval. The recorder is accessible in both Command Mode and Data Retrieval Mode (see Modes diagram). Downloading in Data Retrieval Mode allows the same deployment to continue when returning to Measurement Mode. Note that the instrument does not perform measurements while in Data Retrieval Mode.

Serial output

When serial output is enabled, the instrument streams telemetry data directly over the active communication interface, which may be either serial or Ethernet. The data transmitted matches exactly what is stored in the telemetry file.

3.7 Triggers

The Signature Series instruments support triggered operation, allowing for customized ping timing and the setup of Master/Slave configurations. In such setups, one instrument acts as a Master, sending trigger signals, while one or more instruments operate as Slaves, responding to those signals. Alternatively, an external device may serve as the Master, with all Signature units configured as Slaves.

The electronic platform used by Signature instruments supports four main trigger types. Availability of each type depends on the harness and cable configuration of the instrument.

1. Internal sampling
2. TTL Trigger
3. RS-485 Trigger
4. Command trigger

Configuration of trigger type and Master/Slave setup is managed via the **SETTRIG** command.

A trigger signal initiates a complete ping (i.e. pulse transmit and receive sequence). After completing the ping, the instrument resumes monitoring the trigger input line. Trigger signals received during an active ping are ignored.

As the Signature instruments support multiple ping types (e.g., altimeter, HR, velocity on slanted beams), which may be transmitted on different transducers depending on the instrument's configuration, the sequence of transducers and ping types triggered follows the user-defined configuration. If precise ping synchronization is required, the slot table returned by the READCFG command, along with exact ping timestamps available in the standard data file, can be used to determine timing accurately.

Latency analysis is outside the scope of this documentation. Typical latencies are under 10 ms. For applications requiring high-precision synchronization, please contact Nortek for assistance.

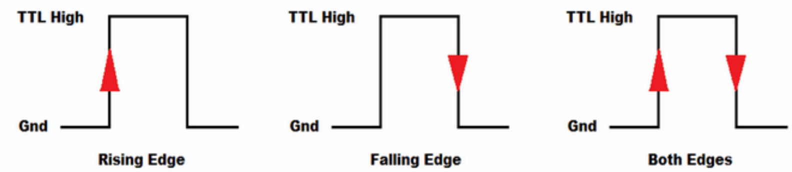
For an instrument to be used as a master when triggering, the trigger type needs to be set to **RS485EDGES**. Only continuous measurement configurations are supported in this mode, and all synchronized instruments must be of the same frequency and have the same deployment configuration.

Internal Sampling

The instrument uses internal triggers based on its clock and the user-defined sampling rate.

TTL Trigger

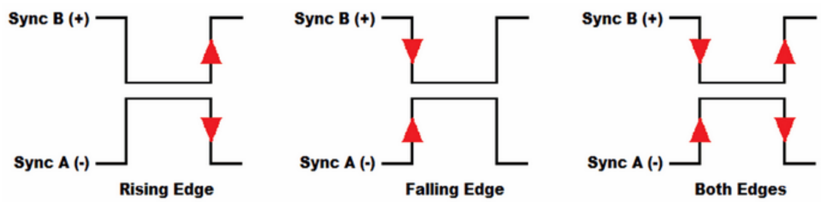
The Signature can be triggered on rising edge, falling edge, or both edges of a TTL signal. The requirements for the TTL input is V_{low} < 0.7 V and V_{high} > 2.5 V, while the TTL input line tolerates voltages between 0-5.5 V. The pulse length should be minimum 1 ms.



Trigger types

RS-485 Trigger

The Signature can be triggered on rising edge, falling edge, or both edges of a RS485 signal. The following figure shows the polarities of the differential RS-485 signal pair for the trigger types.



RS-485 trigger types

Command Trigger

The instrument is triggered by the command **TRIG** followed by CR (Carriage Return) and/or LF (Line Feed). The trigger time will be when the first character of the CR/LF is received.

4 Commands

This chapter covers the commands that can be used to control an Signature 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 tying to save the configuration or start/deploy the instrument. This doesn't necessarily mean that something is wrong with the instrument, but is most often a sign that the configuration isn't going to work. Any ERROR response can be interrogated with **GETERROR**, e.g.:

  ```
  SAVE,CONFIG
  ERROR
  GETERROR
  GETERROR,NUM=56,STR="Invalid setting: Avg Average Interval too low for the configured number of pings and profiling distance",LIM="GETAVG1LIM,AI=([360;1800])"
  OK
  ```

  Here, the instrument is reporting that we have set the average interval to be too short, and it provides the limits for the AI that are allowed if we are going to keep the same number of pings and profiling distance. You could increase the average interval, decrease the number of pings, or decrease the profiling distance (i.e. number/size of cells) to fix the error.

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
```

4.1 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
SETCLOCK	Set instrument clock	COMMAND RETRIEVAL
GETCLOCK	Get instrument clock	COMMAND RETRIEVAL
SETCLOCKSTR	Set instrument clock as string	COMMAND RETRIEVAL
GETCLOCKSTR	Get instrument clock as string	COMMAND RETRIEVAL
SETPLAN	Set deployment plan parameters	COMMAND
GETPLAN	Get deployment plan parameters	COMMAND
GETPLANLIM	Get deployment plan parameter limits	COMMAND
SETAVG	Set instrument average mode settings	COMMAND
GETAVG	Get instrument average mode settings	COMMAND
GETAVGLIM	Get instrument average mode limits	COMMAND
SETBURST	Set burst settings	COMMAND
GETBURST	Get burst settings	COMMAND
GETBURSTLIM	Get burst setting limits	COMMAND
SETBURSTHR	Set high resolution profile burst settings	COMMAND
GETBURSTHR	Get high resolution profile burst settings	COMMAND
GETBURSTHRLIM	Get high resolution profile burst setting limits	COMMAND
SETECHO	Set echosounder settings	COMMAND
GETECHO	Get echosounder settings	COMMAND
GETECHOLIM	Get echosounder setting limits	COMMAND
READECHO	Read raw echosounder parameters	COMMAND
SETALTERNATE	Set instrument alternating plan configuration	COMMAND
GETALTERNATE	Get instrument alternating plan configuration	COMMAND

GETALTERNATELIM	Get instrument alternating plan limits	COMMAND
GETMEM	Get recorder data memory usage	COMMAND
SETTRIG	Set trigger settings	COMMAND
GETTRIG	Get trigger settings	COMMAND
GETTRIGLIM	Get trigger setting limits	COMMAND
TRIG	Trigger a specified measurement	MEASUREMENT
SETEXTSENSOR	Set external sensor settings	COMMAND
GETEXTSENSOR	Get external sensor settings	COMMAND
GETEXTSENSORLIM	Get external sensor setting limits	COMMAND
GETPWR	Returns the power consumption in mW for the various measurements enabled as well as the overall value	COMMAND
READABSPRESSURE	Returns the absolute pressure reading from the instrument	COMMAND
GETPRECISION	Returns the precision for primary plan	COMMAND
GETPRECISION1	Returns the precision for secondary plan	COMMAND
SETUSER	Set instrument user settings	COMMAND
GETUSER	Get instrument user settings	COMMAND
GETUSERLIM	Get instrument calibration parameter limits	COMMAND
ID	Get instrument Id	COMMAND
SETDEFAULT	Reload default settings	COMMAND
SAVE	Save settings for next measurement	COMMAND
DEPLOY	Deploy the instrument	COMMAND
START	Start the instrument	COMMAND
MC	Go into command mode	CONFIRMATION
RM	Go into data retrieval mode	CONFIRMATION
CO	Go into measurement mode	CONFIRMATION RETRIEVAL
FWRITE	Write tag or string to file	COMMAND CONFIRMATION RETRIEVAL
POWERDOWN	Set instrument in sleep mode	COMMAND
ERASE	Erase all files on the recorder	COMMAND
FORMAT	Format the recorder	COMMAND
SECREBOOT	Reboots the SEC	COMMAND
LISTFILES	Lists the files on instrument	COMMAND

		RETRIEVAL
DOWNLOAD	Read file data	COMMAND RETRIEVAL
READCFG	Read current configuration	COMMAND
INQ	Inquires the instrument state	COMMAND CONFIRMATION RETRIEVAL MEASUREMENT
GETSTATE	Returns information about the current operational state of the instrument	COMMAND CONFIRMATION MEASUREMENT RETRIEVAL
GETERROR	Returns a full description of the last error condition to occur	COMMAND CONFIRMATION MEASUREMENT RETRIEVAL
GETALL	Retrieves all relevant configuration information for the instrument	COMMAND
RECSTAT	Return recorder state	COMMAND RETRIEVAL
GETMISCLIM	This command returns configuration limits that cannot be returned through the relevant commands	COMMAND
GETXFAVG	Returns the "Beam to XYZ" transfer matrix for average measurements	COMMAND
GETXFBURST	Returns the "Beam to XYZ" transfer matrix for burst measurements	COMMAND
SETTMAVG	Set averaging mode telemetry settings	COMMAND
GETTMAVG	Get averaging mode telemetry settings	COMMAND
GETTMAVGLIM	Get instrument averaging mode limits	COMMAND
SETTMBURST	Set burst mode telemetry settings	COMMAND
GETTMBURST	Get burst mode telemetry settings	COMMAND
GETTMBURSTLIM	Get instrument burst mode limits	COMMAND
SETTMALTI	Set altimeter telemetry settings	COMMAND
GETTMALTI	Get altimeter telemetry settings	COMMAND
GETTMALTILIM	Get instrument altimeter limits	COMMAND
TMSTAT	Returns the length of the telemetry file	COMMAND RETRIEVAL
DOWNLOADTM	Read telemetry file data	COMMAND

		RETRIEVAL
STOREHEADERTM	Stores the GETALL to the telemetry file	COMMAND
ERASETM	Erase the telemetry file	COMMAND RETRIEVAL
TAG	Write a Tag to output file and data output	COMMAND CONFIRMATION MEASUREMENT RETRIEVAL
SETUSERECHO- MAJORANGLE	Set calibration values for major beam angle	COMMAND
GETUSERECHO- MAJORANGLE	Get calibration values for major beam angle	COMMAND
GETUSERECHO- MAJORANGLELIM	Get limits for calibration values for major beam angle	COMMAND
SETUSERECHO- MAJOROFFSET	Set calibration values for major beam angle offset	COMMAND
GETUSERECHO- MAJOROFFSET	Get calibration values for major beam angle offset	COMMAND
GETUSERECHO- MAJOROFFSETLIM	Get limits for calibration values for major beam angle offset	COMMAND
SETUSERECHO- MINORANGLE	Set calibration values for minor beam angle	COMMAND
GETUSERECHO- MINORANGLE	Get calibration values for minor beam angle	COMMAND
GETUSERECHO- MINORANGLELIM	Get limits for calibration values for minor beam angle	COMMAND
SETUSERECHO- MINOROFFSET	Set calibration values for minor beam angle offset	COMMAND
GETUSERECHO- MINOROFFSET	Get calibration values for minor beam angle offset	COMMAND
GETUSERECHO- MINOROFFSETLIM	Get limits for calibration values for minor beam angle offset	COMMAND
SETUSERECHOGAIN	Set calibration values for gain	COMMAND
GETUSERECHOGAIN	Get calibration values for gain	COMMAND
GETUSERECHOGAINLIM	Get limits for calibration values for gain	COMMAND
SETUSERECHO- TWOWAYANGLE	Set calibration values for two way beam angle	COMMAND
GETUSERECHO- TWOWAYANGLE	Get calibration values for two way beam angle	COMMAND

GETUSERECHO-TWOWAYANGLELIM	Get limits for calibration values for two way beam angle	COMMAND
PTPSET	Set precision time protocol parameters	COMMAND
PTPGET	Get precision time protocol parameters	COMMAND
BBPWAKEUP	Wakes up the Doppler processor (ethernet interface only)	COMMAND CONFIRMATION MEASUREMENT RETRIEVAL
SETWAVEPROC	Set wave processing settings	COMMAND
GETWAVEPROC	Get wave processing settings	COMMAND
GETWAVEPROCLIM	Get wave processing setting limits	COMMAND
SETTMWAVE	Set wave telemetry settings	COMMAND
GETTMWAVE	Get wave telemetry settings	COMMAND
GETTMWAVELIM	Get wave telemetry setting limits	COMMAND
GETHW	Returns hardware specifications	COMMAND CONFIRMATION MEASUREMENT RETRIEVAL
ADDLICENSE	Add license key	COMMAND
DELETELICENSE	Delete license key	COMMAND
LISTLICENSE	Lists all license keys in instrument	COMMAND

4.2 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
```

4.3 Instrument main settings

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

Instrument main settings

Argument	Description
BR	Baud Rate 300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200, 230400, 460800, 625000, 1250000 Unit: [bit/s]
RS	Serial protocol 232, 422
LED	Enable/disable LED blink in head. When set to "ON24H" the LED will illuminate the first 24 hours of the measurement. "OFF": Turn LED Off "ON": Turn LED On "ON24H": Turn LED On for 24 hours
ORIENT	Sets the instrument orientation. Not used for DVL. "AUTOXUPDOWN": Auto X up/down "AUTOYUPDOWN": Auto Y up/down "AUTOZUPDOWN": Auto Z up/down "ZUP": Z up "ZDOWN": Z down "XUP": X up "XDOWN": X down "YUP": Y up "YDOWN": Y down "AHRS3D": The instrument will detect and change the orientation as the instrument is moved. Used if the instrument has an AHRS installed and does not have a constant defined UP direction
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,LED="ON"
```

GETINST

Get instrument main settings

Example:
```
GETINST,BR=460800,RS=232,LED="ON",ORIENT="AUTOZUPDOWN",CMTOUT=300,D
MTOUT=60,CFMTOUT=60
```

GETINSTLIM

Set instrument main setting limits

4.4 Clock settings

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

Instrument Real Time Clock specified in date parts

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

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

SETCLOCK

Set instrument clock

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

GETCLOCK

Get instrument clock

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

4.5 Clock settings as strings

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

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

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

SETCLOCKSTR

Set instrument clock as string

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

GETCLOCKSTR

Get instrument clock as string

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

4.6 Deployment plan parameters

Commands: SETPLAN, GETPLAN, GETPLANLIM,
Command type: CONFIGURATION
Mode: COMMAND

Set and get high-level plan configuration settings.

Argument	Description
MIAVG	Averaging measurements interval Unit: [s]
AVG	Enable(1)/disable(0) averaging measurements.
SA	Salinity. Unit: [ppt]
BURST	Enable(1)/disable(0) burst measurements.
MIBURST	Burst measurements interval. Unit: [s]
SV	Sound velocity. SV = 0 will set sensor to use measured sound velocity Unit: [m/s]
FN	Filename of the raw data file where all the measured binary data will be stored. If FN="", no data is stored on the recorder. FN must be changed if a file FN with a different configuration already exists on the recorder. Values: ['a'; 'z']; ['A'; 'Z']; ['0'; '9']; '_'; '.' | Max Length: 30
SO	Enable(1)/disable(0) serial output With this option enabled, the instrument will stream all collected raw binary data live, either over serial or ethernet connection.
FREQ	Transmit frequency. This is normally the instruments (one) frequency. In the case of the Signature 55 we may choose between 55Hz or 75Hz. Unit: [khz]
NSTT	Number slot time table. Set to 0 giving the default number of slots

Note: The valid range for the various arguments should be verified using the GETPLANLIM command, as the values listed here may change with firmware versions and instrument frequencies.

SETPLAN

Set deployment plan parameters

Example:
```
SETPLAN,FREQ=75
```

GETPLAN

Get deployment plan parameters

GETPLANLIM

Get deployment plan parameter limits

4.7 Average mode settings

Commands: SETAVG, GETAVG, GETAVGLIM,
Command type: CONFIGURATION
Mode: COMMAND

The instrument averaging mode settings and relevant limits.

Argument	Description
NC	Number of cells
CS	Cell size Unit: [m]
BD	Blanking distance Unit: [m]
DF	Data format 3: DF3 Binary v3
CY	Co-ordinate system BEAM, XYZ, ENU
PL	Power level (range -20.0 dB to 0.0 dB, -100 dB to switch off transmit). Unit: [dB] | Values: [-20; 0]; -100
AI	Average interval Unit: [s]
VR	Velocity range along beam Unit: [m/s]
NPING	Number of pings
NB	Number of beams
CH	Beam selection. E.g. CH=123 will enable beams 1, 2 and 3.
MUX	Enable(1)/disable(0) multiplexing. 0: ping all beams in parallel 1: ping beams in sequence

BW	Bandwidth selection. NARROW, BROAD
ALTI	Enable(1)/disable(0) altimeter measurements
BT	Enable(1)/disable(0) bottom tracking
ICE	Enable(1)/disable(0) ice tracking
ALTISTART	Altimeter start Unit: [m]
ALTIEND	Altimeter start Unit: [m]
RAWALTI	Raw altimeter recording interval

Note: The actual valid range for the various parameters for your specific instrument and firmware can be found by using the GETAVGLIM command. This command has the same arguments as the SETAVG/GETAVG commands shown in the list above. The output format for limits is described in Data Limit Formats

SETAVG

Set instrument average mode settings

Example:
`SETAVG,BD=1`

GETAVG

Get instrument average mode settings

GETAVGLIM

Get instrument average mode limits

4.8 Burst settings

Commands: SETBURST, GETBURST, GETBURSTLIM,
Command type: CONFIGURATION
Mode: COMMAND

Burst profile settings and relevant limits.

Argument	Description
NC	Number of cells.

NB	Number of beams.
CS	Cell size. Unit: [m]
BD	Blanking distance. Unit: [m]
DF	Data format 3: DF3 Binary v3
CY	Co-ordinate System. BEAM, XYZ, ENU
PL	Power level (range -20.0 dB to 0.0 dB, -100 dB to switch off transmit). This parameter is instrument dependent and dependent on the power consumption of the deployment as a whole. Unit: [dB] | Values: [-20; 0]; -100
SR	Sampling rate. Unit: [Hz]
NS	Number of samples.
VR	Velocity range along beam. Unit: [m/s]
VR5	Velocity range along beam 5. Unit: [m/s]
NPING	Number of pings.
CH	Beam selection. E.g. CH=123 will enable beams 1, 2 and 3.
ALTI	Enable(1)/disable(0) altimeter measurements.
BT	Enable(1)/disable(0) bottom tracking.
DISV	Disable(1)/enable(0) velocity measurements.
ECHO	Enable(1)/disable(0) echosounder.
RAWALTI	Raw altimeter recording interval.
ALTISTART	Altimeter start. Unit: [m]
ALTIEND	Altimeter start. Unit: [m]
HR	Enable(1)/disable(0) high resolution mode.
HR5	Enable(1)/disable(0) high resolution mode for beam 5.

Note: The actual valid range for the various parameters for your specific instrument and firmware 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

Example:
```
SETBURST,CS=1
```

GETBURST

Get burst settings

GETBURSTLIM

Get burst setting limits

4.9 High resolution burst settings

Commands: SETBURSTHR, GETBURSTHR, GETBURSTHRLIM,
Command type: CONFIGURATION
Mode: COMMAND

License: High Resolution

Burst high resolution profile settings and relevant limits.

Argument	Description
PROC	Input to processing the HR profile. 0: Pulse Coherent Processing using a single ambiguity. 1: Pulse Coherent Processing with Extended Velocity Range (EVR).
LAG	Distance between two transmit pulses on the slanted beams. Unit: [m]
LAG5	Distance between two transmit pulses on vertical beam. Unit: [m]
SCORR	Number of ambiguities (side correlators) to resolve when using Extended Velocity Range (EVR).
NC	Number of cells.
CS	Cell size. Unit: [m]
BD	Blanking distance. Unit: [m]

PL	Power level (range -20.0 dB to 0.0 dB, -100 dB to switch off transmit). This parameter is instrument dependent and dependent on the power consumption of the deployment as a whole. Unit: [dB] | Values: [-20; 0]; -100

SETBURSTHR

Set high resolution profile burst settings

Example:
`SETBURSTHR,BD=3`

GETBURSTHR

Get high resolution profile burst settings

GETBURSTHRLIM

Get high resolution profile burst setting limits

4.10 Echosounder settings

Commands: SETECHO, GETECHO, GETECHOLIM,
Command type: CONFIGURATION
Mode: COMMAND

Echosounder settings and relevant limits

Argument	**Description**
NC	Number of cells
BINSIZE	Bin size Unit: [m]
BD	Blanking distance Unit: [m]
DF	Data format 3: DF3 Binary v3
FREQ1	Enable and set frequency 1 of echogram. Unit: [kHz]
NBINF1	Used only on the Signature100 which supports up to 5 packages pr frequency index. Processing is then done on the different frequency bands.

	For other instruments it must be set to 1. 1
XMIT1	Transmission length on frequency 1. Unit: [msec]
PL1	Power level on frequency 1 (range -20.0 dB to 0.0 dB, -100 dB to switch off transmit). This parameter is instrument dependent and dependent on the power consumption of the deployment as a whole. Unit: [dB] | Values: [-20; 0]; -100
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]
NBINF2	Used only on the Signature100 which supports up to 5 packages pr frequency index. Processing is then done on the different frequency bands. For other instruments it must be set to 1.
XMIT2	Transmission length on frequency 2. Unit: [msec]
PL2	Power level on frequency 2 (range -20.0 dB to 0.0 dB, -100 dB to switch off transmit). This parameter is instrument dependent and dependent on the power consumption of the deployment as a whole. Unit: [dB] | Values: [-20; 0]; -100
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.
FREQ3	Enable and set frequency 3 of echogram. Unit: [kHz]
NBINF3	Used only on the Signature100 which supports up to 5 packages pr frequency index. Processing is then done on the different frequency bands. For other instruments it must be set to 1.
XMIT3	Transmission length on frequency 3. Unit: [msec]
PL3	Power level on frequency 3 (range -20.0 dB to 0.0 dB, -100 dB to switch off transmit). This parameter is instrument dependent and dependent on the power consumption of the deployment as a whole. Unit: [dB] | Values: [-20; 0]; -100
PULSECOMP3	Enable/disable pulse compression on frequency 3.

CH3	Frequency 3 channel beam selection. E.g. CH=123 will enable beams 1, 2 and 3.
RAW3	Frequency 3 raw echo data recording interval.

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

SETECHO

Set echosounder settings

Example:
`SETECHO,NC=30,BINSIZE=1`

GETECHO

Get echosounder settings

GETECHOLIM

Get echosounder setting limits

4.11 Read raw echosounder parameters

Command: READECHO
Command type: INFO
Mode: COMMAND **License:** EchoSounder

This command provides information of the raw echosounder data records. It is typically used in software by parsing the header of the ad2cp-files, enabling the software to know the amount of data that is used for the raw echo records in the file.

The parameters all count to three because we may send out 3 pings. This corresponds to FREQ1, FREQ2 and FREQ3 and their corresponding parameters in SETECHO.

Argument	Description
NRAWSAMP1	Number of samples of raw data in first ping.
NRAWXMIT1	The number of samples for the stored, synthetic, transmit pulse which is stored when pulse compression is enabled. When pulse compression is disabled it returns zero since the synthetic transmit pulse is not output in those cases.
SAMPFREQ1	Sampling frequency of the raw data.
RANGESTART1	The "distance" of the first sample with reference to the end of the transmit pulse which is the zero reference.

NRAWSAMP2	Number of samples of raw data in second ping.
NRAWXMIT2	The number of samples for the stored, synthetic, transmit pulse which is stored when pulse compression is enabled. When pulse compression is disabled it returns zero since the synthetic transmit pulse is not output in those cases.
SAMPFREQ2	Sampling frequency of the raw data.
RANGESTART2	The "distance" of the first sample with reference to the end of the transmit pulse which is the zero reference.
NRAWSAMP3	Number of samples of raw data in third ping.
NRAWXMIT3	The number of samples for the stored, synthetic, transmit pulse which is stored when pulse compression is enabled. When pulse compression is disabled it returns zero since the synthetic transmit pulse is not output in those cases.
SAMPFREQ3	Sampling frequency of the raw data.
RANGESTART3	The "distance" of the first sample with reference to the end of the transmit pulse which is the zero reference.

Example:
```
READECHO
1974,0,0,0,0,0,4464.3,0.0,0.0,0.750,0.000,0.000
OK
```

4.12 Alternating plan configuration

Commands: SETALTERNATE, GETALTERNATE, GETALTERNATELIM,
Command type: CONFIGURATION
Mode: COMMAND

The SETALTERNATE/GETALTERNATE command allows two different configurations to be run consecutively in time. The primary configuration (defined by SETPLAN, SETBURST, SETAVG, SETTMAVG, SETBT) is run for "PLAN" seconds, after which the unit powers down for a given period of time ("IDLE" seconds). The alternate configuration (defined by SETPLAN1, SETBURST1, SETAVG1, SETTMAVG1, SETBT1) is then run for "PLAN1" seconds and the unit powers down for "IDLE1" seconds. The configuration is then switched back to the primary and the process is repeated.

Argument	Description
EN	Enable or disable the alternate configuration mode
PLAN	Primary configuration run time Unit: [s]
IDLE	Primary configuration idle time Unit: [s]
PLAN1	Alternate configuration run time

	Unit: [s]
IDLE1	Alternate configuration idle time Unit: [s]

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

SETALTERNATE

Set instrument alternating plan configuration

Example:
```
SETALTERNATE,EN=1,PLAN=60,IDLE=60,PLAN1=60,IDLE1=60
```

GETALTERNATE

Get instrument alternating plan configuration

GETALTERNATELIM

Get instrument alternating plan limits

4.13 Get memory usage

Command: GETMEM
Command type: INFO
Mode: COMMAND

Returns the amount of memory that will be stored on the recorder per hour for the burst and average data as well as the combined plan value. Alternate mode values is also supported.

Argument	Description
PLAN	Combined burst and average memory usage for primary plan Unit: [MB/h]
BURST	Burst memory usage for primary plan Unit: [MB/h]
AVG	Average data memory usage for primary plan Unit: [MB/h]
PLAN1	Combined burst and average memory usage for secondary plan Unit: [MB/h]

BURST1	Burst memory usage for secondary plan Unit: [MB/h]
AVG1	Average data memory usage for secondary plan Unit: [MB/h]
TOTAL	Total memory usage Unit: [MB/h]

Example:
`GETMEM,TOTAL`

4.14 Trigger settings

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

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

Argument	Description
EN	Enable/disable external trigger functionality.
TRIG	Specifies trigger type. The available trigger types will depend on the harness/cable used with the instrument. "TTLEDGES": Trigger on both rising- and falling edge of a TTL signal. "TTLRISE": Trigger on rising edge of a TTL signal. "TTLFALL": Trigger on falling edge of a TTL signal. "RS485EDGES": Trigger on both edges of a RS-485 signal. "RS485RISE": Trigger on rising edge of a RS-485 signal. "RS485FALL": Trigger on falling edge of a RS-485 signal. "COMMAND": Trigger on Command. When the TRIG parameter of the SETTRIG command is set to "COMMAND" the AD2CP is trigged by sending a "TRIG[CrLf]" command. The trigger time will be when the end of the [CrLf] is received
TRIGOUTPUT	TRIGOUTPUT=1 enables master trigger output. This enables several instruments to be synchronized together with one instrument acting as master. For an instrument to be used as a master when triggering, the trigger type needs to be set to RS485EDGES. Only continuous measurement configurations are supported in this mode, and all synchronized instruments must be onf the same frequency and have the same deployment configuration.

SETTRIG

Set trigger settings

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

GETTRIG

Get trigger settings

GETTRIGLIM

Get trigger setting limits

4.15 Trigger measurement

Command: TRIG
Command type: ACTION
Mode: MEASUREMENT

Command used for trigging measurement when Trigger is enabled and trigger type equals "COMMAND".

Argument	Description
ID	Counting number (optional).

Example:
```
TRIG,ID=123
```

4.16 External sensor settings

Commands: SETEXTSENSOR, GETEXTSENSOR, GETEXTSENSORLIM,
Command type: CONFIGURATION
Mode: COMMAND

External sensor configuration.

Argument	Description
EN	Enable external sensor.

SETEXTSENSOR

Set external sensor settings

Example:

```
SETEXTSENSOR,EN=0
OK
```

GETEXTSENSOR

Get external sensor settings

Example:

```
GETEXTSENSOR,EN
0
OK
```

GETEXTSENSORLIM

Get external sensor setting limits

4.17 Get power use

Command: GETPWR
Command type: INFO
Mode: COMMAND

Returns the power consumption in milliWatts for the various measurements enabled as well as the overall value. The plan values include the sleep mode power consumption in addition to the sum of average and burst mode values. The reported values are adjusted according to the input voltage to the system when the command is executed.

Argument	Description
PLAN	Combined power use on both burst and average sampling for secondary plan Unit: [mW]
BURST	Power use on burst data sampling for secondary plan Unit: [mW]
AVG	Power use on average data sampling for secondary plan Unit: [mW]
TOTAL	Total power use Unit: [mW]

Example:

```
GETPWR,TOTAL
```

4.18 Read absolute pressure

Command: READABSPRESSURE
Command type: INFO
Mode: COMMAND

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

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

Example:

```
READABSPRESSURE
10.812
OK
```

4.19 Get measured precision

Commands: GETPRECISION, GETPRECISION1,
Command type: INFO
Mode: COMMAND

Returns the precision in the horizontal plane and along the beam in cm/s for the various measurement modes.

Argument	Description
AVGHORZ	Precision in the horizontal plane in average mode Unit: [cm/s]
BURSTHORZ	Precision in the horizontal plane in burst mode Unit: [cm/s]
BEAM5	Precision in the vertical plane (beam 5) in burst mode Unit: [cm/s]
AVGBEAM	Precision along beam in average mode Unit: [cm/s]
BURSTBEAM	Precision along beam in burst mode Unit: [cm/s]

GETPRECISION

Returns the precision for primary plan

Example:

```
GETPRECISION,AVGHORZ
```

GETPRECISION1

Returns the precision for secondary plan

Example:
```
GETPRECISION1,AVGHORZ
```

4.20 Instrument user settings

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

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

Argument	Description
POFF	Set the offset value of the pressure sensor. Unit: [dbar] | Values: [0; 11]
DECL	Magnetic declination. The value will be added to the compass value 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=100,HY=100,HZ=100
OK
SAVEUSER
OK
```

GETUSER

Get instrument user settings

GETUSERLIM

Get instrument calibration parameter limits

4.21 Get instrument ID

Command: ID
Command type: INFO
Mode: COMMAND

Commands for accessing instrument name and serial number

Argument	Description
STR	Instrument name Values: '_'; '.' | Max Length: 64
SN	Serial number Values: [0; 2147483647]

Example:

```
ID
"Signature1000",900002
ID,STR
"Signature1000"
```

4.22 Reload default settings

Command: SETDEFAULT
Command type: ACTION
Mode: COMMAND

Reload default settings.

Argument	Description
CONFIG	Restore all settings below except USER and INST to default values. Legacy argument ALL acts as CONFIG.
AVG	Restore average mode settings to default.
INST	Restore instrument main settings to default.
TMAVG	Restore telemetry average settings to default.
PLAN	Restore plan settings to default.
BURST	Restore burst settings to default.
PTP	Restore precision time protocol to default.
BT	Restore bottom track settings to default.

USER	Restore user calibration to default.
TMBURST	Restore telemetry burst to default.
TMALTI	Restore altimeter telemetry settings to default.
DVL	Restore DVL settings to default.

Example:
`SETDEFAULT,CONFIG`

4.23 Save settings

Command: SAVE
Command type: ACTION
Mode: COMMAND

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

Argument	Description
CONFIG	Save all settings except INST and USER settings. Legacy argument ALL acts as CONFIG.
AVG	Save AVG settings.
INST	Save INST settings.
TMAVG	Save telemetry average settings.
PLAN	Save plan settings.
BURST	Save burst settings.
PTP	Save precision time protocol settings.
TMBT	Save telemetry bottom track settings.
USER	Save user instrument settings.
TMBURST	Save telemetry burst settings.
TMALTI	Save altimeter telemetry settings.
DVL	Save DVL settings.

Example:
`SAVE,CONFIG`

4.24 Deploy instrument

Command: DEPLOY
Command type: ACTION
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
```

4.25 Start instrument

Command: START
Command type: ACTION
Mode: COMMAND

Start the instrument (go in measurement mode). Specify the order of the alternating plans by using the PLAN argument.

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

Argument	Description
PLAN	Select plan to start 0: Plan 1: Plan1

Example:
```
START,PLAN=0
```

4.26 Enter command mode

Command: MC
Command type: ACTION
Mode: CONFIRMATION

Sets instrument in command mode from confirmation mode.

Example:
```
MC
```

4.27 Data retrieval mode

Command: RM
Command type: ACTION
Mode: CONFIRMATION

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

Example:
```
RM
```

4.28 Enter measurement mode

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

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

Example:
```
CO
```

4.29 Write to file

Command: FWRITE
Command type: ACTION
Mode: COMMAND, CONFIRMATION, RETRIEVAL

Write tag or string to file

Argument	Description
FNUM	File identifier for telling which file the info is written to. Default is 0. 0: File defined in the PLAN command 1: Telemetry file Default: 0

ID	Identifier. Tell the parser how to interpret the string. Default value: 0 0: Comment 1: Dive location 2: Surface location Default: 0
STR	String. Max Length: 200

Note: Fwrite STR and B64STR cannot be set together.

Example:
```
FWRITE,FNUM=1,STR="ABCDEF"
```

4.30 Power down

Command: POWERDOWN
Command type: ACTION
Mode: COMMAND

Power down the instrument to set it in sleep mode.

Example:
```
POWERDOWN
```

4.31 Erase files on recorder

Command: ERASE
Command type: ACTION
Mode: COMMAND

Erase all files on the recorder

Argument	Description
CODE	Code should be 9999 9999

Example:
```
ERASE,CODE=9999
```

4.32 Format recorder

Command: FORMAT
Command type: ACTION
Mode: COMMAND

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

Argument	Description
CODE	Code should be 9999 9999

Example:
`FORMAT,CODE=9999`

4.33 SEC reboot

Command: SECREBOOT
Command type: ACTION
Mode: COMMAND

Reboots the SEC (Interface processor). This command makes it possible to reboot the SEC without going via the web interface.

Argument	Description

Example:
`SECREBOOT,9999`

4.34 List files

Command: LISTFILES
Command type: INFO
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.

Example:
`LISTFILES,OPT="NORMAL"`

4.35 Download

Command: DOWNLOAD
Command type: ACTION
Mode: COMMAND, RETRIEVAL

This command reads a file from the recorder to the terminal window.

Argument	Description
FN	Filename Values: ['a'; 'z']; ['A'; 'Z']; ['0'; '9']; '_'; '.' | Max Length: 30
SA	Start address (offset) of the first byte to be returned
LEN	Number of bytes to be downloaded
CRC	Use Cyclic redundancy check
CKS	Use Checksum

Note: If no parameters other than the file name are sent with the DOWNLOAD command, the complete file is directly returned, without the number of bytes to follow. The end of the file can then be detected by parsing the OK<CR><LF>.

The parameters can be used to download the file in several pieces. The number of bytes to follow will then be returned in ASCII format and terminated with <CR><LF> before the data is output. The end of the file stream is terminated with OK<CR><LF>. A cyclic redundancy check or a checksum will then be added to verify data integrity during download. The complete file can also be downloaded in this way by specifying SA=0 and a large value for LEN. The actual file size is then returned before the data follows.

Example:

```
DOWNLOAD,FN="TestFile.ad2cp",SA=0,LEN=4096,CRC=1,CKS=0
4096
<binary or ASCII data>
23432
OK
```

4.36 Read configuration

Command: READCFG
Command type: CONFIGURATION
Mode: COMMAND

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

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

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

4.37 Inquire state

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

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

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

Note: Parameters used to represent instrument mode:

0000: Bootloader/Firmware upgrade

0001: Measurement

0002: Command

0004: Data Retrieval

0005: Confirmation

0006: FTP-mode

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

4.38 Get instrument state

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

Returns information about the current operational state of the instrument

Argument	Description
MODE	Current instrument state. 1: Measurement (START command received). 2: In command mode. 3: DEPLOY command received and deployment running. 4: Data retrieval. 5: Confirmation. 6: Network FTP. 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. 12: internal processing in progress.
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 BREAK command. 3: Last startup/reboot caused by Wakeup by Real time clock. 4: Last startup/reboot caused by WatchDog
INTPROC	Internal processing Active

Example:
```
GETSTATE,WAKEUP,CURRTIME
```

4.39 Get error

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

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

Argument	Description
NUM	Integer error value
STR	Text description
LIM	Valid limits as text

Example:

```
SETAVG,CS=2.5
OK
SAVE,CONFIG
ERROR
GETERROR
40,"Invalid setting: Avg Cell Size","GETAVGLIM,CS=([0.20;2.00])"
OK
```

4.40 Get all

Command: GETALL
Command type: INFO
Mode: COMMAND

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

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

Example:

```
See GETALL chapter.
```

4.41 Get recorder state

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

Returns recorder state.

Argument	Description
SS	Number of bytes in a sector.

	Unit: [bytes]
CS	Number of bytes in one cluster. Unit: [bytes]
FC	Number of bytes in free clusters. Unit: [bytes]
TC	Total number of bytes in clusters. Unit: [bytes]
VS	Volume Size in bytes. Unit: [bytes]
FF	Number of free files.
TF	Total number of files.
PNM	SD card type and firmware

Example:
`RECSTAT,VS`

4.42 Get configuration limits

Command: GETMISCLIM
Command type: INFO
Mode: COMMAND

Returns configuration limits.

Argument	Description
AVGPR	Returns the total profiling range for averaged measurements (SETAVG) Unit: [m]
BURSTPR	Returns the total profiling range for burst profile measurements (SETBURST) Unit: [m]
BURSTHRPR	Returns the total profiling range for burst HR profile measurements (SETBURSTHR) Unit: [m]

Note: The output format for limits is described in Data Limit Formats

Example:
`GETMISCLIM,BURSTPR`

4.43 Get transfer matrices

Commands: GETXFAVG, GETXFBURST,
Command type: INFO
Mode: COMMAND

Returns the "Beam to XYZ" transfer matrix for the current setup. If the number of beams is 1 or 2, only ROWS and COLS are returned.

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

Note: Return is ROWS, COLS, M11, M12 ... M43, M44.

Cell indexes are: first number for row and second index for column.

GETXFAVG

Returns the "Beam to XYZ" transfer matrix for average measurements

Example:
GETXFAVG

GETXFBURST

Returns the "Beam to XYZ" transfer matrix for burst measurements

Example:
```
GETXFBURST
GETXFBURST,ROWS=4,COLS=4,M11=1.183,M12=0.000,M13=-
1.183,M14=0.000,M21=0.000,M22=1.183,M23=0.000,M24=-
1.183,M31=0.552,M32=0.000,M33=0.552,M34=0.000,M41=0.000,M42=0.552,M
43=0.000,M44=0.552
GETXFAVG,ROWS=3,COLS=3,M11=1.183,M12=0.000,M13=-
1.183,M21=1.183,M22=-2.366,M23=1.183,M31=0.552,M32=0.000,M33=0.552
```

4.44 Average mode telemetry settings

Commands: SETTMAVG, GETTMAVG, GETTMAVGLIM,
Command type: CONFIGURATION
Mode: COMMAND

The instrument averaging mode telemetry settings and relevant limits.

Argument	Description
EN	Enable averaging mode telemetry.
CD	Cells divisor. Define the interval at which cells should be outputted from your profile. If CD=1 every cell will be included, if CD=2 every second cell will be outputted etc.
PD	Profile divisor. Define how often the collected profile should be outputted. If PD=1 the profile from each ping will be outputted, if PD=2 every second profile will be outputted etc. Note that PD>1 is not available when averaging is enabled.
AVG	Number of seconds included in the telemetry average. If 0, no averaging is done and every ping will be outputted Unit: [s]
TV	Enable(1)/disable(0) velocity output.
TA	Enable(1)/disable(0) amplitude output
TC	Enable(1)/disable(0) correlation output.
CY	Coordinate system. BEAM, XYZ, ENU
FO	Enable(1)/disable(0) file output. The telemetry data will be written to the file named telemetryfile.bin on the recorder
SO	Enable(1)/disable(0) serial output. The telemetry data will be outputted live, either over serial or ethernet connection

DF	Telemetry data format. For examples, see Telemetry Data Format chapter.
DISTILT	Disable tilt.
TPG	Enable(1)/disable(0) output of the Percentage Good value.
MAPBINS	Enable vertical bin mapping.
CORRTH	Correlation threshold

Note: It is recommended to enable TPG when averaging (AVG>0) is also enabled.

The actual valid range for the various parameters for your specific instrument and firmware can be found by using the GETTMAVGLIM command. This command has the same arguments as the SETTMAVG/GETTMAVG commands shown in the list above. The output format for limits is described in Data Limit Formats.

SETTMAVG

Set averaging mode telemetry settings

Example:
```
SETTMAVG,EN=1,AVG=30
```

GETTMAVG

Get averaging mode telemetry settings

GETTMAVGLIM

Get instrument averaging mode limits

4.45 Burst mode telemetry settings

Commands: SETTMBURST, GETTMBURST, GETTMBURSTLIM,
Command type: CONFIGURATION
Mode: COMMAND

The instrument burst mode telemetry settings and relevant limits.

Argument	Description
EN	Enable(1)/disable(0) burst telemetry
NS	Number of burst samples in telemetry to output
CY	Coordinate System BEAM, XYZ, ENU

FO	Enable(1)/disable(0) file output. The telemetry data will be written to the file named telemetryfile.bin on the recorder
SO	Enable(1)/disable(0) serial output. The telemetry data will be outputted live, either over serial or ethernet connection
DF	Telemetry data format. For examples, see Telemetry Data Format chapter.
ENAVG	Enable(1)/disable(0) averaging over the number of samples (NS) value
MAPBINS	Enable(1)/disable(0) vertical bin mapping

Note: The actual valid range for the various parameters for your specific instrument and firmware can be found by using the GETTMBURSTLIM command. This command has the same arguments as the SETTMBURST/GETTMBURST commands shown in the list above. The output format for limits is described in Data Limit Formats.

SETTMBURST

Set burst mode telemetry settings

Example:
```
SETTMBURST,EN=1,NS=0
```

GETTMBURST

Get burst mode telemetry settings

GETTMBURSTLIM

Get instrument burst mode limits

4.46 Altimeter telemetry settings

Commands: SETTMALTI, GETTMALTI, GETTMALTILIM,
Command type: CONFIGURATION
Mode: COMMAND

The instrument altimeter telemetry settings and relevant limits.

Argument	Description
EN	Enable(1)/disable(0) altimeter telemetry.
TS	Include time stamp.
TQ	Include quality parameter.

FO	Enable(1)/disable(0) file output. The telemetry data will be written to the file named telemetryfile.bin on the recorder
SO	Enable(1)/disable(0) serial output. The telemetry data will be outputted live, either over serial or ethernet connection
DF	Altimeter telemetry format. For examples, see Telemetry Data Format chapter. 200: NMEA (PNORA) format without tags. 201: NMEA (PNORA) format with tags.
TPR	Enable(1)/disable(0) output of pitch and roll (in DF200/201 data format).

SETTMALTI

Set altimeter telemetry settings

Example:
```
SETTMALTI,EN=1,TS=1,TQ=1
```

GETTMALTI

Get altimeter telemetry settings

GETTMALTILIM

Get instrument altimeter limits

4.47 Get telemetry file size

Command: TMSTAT
Command type: ACTION
Mode: COMMAND, RETRIEVAL

This command returns the length (# of bytes) of the telemetry file.

Example:
```
TMSTAT
13500
OK
```

4.48 Download telemetry

Command: DOWNLOADTM
Command type: ACTION
Mode: COMMAND, RETRIEVAL

This command enables reading the telemetry file which can be created during measurement by using the appropriate SETTMxxx commands. The formats are described in the section detailing Data Formats.

Argument	Description
SA	Start address (offset) of the first byte to be returned.
LEN	Number of bytes to be downloaded.
CRC	Use Cyclic redundancy check (CRC-16). CRC=1 enables CRC. CRC cannot be enabled when CKS=1.
CKS	Use Checksum. CKS=1 enables checksum. CKS cannot be enabled when CRC=1.

Note: If no parameters are sent with the DOWNLOADTM command the complete file is directly returned, without the number of bytes to follow. The end of the file can then be detected by parsing the OK<CR><LF>.

The parameters can be used to download the telemetry file in several pieces. The number of bytes to follow will then be returned in ASCII format and terminated with <CR><LF> before the data is output. The end of telemetry stream is terminated with OK<CR><LF>. A cyclic redundancy check or a checksum will then be added to be able to verify data integrity during download. The complete file can also be downloaded in this way by specifying SA=0 and a large value for LEN. The actual file size is then returned before the data follows. See also TMSTAT for retrieving file information.

Example:
```
DOWNLOADTM,SA=0,LEN=4096,CRC=1,CKS=0
4096 (number of bytes that follow)
<binary or ASCII data>
23432 (checksum/crc value)
OK
```

4.49 Save configuration to file

Command: STOREHEADERTM
Command type: ACTION
Mode: COMMAND

This command stores the GETALL (complete configuration) to the telemetry file.

Example:

```
STOREHEADERTM
```

4.50 Erase telemetry file

Command: ERASETM
Command type: ACTION
Mode: COMMAND, RETRIEVAL

Erase the telemetry file.

Argument	Description
CODE	Code should be 9999 9999

Note: The telemetry file can also be erased over FTP.

Example:

```
ERASETM,CODE=9999
```

4.51 Write tag output

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

Write a Tag to output file and data output.

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

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

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

Example:

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

OK

4.52 Major beam angle calibration

Commands: SETUSERECHOMAJORANGLE, GETUSERECHOMAJORANGLE, GETUSERECHOMAJORANGLELIM,
Command type: CONFIGURATION
Mode: COMMAND

Commands that enable storing echosounder calibration values. The Signature echosounder does not use any of these values in its processing, but they provide users a way to store their calibration values together with the measure data. So these only serve as an option for storing calibration data in the instrument which are then output in the data file header.

These commands allow fifth order polynomials with individual enable. The scheme is to calculate the polynomial over the frequency range of the transducer, using the center frequency from the beam list as origin.

Argument	Description
EN	Enable storing calibration values
P0	Constant
P1	First order polynomial
P2	Second order polynomial
P3	Third order polynomial
P4	Forth order polynomial
P5	Fifth order polynomial

Note: Integration manual refers to this as a MajorAxis3dBBeamAngle command.

SETUSERECHOMAJORANGLE

Set calibration values for major beam angle

Example:
```
SETUSERECHOMAJORANGLE,EN=1,P0=0.123,P1=1.23,P2=12.3
```

GETUSERECHOMAJORANGLE

Get calibration values for major beam angle

GETUSERECHOMAJORANGLELIM

Get limits for calibration values for major beam angle

4.53 Major beam angle offset calibration

Commands: SETUSERECHOMAJOROFFSET, GETUSERECHOMAJOROFFSET, GETUSERECHOMAJOROFFSETLIM,
Command type: CONFIGURATION
Mode: COMMAND

Commands that enable storing echosounder calibration values. The Signature echosounder does not use any of these values in its processing, but they provide users a way to store their calibration values together with the measure data. So these only serve as an option for storing calibration data in the instrument which are then output in the data file header.

These commands allow fifth order polynomials with individual enable. The scheme is to calculate the polynomial over the frequency range of the transducer, using the center frequency from the beam list as origin.

Argument	Description
EN	Enable storing calibration values
P0	Constant
P1	First order polynomial
P2	Second order polynomial
P3	Third order polynomial
P4	Forth order polynomial
P5	Fifth order polynomial

Note: Integration manual refers to this as a MajorAxis3dBBeamAngleOffset command.

SETUSERECHOMAJOROFFSET

Set calibration values for major beam angle offset

Example:
```
SETUSERECHOMAJOROFFSET,EN=1,P0=0.123,P1=1.23,P2=12.3
```

GETUSERECHOMAJOROFFSET

Get calibration values for major beam angle offset

GETUSERECHOMAJOROFFSETLIM

Get limits for calibration values for major beam angle offset

4.54 Minor beam angle calibration

Commands: SETUSERECHOMINORANGLE, GETUSERECHOMINORANGLE, GETUSERECHOMINORANGLELIM,
Command type: CONFIGURATION
Mode: COMMAND

Commands that enable storing echosounder calibration values. The Signature echosounder does not use any of these values in its processing, but they provide users a way to store their calibration values together with the measure data. So these only serve as an option for storing calibration data in the instrument which are then output in the data file header.

These commands allow fifth order polynomials with individual enable. The scheme is to calculate the polynomial over the frequency range of the transducer, using the center frequency from the beam list as origin.

Argument	Description
EN	Enable storing calibration values
P0	Constant
P1	First order polynomial
P2	Second order polynomial
P3	Third order polynomial
P4	Forth order polynomial
P5	Fifth order polynomial

Note: Integration manual refers to this as a MinorAxis3dBBeamAngle command.

SETUSERECHOMINORANGLE

Set calibration values for minor beam angle

Example:
```
SETUSERECHOMINORANGLE,EN=1,P0=0.123,P1=1.23,P2=12.3
```

GETUSERECHOMINORANGLE

Get calibration values for minor beam angle

GETUSERECHOMINORANGLELIM

Get limits for calibration values for minor beam angle

4.55 Minor beam angle offset calibration

Commands: SETUSERECHOMINOROFFSET, GETUSERECHOMINOROFFSET, GETUSERECHOMINOROFFSETLIM,
Command type: CONFIGURATION
Mode: COMMAND

Commands that enable storing echosounder calibration values. The Signature echosounder does not use any of these values in its processing, but they provide users a way to store their calibration values together with the measure data. So these only serve as an option for storing calibration data in the instrument which are then output in the data file header.

These commands allow fifth order polynomials with individual enable. The scheme is to calculate the polynomial over the frequency range of the transducer, using the center frequency from the beam list as origin.

Argument	Description
EN	Enable storing calibration values
P0	Constant
P1	First order polynomial
P2	Second order polynomial
P3	Third order polynomial
P4	Forth order polynomial
P5	Fifth order polynomial

Note: Integration manual refers to this as a MinorAxis3dBBeamAngleOffset command.

SETUSERECHOMINOROFFSET

Set calibration values for minor beam angle offset

Example:
```
SETUSERECHOMINOROFFSET,EN=1,P0=0.123,P1=1.23,P2=12.3
```

GETUSERECHOMINOROFFSET

Get calibration values for minor beam angle offset

GETUSERECHOMINOROFFSETLIM

Get limits for calibration values for minor beam angle offset

4.56 Echosounder gain calibration

Commands: SETUSERECHOGAIN, GETUSERECHOGAIN, GETUSERECHOGAINLIM,
Command type: CONFIGURATION
Mode: COMMAND

Commands that enable storing echosounder calibration values. The Signature echosounder does not use any of these values in its processing, but they provide users a way to store their calibration values together with the measure data. So these only serve as an option for storing calibration data in the instrument which are then output in the data file header.

These commands allow fifth order polynomials with individual enable. The scheme is to calculate the polynomial over the frequency range of the transducer, using the center frequency from the beam list as origin.

Argument	**Description**
EN	Enable storing calibration values
P0	Constant
P1	First order polynomial
P2	Second order polynomial
P3	Third order polynomial
P4	Forth order polynomial
P5	Fifth order polynomial
FREQA	Calibration frequency A Unit: [Hz]
GAINFA	Specify the gain for frequency A Unit: [dB]
FREQB	Calibration frequency B Unit: [Hz]
GAINFB	Specify the gain for frequency B Unit: [dB]
FREQC	Calibration frequency C Unit: [Hz]
GAINFC	Specify the gain for frequency C Unit: [dB]

Note: Integration manual refers to this as a Gain command.

SETUSERECHOGAIN

Set calibration values for gain

Example:
`SETUSERECHOGAIN,EN=1,P0=0.123,P1=1.23,P2=12.3`

GETUSERECHOGAIN

Get calibration values for gain

GETUSERECHOGAINLIM

Get limits for calibration values for gain

4.57 Two way beam angle calibration

Commands: SETUSERECHOTWOWAYANGLE, GETUSERECHOTWOWAYANGLE, GETUSERECHOTWOWAYANGLELIM,
Command type: CONFIGURATION
Mode: COMMAND

Commands that enable storing echosounder calibration values. The Signature echosounder does not use any of these values in its processing, but they provide users a way to store their calibration values together with the measure data. So these only serve as an option for storing calibration data in the instrument which are then output in the data file header.

These commands allow fifth order polynomials with individual enable. The scheme is to calculate the polynomial over the frequency range of the transducer, using the center frequency from the beam list as origin.

Argument	**Description**
EN	Enable storing calibration values
P0	Constant
P1	First order polynomial
P2	Second order polynomial
P3	Third order polynomial
P4	Forth order polynomial
P5	Fifth order polynomial

Note: Integration manual refers to this as a TwoWayBeamAngle command.

SETUSERECHOTWOWAYANGLE

Set calibration values for two way beam angle

Example:
```
SETUSERECHOTWOWAYANGLE,EN=1,P0=0.123,P1=1.23,P2=12.3
```

GETUSERECHOTWOWAYANGLE

Get calibration values for two way beam angle

GETUSERECHOTWOWAYANGLELIM

Get limits for calibration values for two way beam angle

4.58 Precision time protocol

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

Configure the precision time protocol parameters.

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

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

PTPSET

Set precision time protocol parameters

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

```
OK
```

PTPGET

Get precision time protocol parameters

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

4.59 Wake Doppler processor

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

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.

Example:
```
BBPWAKEUP
```

4.60 Wave processing settings

Commands: SETWAVEPROC, GETWAVEPROC, GETWAVEPROCLIM,
Command type: CONFIGURATION
Mode: COMMAND

License: WaveProcessing

Wave processing settings.

Argument	Description
EN	Enable(1)/disable(0) internal wave processing.
MOUNTHEIGHT	Distance the instrument head (the pressure sensor location) is over the sea floor; this is used to correctly calculate the transfer functions used for the pressure and velocity. Unit: [m]
MODE	Processing method used in the calculation. "MLMST":

	"SUV":
BANDSEPFREQ	Frequency to separate between sea and swell. Unit: [Hz]
FREQSTART	Start frequency sets the start of the frequency range that processing is done over. Unit: [Hz]
FREQSTEP	Frequency step. Unit: [Hz]
FREQEND	End frequency sets the end of the frequency range that processing is done over. Unit: [Hz]

SETWAVEPROC

Set wave processing settings

Example:

```
SETWAVEPROC,EN=1,MOUNTHEIGHT=0.5,MODE="MLMST",BANDSEPFREQ=0.2,FREQS
TART=0.02,FREQSTEP=0.01,FREQEND=0.99
OK
```

GETWAVEPROC

Get wave processing settings

GETWAVEPROCLIM

Get wave processing setting limits

4.61 Wave telemetry settings

Commands: SETTMWAVE, GETTMWAVE, GETTMWAVELIM,
Command type: CONFIGURATION
Mode: COMMAND

License: WaveProcessing

Wave telemetry settings.

Argument	Description
EN	Enable/disable processed wave data telemetry.
DF	Telemetry data format. For examples, see Telemetry Data Format chapter. 500: Binary data format

	501: Wave NMEA format 502: Legacy binary data format
TWAP	Enable output of wave parameter estimates
TEDS	Enable output of Energy Density Spectrum
TWBS	Enable output of Wave band estimates
TFS	Enable output of Fourier coefficient spectrum
SO	Enable Serial Output
FO	Enable File Output
TWDR	Enable Transmit Wave Direction

SETTMWAVE

Set wave telemetry settings

Example:
```
SETTMWAVE,EN=1,DF=501,TWAP=1,TEDS=0,TWBS=0,TFS=0,SO=1,FO=0,TWDR=0
OK
```

GETTMWAVE

Get wave telemetry settings

GETTMWAVELIM

Get wave telemetry setting limits

4.62 Get hardware specifications

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

Returns Firmware versions and Board revisions.

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

INTERFACE	Board revision. Example: C-0
ANALOG	Board revision. Example: C-0
ANALOG2	Board 2 revision. Example: C-0
SENSOR	Board revision. Example: C-0

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

4.63 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
```

4.64 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
```

4.65 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,"4X218TRTRPNUB","High Resolution",4
LISTLICENSE,"JKHHFNH3RPNUB","Wave Mode",6
LISTLICENSE,"WF3CJR6PRPNUB","Current Profiler",1
OK
$PNOR,LISTLICENSE*76
$PNOR,LISTLICENSE,KEY="4X218TRTRPNUB",DESC="High
Resolution",TYPE=4*73
$PNOR,LISTLICENSE,KEY="JKHHFNH3RPNUB",DESC="Wave Mode",TYPE=6*00
$PNOR,LISTLICENSE,KEY="WF3CJR6PRPNUB",DESC="Current
Profiler",TYPE=1*1C
$PNOR,OK*2B
```

5 GETALL

The GETALL command retrieves all configuration settings for the instrument and can output this information to the command line or a data file. Its response, commonly referred to as the header, is required to generate a valid .ad2cp file compatible with Nortek post-processing software. Note that the GETALL header information must match the structure and content of the subsequent data for the file to be interpreted correctly.

Argument	Description
FN	Write the output to this file

Example: The following output is from a GETALL command executed on a Signature 1000 using the default Average measurement configuration plan. Different configurations - such as enabling Burst measurements - will result in additional or modified parameters in the GETALL response. For detailed explanations of each parameter, refer to the relevant sections in the Commands chapter. Note that not all parameters are listed there; factory-set calibration parameters are excluded, as they cannot be modified.

Parameter	Example	Description
GETCLOCKSTR	TIME="2021-06-08 16:12:44"	Current time on the instrument real-time clock
ID	STR="Signature1000",SN=900002	Instrument name and serial number
GETHW	FW=2214,FPGA=185,DIGITAL="H-0",INTERFACE="D-1",ANALOG="C-4",SENSOR="C-0(AHRS)",BOOT=12,FWMINOR=12	Firmware and electronics revision numbers
BOARDSENSGET	AV=23,NB=5,HF=1000,TTR=2.0000,TTRB5=2.0000,TTRB5AUX=0.0000,AUXRS=0	Production information
GETPWR	PLAN=53.51,BURST=0.00,AVG=50.78,PLAN1=0.00,BURST1=0.00,AVG1=0.00,TOTAL=53.51	Power consumption in mW for current plan
GETMEM	PLAN=0.388,BURST=0.000,AVG=0.388,PLAN1=0.000,BURST1=0.000,AVG1=0.000,TOTAL=0.388	Memory consumption in MB for current plan
GETPRECISION	AVGHORZ=0.95,BURSTHORZ=-9.99,BEAM5=-9.99,AVGBEAM=0.57,BURSTBEAM=-9.99	Horizontal and along-beam precision values
GETPLAN	MIAVG=600,AVG=1,DIAVG=0,VD=0,MV=10,SA=35.0,BURST=0,MIBURST=3600,DIBURST=0,SV=0.00,FN="Data.ad2cp",SO=0,FREQ=1000,NSTT=0	Current PLAN settings
GETAVG	NC=60,CS=0.50,BD=0.10,CY="BEAM",PL=0.0,AI=60,VP=0.000,VR=2.50,DF=3,NPING=60,NB=4,CH=0,MUX=0,BW="BROAD",ALTI=0,BT=0,ICE=0,ALTISTART=0.10,ALTIEND=30.00,RAWALTI=0	Current AVG settings

GETXFAVG	ROWS=4,COLS=4,M11=1.1831,M12=0.0000,M13=-1.1831,M14=0.0000,M21=0.0000,M22=-1.1831,M23=0.0000,M24=1.1831,M31=0.5518,M32=0.0000,M33=0.5518,M34=0.0000,M41=0.0000,M42=0.5518,M43=0.0000,M44=0.5518	Transformation matrix values
GETUSER	POFF=9.50,DECL=0.00,HX=0,HY=0,HZ=0	Pressure and compass offset values
GETINST	BR=9600,RS=422,LED="ON24H",ORIENT="AHRS3D",CMTOUT=300,DMTOUT=60,CFMTOUT=60	Instrument settings
GETCOMPASSCAL	DX=247,DY=138,DZ=-40,M11=32418,M12=-935,M13=-7,M21=1229,M22=32767,M23=-213,M31=-979,M32=1367,M33=31081	Current compass calibration values
READAHRS	STR="OSv6m1_ng2_1.0.0.2 Oct 29 2019, SerialNumber=60000371,type=OS3DM"	If present; AHRS name and firmware version
RECSTAT	SS=512,CS=32768,FC=4002381824,TC=4040163328,VS=4040163328,FF=4082,TF=4095	Recorder status
BEAMCFGLIST	BEAM=1,THETA=25.00,PHI=0.00,FREQ=1000,BW=25,BRD=1,HWBEAM=1,ZNOM=60.00,DIA=0.0	Instrument beam configuration
	BEAM=2,THETA=25.00,PHI=-90.00,FREQ=1000,BW=25,BRD=1,HWBEAM=2,ZNOM=60.00,DIA=0.0	
	BEAM=3,THETA=25.00,PHI=180.00,FREQ=1000,BW=25,BRD=1,HWBEAM=3,ZNOM=60.00,DIA=0.0	
	BEAM=4,THETA=25.00,PHI=90.00,FREQ=1000,BW=25,BRD=1,HWBEAM=4,ZNOM=60.00,DIA=0.0	
	BEAM=5,THETA=0.00,PHI=0.00,FREQ=1000,BW=25,BRD=1,HWBEAM=5,ZNOM=60.00,DIA=0.0	
BEAMIMPLIST	BEAM=1,P0=1.00000e+02,P1=0.00000e+00,P2=0.00000e+00,P3=0.00000e+00,P4=0.00000e+00,T1=0.00000e+00	Instrument beam configuration
	BEAM=2,P0=1.00000e+02,P1=0.00000e+00,P2=0.00000e+00,P3=0.00000e+00,P4=0.00000e+00,T1=0.00000e+00	
	BEAM=3,P0=1.00000e+02,P1=0.00000e+00,P2=0.00000e+00,P3=0.00000e+00,P4=0.00000e+00,T1=0.00000e+00	
	BEAM=4,P0=0.00000e+00,P1=0.00000e+00,P2=0.00000e+00,P3=0.00000e+00,P4=0.00000e+00,T1=0.00000e+00	
	BEAM=5,P0=0.00000e+00,P1=0.00000e+00,P2=0.00000e+00,P3=0.00000e+00,P4=0.00000e+00,T1=	

	0.00000e+00	
LISTLICENSE	KEY="D1EWY135MB7K7",DESC="Burst Mode",TYPE=2	Installed instrument licences
	KEY="E39YR4H2MB7K7",DESC="Averaging Mode",TYPE=1	
	KEY="99BHT6KRMB7K7",DESC="16GB Recorder",TYPE=11	
CALCOMPGET	DX=247,DY=138,DZ=-40,M11=32418,M12=-935,M13=-7,M21=1229,M22=32767,M23=-213,M31=-979,M32=1367,M33=31081	Compass calibration values
CALTEMPGET	SC=1.00000	Water temperature calibration value
CALTILTGET	PO=0.00,RO=0.00,MAGG=1,HO=0.00	Tilt sensor calibration values
CALACCLGET	AX=1.000000E+00,B0X=-5.838510E-04,B1X=4.038870E-03,B2X=1.900822E-03,B3X=0.000000E+00,A1X=-3.905594E-05,A2X=-2.741814E-06,A3X=0.000000E+00	Accelerometer calibration values
	AY=1.000000E+00,B0Y=-8.213488E-03,B1Y=9.444051E-03,B2Y=1.928553E-03,B3Y=0.000000E+00,A1Y=-3.572553E-05,A2Y=2.868473E-06,A3Y=0.000000E+00	
	AZ=1.000000E+00,B0Z=1.251627E-02,B1Z=8.488446E-05,B2Z=1.951933E-03,B3Z=0.000000E+00,A1Z=8.686632E-05,A2Z=2.995133E-06,A3Z=0.000000E+00	
CALGYROGET	AX=1.092663E+00,B0X=-1.458604E-02,B1X=-1.925275E-03,B2X=-5.933643E-05,B3X=0.000000E+00,A1X=1.739189E-04,A2X=-7.793307E-06,A3X=0.000000E+00	Gyro calibration values
	AY=1.086592E+00,B0Y=1.325879E-02,B1Y=4.371032E-04,B2Y=3.761053E-05,B3Y=0.000000E+00,A1Y=-2.659187E-04,A2Y=2.339482E-06,A3Y=0.000000E+00	
	AZ=1.087252E+00,B0Z=-4.356480E-02,B1Z=5.960464E-08,B2Z=1.316518E-05,B3Z=0.000000E+00,A1Z=4.348904E-05,A2Z=2.026558E-06,A3Z=0.000000E+00	
CALPRESSGET	MT=1,RREF=4.5300000000e+02,RB0=0.0000000000e+00,RB1=0.0000000000e+00,RB2=0.0000000000e+00,RB3=0.0000000000e+00,T0=-3.7026600000e+03,T1=2.7309500000e+00,T2=-	Pressure sensor calibration values

	6.8649568750e-04,T3=5.9816006250e-08,ID="L0165"	
CALPRESSCOEFFGET	A0=5.4010770000e+01,A1=-4.2163353130e-02,A2=1.1131470000e-05,A3=-9.9709175000e-10,B0=-3.1236960000e+00,B1=2.9407478130e-03,B2=-7.7748162500e-07,B3=6.8127600000e-11	Pressure sensor calibration coefficient values
	C0=-7.5628700000e-03,C1=6.2269425000e-06,C2=-1.6925208750e-09,C3=1.5250131250e-13,D0=4.1404246880e-05,D1=-3.4084846880e-08,D2=9.3360593750e-12,D3=-8.5032206250e-16	
CALROTACCLGET	M11=0.99685,M12=-0.00455,M13=0.01998,M21=0.00674,M22=0.99572,M23=-0.00375,M31=-0.02500,M32=0.00529,M33=0.99523	Accelerometer rotation matrix values
CALROTGYROGET	M11=1.00000,M12=-0.01459,M13=-0.00088,M21=0.00841,M22=1.00000,M23=-0.01623,M31=-0.02101,M32=0.00059,M33=1.00000	Gyro rotation matrix values
CALECHOGET	CHA0=0.00,CHB0=0.00,CHC0=0.00	Echosounder calibration values

6 User Cases

6.1 Checking instrument state over Ethernet

In this example a user connects to and powers the Ethernet port, but is unsure of the current operational state. If power is applied while in measurement mode, it will continue the measurement but not wake the Ethernet processor (BBP). If power is applied while in deployment state a re-synch will occur and resume sleep mode. Hence it is necessary to use **BBPWAKEUP** in both cases.

A typical sequence starts by wanting to know the state of the instrument before proceeding with either a new measurement or data retrieval.

```
% Waking up the BBP to make sure commands are received
BBPWAKEUP
$PNOR,OK*2B

% Inquiring the state the of the instrument
GETSTATE
$PNOR,GETSTATE,MODE=0010,DEPTIME=27521,MEASTIME=27521,CURRTIME="2015-09-28
11:21:16",WAKEUP=2,INTPROC=0*01
$PNOR,OK*2B
```

This indicates the instrument has been configured to deploy and has started its scheduled deployment for 27521 seconds. See GETSTATE for more information.

Depending on the desired action, send Break usually followed by; either MC to enter command mode, RM for data retrieval or START/DEPLOY/CO to start/schedule/continue a deployment.

6.2 Download telemetry file via FTP

In this example, a Signature1000 is set up to measure currents for 2 minutes every 10 minutes and waves every hour (4096 samples at 4 Hz). The raw current data are processed and a subset is saved as a telemetry file and made available on FTP.

Configuration example:

```
% Recommended starting point for configuration file
SETDEFAULT,CONFIG
$PNOR,OK*2B

% Configuration for instrument:
SETPLAN,MIAVG=600,AVG=1,DIAVG=0,VD=0,MV=10,SA=35,BURST=1,MIBURST=3600,DIBUR
ST=0,SV=0,FN="Ex3.ad2cp",SO=0,FREQ=1000
$PNOR,OK*2B
SETAVG,NC=21,CS=1,BD=0.2,CY="ENU",PL=0,AI=120,VR=2.5,DF=3,NPING=13,NB=4,CH=
0,MUX=0,BW="BROAD",ALTI=0,BT=0,ICE=0,ALTISTART=1,ALTIEND=30
$PNOR,OK*2B
SETBURST,NC=13,NB=4,CS=1,BD=9.5,CY="BEAM",PL=0,SR=4,NS=4096,VR=2.5,DF=3,NPI
```

```
NG=1,CH=0,VR5=2.5,ALTI=1,BT=0,DISV=0,RAWALTI=1,ALTISTART=4.8,ALTIEND=33.1
$PNOR,OK*2B

% Configuration for telemetry file:
SETTMAVG,EN=1,CD=2,PD=1,AVG=120,TV=1,TA=1,TC=1,CY="ENU",FO=1,SO=0,DF=100
$PNOR,OK*2B
SAVE,CONFIG
$PNOR,OK*2B
DEPLOY,TIME="2014-11-18 14:40:00"
$PNOR,OK*2B
```

Go to *ftp://your-IP-address* to find the telemetry file (telemetryfile.bin). Here is part of the result from the above configuration. Note that the data were collected in air.

```
$PNORI,4,Signature1000900002,4,11,0.20,1.00,0*1B
$PNORS,091715,143440,00000000,2A4C0000,14.3,1300.0,278.3,15.7,-33.0,0.000,-
262.45,0,0*65
$PNORC,091715,142440,1,0.24,-1.35,-2.21,-
1.69,1.37,169.7,C,79,84,67,102,11,13,8,11*2B
$PNORC,091715,142440,3,0.64,-0.28,-1.91,-
1.32,0.70,113.9,C,79,84,66,96,12,14,7,20*13
$PNORC,091715,142440,5,0.08,-0.50,-1.76,-
1.48,0.51,171.2,C,78,84,66,92,11,13,7,24*1D
$PNORC,091715,142440,7,-0.37,0.97,-1.02,-
1.07,1.04,339.0,C,78,84,66,67,11,14,10,10*21
$PNORC,091715,142440,9,-0.94,0.57,-0.76,-
1.11,1.10,301.1,C,78,83,65,69,12,15,9,10*10
$PNORC,091715,142440,11,-0.37,0.76,-0.95,-
1.06,0.85,334.0,C,78,83,65,66,13,15,8,8*14
$PNORC,091715,142440,13,0.05,-0.25,-1.64,-
1.36,0.26,168.4,C,78,84,66,82,11,14,9,33*2F
$PNORC,091715,142440,15,-0.20,0.20,-1.36,-
1.32,0.28,314.6,C,78,84,66,67,11,13,9,7*16
$PNORC,091715,142440,17,0.19,0.17,-1.47,-
1.13,0.25,48.0,C,78,84,65,69,12,16,9,2*0D
$PNORC,091715,142440,19,-0.91,0.45,-0.90,-
1.19,1.02,296.5,C,78,84,65,66,12,14,10,8*27
$PNORC,091715,142440,21,-0.49,0.66,-1.00,-
1.11,0.82,323.1,C,78,84,65,67,12,14,11,10*13
```

After downloading the telemetry file, erase it either via FTP or commands. Only the telemetry file can be deleted using FTP.

```
% Erasing telemetry file
ERASETM,9999
$PNOR,OK*2B

% Continuing the configured deployment plan
```

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

For use with external controller it can be interesting to note the following: if the instrument is started at e.g. 12:00, the first current profile is finished at 12:02 (120 seconds) and the next starts about 12:10. That leaves us with 8 minutes to download the telemetry file to FTP before next current profile starts. The clock drifts with about 1 sec/week. Since DEPLOY was used the measurement intervals will resynchronize according to the deployment time and the instrument configuration (see DEPLOY for more information), thus is should be easier to schedule automatic data download as the window 12:02 to 12:10 remains.

6.3 Download telemetry file over serial port

In this example the user wishes to download the telemetry file in 4096 byte chunks. To achieve this you must connect via the Terminal Emulator or other console while the instrument is measuring.

```
% Send Break
CONFIRM
OK

%Going into Data Retrieval Mode
RM
NORTEK AS.
Version 2214_12 (Apr 15 2021 07:31:19)
DATA RETRIEVAL MODE
$PNOR,OK*2B

% Checking the size of the telemetry file. Return in bytes
TMSTAT
95558
$PNOR,OK*2B

% Outputting the telemetry file over serial port in 4096 byte chunks
DOWNLOADTM,0,4096,CKS=1
[OUTPUT]
$PNOR,OK*2B

% Next 4096 byte chunk, etc
DOWNLOADTM,4097,4096,CKS=1
[OUTPUT]
$PNOR,OK*2B

% Erasing telemetry file
ERASETM,9999
$PNOR,OK*2B

% Continuing the configured deployment plan
```

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

Copy the returned text and paste to file, or check "Record to file" before retrieving the file and the file will appear by default in: C:\Users\xxxx\Documents\Nortek\Deployment\Online.

Parameters can be added to the DOWNLOADTM command to set start address, length of file, etc. (see section DOWNLOADTM).

6.4 View memory and power requirements, Signature100

If you create a custom deployment outside of the Deployment Wizard, you will not be able to open it in the software. This means that you will not be able to see the power and memory requirements of your plan in the summary pane. However, once you have configured the instrument, you can use commands to retrieve the requirements for the plan you have chosen.

```
% Retrieve power requirements for the default Signature100 plan (mWatts)
GETPWR
$PNOR,GETPWR,PLAN=268.61,BURST=0.00,AVG=266.94,PLAN1=0.00,BURST1=0.00,AVG1=
0.00,TOTAL=268.61*5F
$PNOR,OK*2B

% Retrieve memory requirements for the default Signature100 plan
(Mbytes/hour)
GETMEM
$PNOR,GETMEM,PLAN=0.103,BURST=0.000,AVG=0.103,PLAN1=0.000,BURST1=0.000,AVG1
=0.000,TOTAL=0.103*72
$PNOR,OK*2B
```

6.5 Average velocity data and NMEA, Signature 55

Either use the Deployment wizard to create a .deploy file which can be uploaded via the Terminal Emulator, or set the configuration through commands (seen below). The .deploy file can also be uploaded then customized via commands once in the Terminal Emulator.

In this example: Signature55, configured to alternate between fine and coarse current profiles (3:1). In this case the user wanted to download the averaged fine profile upon request.

Configuration example:

```
% Recommended starting point for configuration file
SETDEFAULT,CONFIG
$PNOR,OK*2B

% Setting plan for "Fine" profile
SETPLAN,MIAVG=600,AVG=1,DIAVG=0,VD=0,MV=10,SA=35,BURST=0,MIBURST=120,DIBURS
T=0,SV=0,FN="Data.ad2cp",SO=0,FREQ=75
$PNOR,OK*2B
SETAVG,NC=109,CS=5,BD=2,CY="ENU",PL=-
```

```
6,AI=180,VR=1,DF=3,NPING=137,NB=3,CH=0,MUX=0,BW="BROAD",ALTI=0,BT=0,ICE=0
$PNOR,OK*2B

% Setting plan for "Coarse" profile
SETPLAN1,MIAVG=1800,AVG=1,DIAVG=0,VD=0,MV=10,SA=35,BURST=0,MIBURST=120,DIBU
RST=0,SV=0,FN="Data.ad2cp",SO=0,FREQ=55
$PNOR,OK*2B
SETAVG1,NC=54,CS=20,BD=2,CY="ENU",PL=-
2,AI=180,VR=1,DF=3,NPING=60,NB=3,CH=0,MUX=1,BW="NARROW",ALTI=0,BT=0,ICE=0
$PNOR,OK*2B

% Setting the alternating measurement intervals and ratios of "Fine" and
"Coarse"
SETALTERNATE,EN=1,PLAN=1380,IDLE=10,PLAN1=180,IDLE1=230
$PNOR,OK*2B

% Setting the telemetry file to average the "Fine" profile over the
averaging interval
SETTMAVG,EN=1,CD=1,PD=1,AVG=180,TV=1,TA=1,TC=1,CY="ENU",FO=1,SO=0,DF=100
$PNOR,OK*2B
SAVE,CONFIG
$PNOR,ERROR*77

% Finding where the error in the configuration is
GETERROR
$PNOR,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])"*6E
$PNOR,OK*2B

% Number of pings too high compared to desired averaging interval with
multiplex enabled.
SETAVG1,NPING=30
$PNOR,OK*2B
SAVE,CONFIG
$PNOR,OK*2B
```

Note that SETTMAVG,AVG must equal the AI set by SETAVG,AI. To set telemetry averaging for the alternate plan use SETTMAVG1, note that these will be recorded to the same telemetryfile.bin file.

Enter START or DEPLOY,TIME to begin the deployment.

6.6 Disable a beam, Signature500

Objective: disable a beam in the Burst measurements that you know will be blocked by an object when the instrument is deployed.

```
% Upload prepared .deploy file to the instrument via the Terminal Window.
The file is read in line-by-line.
$PNOR,SETDEFAULT,ALL*4B
$PNOR,SETPLAN,MIAVG=60,AVG=1,DIAVG=0,VD=0,MV=10,SA=35,BURST=1,MIBURST=1200,
DIBURST=0,SV=0,FN="Data.ad2cp",SO=0,FREQ=1000,NSTT=0*42
$PNOR,SETAVG,NC=21,CS=1,BD=0.2,CY="ENU",PL=0,AI=10,VR=2.5,DF=3,NPING=13,NB=
4,CH=0,MUX=0,BW="BROAD",ALTI=0,BT=0,ICE=0,ALTISTART=0.1,ALTIEND=30,RAWALTI=
1*0B
$PNOR,SETBURST,NC=13,NB=4,CS=1,BD=9.5,CY="BEAM",PL=0,SR=4,NS=4096,VR=2.5,DF
=3,NPING=1,CH=0,VR5=2.5,ALTI=1,BT=0,DISV=0,ECHO=0,RAWALTI=1,ALTISTART=6.9,A
LTIEND=33.1,HR=0,HR5=0*3C
$PNOR,SETTMAVG,EN=0,CD=1,PD=1,AVG=10,TV=1,TA=1,TC=1,CY="ENU",FO=0,SO=1,DF=1
00,DISTILT=0,TPG=0,MAPBINS=0*6D
$PNOR,SAVE,ALL*43
$PNOR,OK*2B
$PNOR,OK*2B
$PNOR,OK*2B
$PNOR,OK*2B
$PNOR,OK*2B
$PNOR,OK*2B

% Disable averaging and enable burst measurements.
SETPLAN,AVG=0,BURST=1
$PNOR,OK*2B

% Configure to measure with just 3 beams, and specify which ones to use.
SETBURST,NB=3,CH=234
$PNOR,OK*2B

% Save the configuration.
SAVE,ALL*43
$PNOR,OK*2B
```

6.7 Average telemetry, Signature1000

Objective: send averaged velocity data over the serial line, and save the telemetry file to the instrument recorder for retrieval later in case of communications failure. Deploy the instrument to start at a later time. In this case, NMEA format in the terminal window is enabled.

```
% Upload prepared .deploy file to the instrument via the Terminal Window.
The file is read in line-by-line.
$PNOR,SETDEFAULT,ALL*4B
$PNOR,SETPLAN,MIAVG=60,AVG=1,DIAVG=0,VD=0,MV=10,SA=35,BURST=1,MIBURST=1200,
DIBURST=0,SV=0,FN="Data.ad2cp",SO=0,FREQ=1000,NSTT=0*42
$PNOR,SETAVG,NC=21,CS=1,BD=0.2,CY="ENU",PL=0,AI=10,VR=2.5,DF=3,NPING=13,NB=
4,CH=0,MUX=0,BW="BROAD",ALTI=0,BT=0,ICE=0,ALTISTART=0.1,ALTIEND=30,RAWALTI=
1*0B
```

```
$PNOR,SETBURST,NC=13,NB=4,CS=1,BD=9.5,CY="BEAM",PL=0,SR=4,NS=4096,VR=2.5,DF
=3,NPING=1,CH=0,VR5=2.5,ALTI=1,BT=0,DISV=0,ECHO=0,RAWALTI=1,ALTISTART=6.9,A
LTIEND=33.1,HR=0,HR5=0*3C
$PNOR,SETTMAVG,EN=0,CD=1,PD=1,AVG=10,TV=1,TA=1,TC=1,CY="ENU",FO=0,SO=1,DF=1
00,DISTILT=0,TPG=0,MAPBINS=0*6D
$PNOR,SAVE,ALL*43
$PNOR,OK*2B
$PNOR,OK*2B
$PNOR,OK*2B
$PNOR,OK*2B
$PNOR,OK*2B
$PNOR,OK*2B

% Modify SETTMAVG line; enable telemetry, enable file output. Serial output
is already enabled (SO=1).
SETTMAVG,EN=1,FO=1*26
$PNOR,OK*2B

% Save the configuration.
SAVE,ALL*43
$PNOR,OK*2B

% Deploy the instrument and specify a time.
DEPLOY,TIME="2021-06-18 13:00:00"*0C
$PNOR,OK*2B

% The instrument is now measuring, and you will see data start to come
through the connection.
```

7 Binary data formats

This chapter describes the binary data formats used by the Signatures. With the Signature instruments you get binary data stored to two files on the recorder:

[filename]_avgd.ad2cp - contains averaged current data for the Signature. The data is averaged over the configured average interval and saved as Data Format 3. Note that some internal quality control has been applied to this data. For more information about the data format and the internal quality control, please refer to the Quality Control chapter. If the onboard wave processing has been enabled, the processed wave data can also be found in this file.

[filename].ad2cp contains the raw single ping data, both for the average currents and burst data such as raw wave data, high resolution measurements or echosounder data. The data is saved as Data Format 3.

The data formats themselves are described in the following chapters. Note that the binary data formats all use a common header that specifies how the rest of the data block should be interpreted, containing information about data type, size etc. The header is documented separately as _HeaderData. A data block is the data from and including one header to the next, and can for example contain data from one velocity profile. We use the leading underscore in the chapter title to emphasize that this is not a complete data set, but is a part used by two or more other data formats. Note that for Data Format 3 we also have two other common data fields shared by several data formats (see illustration in the table below).

Binary data is always outputted as Little Endian.

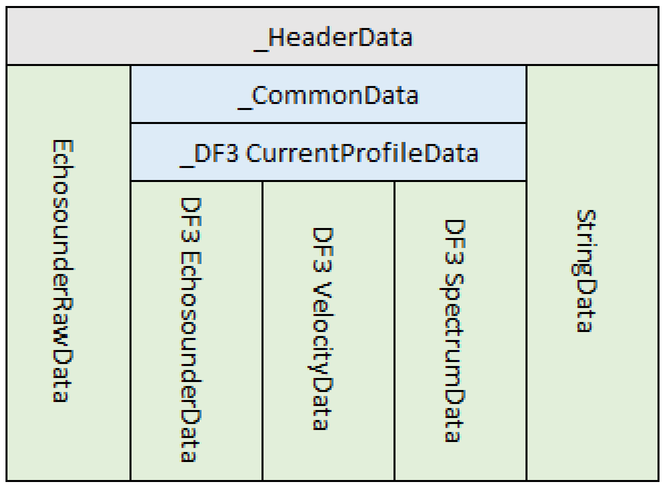


Figure: Showing how common data fields (gray for header and blue for other common's) relate to the sensor specific data fields (green). Note that we use a leading underscore (_) to emphasize that this is not a sensor data format but is common and used by two or more data formats.

About the tables

The tables have the columns **Field**, **Position/Size** and **Description**. While Field and Description provide the name and additional details about a specific field, **Position** and **Size** are described in more detail below:

Position indicates the location of a field in the data block. This refers to the byte position, counting from zero at the first byte. Note that there are two separate counts for the header, and the data following the header, so the count starts from zero again after the header. For instance, the Data Series ID is at position 2 in the header of the data block. The Serial number (see _CommonData) has position 4 in the data following the header.

Some positions are not fixed but depend on the fields in front of them. In these cases, Offset of Data (position 1 of the data following the header, see _CommonData) can be used to determine the position of the following fields. When this is relevant, 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 refers to the data type of the field. For example, the Data Series ID in the header is an unsigned integer of 8 bits (uint8). Note that not all fields have a specific data type; some are objects requiring a certain number of bits. For instance, the status bit mask has a size of 32 bits. For fields where the size is given in bits, a more detailed description of the field and each of the bits is listed separately below the table where they are used.

7.1 _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. 0x15 - Burst data as DF3. 0x16 - Average data as DF3. 0x17 - Bottom Track Data Record. 0x18 - Interleaved Burst Data Record (beam 5). 0x1E - Altimeter Record. 0x1F - Avg Altimeter Raw Record. 0x1A - Burst Altimeter Raw Record. 0x1B - DVL Bottom Track Record. 0x1C - Echo Sounder Record. 0x23 - Echo Sounder Raw Record. 0x24 - Echo Sounder Raw Tx Record. 0x26 - Average data as DF7. 0x30 - Processed Wave Data Record. 0x1D - DVL Water Track Record. 0xC8 - Vector 2 data as DF8. 0xA0 - String Data Record, eg. GPS NMEA data, comment from the FWRITE command. 0x20 - Spectrum data as DF3.
Family id	3 uint8	Defines the Instrument Family. 0x10 is the Signature Family. 0x16 is the DVL family. 0x30 is the Aquadopp Generation 2 family. 0x40 is the Awac Generation 2 family.
Data size	4 unit16/uint32	Number of bytes in the following Data Record. If header size is 10, the data size is represented with a

		uint16. For large datasets, header may have 12 bytes giving room for a uint32 to represent data size.
Data checksum	6/8 uint16	Checksum of the following Data Record.
Header checksum	8/10 uint16	Checksum of all fields of the Header except the Header Checksum itself.

7.2 _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 µs]
Speed of sound	16 uint16	Speed of sound used by the instrument. Raw data given as 0.1 m/s Unit: [m/s]
Temperature	18 int16	Reading from the temperature sensor. Raw data given as 0.01 °C Unit: [°C]

Pressure	20 uint32	Raw data given as 0.001 dBar Unit: [dBar]
Heading	24 uint16	Raw data given as 0.01 degrees Unit: [deg]
Pitch	26 int16	Raw data given as 0.01 degrees Unit: [deg]
Roll	28 int16	Raw data given as 0.01 degrees Unit: [deg]
Cell size	32 uint16	Size of each cell (resolution) on the beam. Raw data given as mm Unit: [m]
Nominal correlation	36 uint8	The nominal correlation for the configured combination of cell size and velocity range Unit: [%]
Battery voltage	38 uint16	Raw value given in 0.1 Volt Unit: [V]
Magnetometer.X	40 int16	X axis flux raw value in last measurement interval
Magnetometer.Y	42 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 wether the ambiguity velocity (in DF3 CurrentProfileData) is 16bit or 32bit long.
56/58	The status field is at 56 or 58 depending on wether the ambiguity velocity (in DF3 CurrentProfileData) is 16bit or 32bit long.
58/60	The status field is at 56 or 58 depending on wether the ambiguity velocity (in DF3 CurrentProfileData) is 16bit or 32bit long.
59/61	The status field is at 59 or 61 depending on wether the ambiguity velocity (in DF3 CurrentProfileData) is 16bit or 32bit long.
60/62	The status field is at 60 or 62 depending on wether the ambiguity velocity (in DF3 CurrentProfileData) is 16bit or 32bit long.
62/64	The status field is at 62 or 64 depending on wether the ambiguity velocity (in DF3 CurrentProfileData) is 16bit or 32bit long.
64/66	The status field is at 64 or 66 depending on wether the ambiguity velocity (in DF3 CurrentProfileData) is 16bit or 32bit long.
72/74	The status field is at 72 or 74 depending on wether 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)

7.3 _DF3 CurrentProfileData

Extends: _CommonData
Used By: DF3 EchosounderData, DF3 VelocityData, DF3 SpectrumData

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

Field	Position	Description

	Size	
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

7.4 DF3 EchosounderData

Extends: _DF3 CurrentProfileData
ID: 0x1c

Data definitions for parsing SIGNATURE V3 echosounder data.

Field	**Position** **Size**	**Description**
Number of cells	30 uint16	Number of echosounder cells
Echosounder frequency	52 uint16	Echosounder frequency Unit: [100 Hz]
Echosounder data	OFFSET int16 * NC	Echosounder amplitude Data Raw data given as 0.01 dB/count Unit: [dB/count]

Position and size variables:

Name	**Description**

NC	Number of echosounder cells given at position 30 in this dataset.
OFFSET	Offset of data given at position 1 in this dataset. Number of bytes from start of record to start of data.

Note: There is no information in the data format on whether pulse compression is used to process the echosounder data. This must be retrieved from GETALL on the instrument or from the GETALL data string found first in a .signature file.

7.5 EchosounderRawData

ID: 0x23, 0x24

This chapter describes the Data Format used to store Echo Sounder RAW. The Echo Sounder RAW format uses the following Record IDs to store data:

Id 0x23 for the Echo Sounder raw sample data format.

Id 0x24 for Echo Sounder raw synthetic transmit pulse data format.

Field	Position Size	Description
Version	0 uint8	Version number of this Data Record Definition. Should be 1.
Offset of data	1 uint8	Number of bytes from start of record to start of non-common data fields. Unit: [# bytes]
Year	2 uint8	Number of years since 1900.
Month	3 uint8	Month number counting from 0 which is January.
Day	4 uint8	Day of the month
Hour	5 uint8	24 hour of the day
Minutes	6 uint8	Minutes.
Seconds	7 uint8	Seconds.
Hundred micro seconds	8 uint16	Hundred micro seconds from last whole second. Unit: [100 µs]
Error status	10 2 * 8 bits	Error bit mask. Object reference given in table below
Status	12	Status bit mask.

	4 * 8 bits	Object reference given in table below
Serial number	16 uint32	Instrument serial number from factory.
Number of samples	20 uint32	Number of following complex samples.
Start sample index	24 uint32	Sample number where the position is equal to the configured blanking distance. Value is calculated with a nominal sound speed of 1500m/s.
Sampling rate	28 float	Raw data sampling rate. Unit: [Hz]
Echosounder raw data	OFFSET int32 * SAMPLES * 2	The Data consists of an array of length as given above in position 20 (Number of samples). Each Sample is of type Complex Fract32 with a real and an imaginary part. typedef struct { fract32 re; fract32 im; } complex_fract32; Fract32 is a signed 32bit data format used to represent values [-1, +1>. Fract32 values range from -2^31 to +2^31 -1. For converting, you take the fract32 as a signed long (32-bit), cast it to float and then divide it with 2^31.

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	Measurement not finished

		The Measurement and data storage/transmit didn't finish before next measurement started.
Sensor read failure	5 bit	Sensor read failure
Tag error beam 1 (In-phase)	8 bit	Tag error beam 1 (In-phase)
Tag error beam 1 (Quadrature-phase)	9 bit	Tag error beam 1 (Quadrature-phase)
Tag error beam 2 (In-phase)	10 bit	Tag error beam 2 (In-phase)
Tag error beam 2 (Quadrature-phase)	11 bit	Tag error beam 2 (Quadrature-phase)
Tag error beam 3 (In-phase)	12 bit	Tag error beam 3 (In-phase)
Tag error beam 3 (Quadrature-phase)	13 bit	Tag error beam 3 (Quadrature-phase)
Tag error beam 4 (In-Phase)	14 bit	Tag error beam 4 (In-phase)
Tag error beam 4 (Quadrature-phase)	15 bit	Tag Error Beam 4 (Quadrature-phase)

Object reference: Status

Status bit mask.

Field	**Position Size**	**Description**
Wake up state	0 uint32	Bit 31-28: Wakeup State 1111 0000 0000 0000 0000 0000 0000 0000 00 = bad power 01 = power applied 10 = break 11 = RTC alarm
Orientation	0 uint32	Bit 27-25: Orientation 0000 1110 0000 0000 0000 0000 0000 0000 4: "UP" 5: "DOWN" 7: "AHRS"

Auto orientation	0 uint32	Bit 24-22: autoOrientation 0000 0001 1100 0000 0000 0000 0000 0000 0: "Fixed" Fixed orientation 1: "Auto" Auto Up Down 2: "Auto3D" 3: "AHRS3D" AHRS3D
Previous Wake up state	0 uint32	Bit 27-18: Previous Wakeup State 0000 0000 0011 1100 0000 0000 0000 0000 0 = bad power 1 = power applied 2 = break 3 = RTC alarm
Last measurement low voltage skip	0 uint32	Bit 17: Last measurement low voltage skip 0000 0000 0000 0010 0000 0000 0000 0000 0 = not skipped 1 = skipped due to low voltage
Active configuration	0 uint32	Bit 16: Active configuration 0000 0000 0000 0001 0000 0000 0000 0000 0 = first configuration 1 = alternate configuration
Echo Index	0 uint32	Bit 15-12: Echo Index 0000 0000 0000 0000 1111 0000 0000 0000 0 = FREQ1 1 = FREQ2 2 = FREQ3
Telemetry Data	0 uint32	Bit 11: Telemetry Data 0000 0000 0000 0000 0000 1000 0000 0000 0 = regular data 1 = telemetry data (i.e. SETTM* commands)
Boost Running	0 uint32	Bit 10: Boost Running 0000 0000 0000 0000 0000 0100 0000 0000 0 = not running during the battery measurement 1 = running during the battery measurement
Echo Frequency Bin	0 uint32	Bit 9-5: Echo Frequency Bin 0000 0000 0000 0000 0000 0011 1110 0000 Echogram bin number when using binned frequency response
BD Scaling	0 uint32	Bit 1: BD Scaling 0000 0000 0000 0000 0000 0000 0000 0010 Hard coded to 1, indicating cm scaling of blanking distance

7.6 DF3 VelocityData

Extends: _DF3 CurrentProfileData
ID: 0x15, 0x16, 0x18, 0x1e, 0x1a, 0x1f

Data definitions for parsing Nortek velocity data format 3.

Field	Position Size	Description
Beams, coordinates and cells	30 2 bytes	Number of beams, coordinate system and number of cells. Object reference given in table below
STM data.Scattering	OFFSET float	Scattering measured by STM sensor
STM data.HighRange	OFFSET + 4 float	High range measured by STM sensor
STM data.reserved	OFFSET + 8 float * 4	4 reserved floats
Velocity data	VEL_START int16 * VEL_NB * VEL_NC	This field exists if the Velocity data included bit of the Config byte is set. 10^(Velocity Scaling) Unit: [m/s]
Amplitude data	AMP_POS uint8 * AMP_NB * AMP_NC	This field exists if the amplitude data included bit of the Config byte is set 0.5 dB/count Unit: [dB]
Correlation data	CORR_POS uint8 * CORR_NB * CORR_NC	This field exists if the Correlation data included bit of the Config byte is set [0 – 100 %] Unit: [%]
Altimeter data.Altimeter distance	ALTI_POS float	Distance to surface from Leading Edge algorithm Unit: [m]
Altimeter data.Altimeter quality	ALTI_POS + 4 uint16	Result of LE algorithm. When quality is deemed too low according to instrument specific limits both the distance and quality are set to 0.
Altimeter data.Altimeter status	ALTI_POS + 6 2 * 8 bits	Altimeter status bit mask
AST data.AST distance	AST_POS float	Distance to surface from Max Peak/AST algorithm Unit: [m]
AST data.AST quality	AST_POS + 4 uint16	Amplitude at which surface is detected with the Max Peak/AST algorithm.

		Raw data in steps of 0.01 dB, i.e. quality of 8000 = 80 dB Unit: [dB]
AST data.AST offset	AST_POS + 6 int16	Offset in step of measurement to velocity measurement Raw data given in 100 µs Unit: [s]
AST data.AST pressure	AST_POS + 8 float	Pressure value measured during the AST/altimeter ping Unit: [dbar]
Altimeter raw data.Num RawSamples	ALTIRAW_START + 8 uint32	Altimeter Raw Data – Number of Samples
Altimeter raw data.Samples distance	ALTIRAW_START + 12 uint16	Distance between samples Raw data given in 0.1mm Unit: [m]
Altimeter raw data.Data samples	ALTIRAW_START + 14 int16 * NRS	Altimeter Raw Data – Samples Raw data given as 16 bits Signed fract
AHRS data.Rotation matrix	AHRS_START float * 3 * 3	AHRS Rotation Matrix [3x3]
AHRS data.Quaternion W	AHRS_START + 36 float	W quaternion
AHRS data.Quaternion X	AHRS_START + 40 float	X quaternion
AHRS data.Quaternion Y	AHRS_START + 44 float	Y quaternion
AHRS data.Quaternion Z	AHRS_START + 48 float	Z quaternion
AHRS data.Gyro X	AHRS_START + 52 float	Gyro in X direction in degrees pr second Unit: [dps]
AHRS data.Gyro Y	AHRS_START + 56 float	Gyro in Y direction in degrees pr second Unit: [dps]
AHRS data.Gyro Z	AHRS_START + 60 float	Gyro in Z direction in degrees pr second Unit: [dps]
Percentage good data	PGD_START uint8 * PGD_LEN	Percent Good Estimate per cell This field exists if the Percentage Good data is included. For the Signature instruments, this will only be relevant for the _avgd.ad2cp file. Unit: [%]
Standard deviation data.Pitch	SD_START int16	Standard deviation on pitch data Raw data in 0.01 degrees

		Unit: [deg]
Standard deviation data.Roll	SD_START + 2 int16	Standard deviation on roll data Raw data in 0.01 degrees Unit: [deg]
Standard deviation data.Heading	SD_START + 4 int16	Standard deviation on heading data Raw data in 0.01 degrees Unit: [deg]
Standard deviation data.Pressure	SD_START + 6 int16	Standard deviation on pressure data Raw data in 0.001 Bar Unit: [bar]

Position and size variables:

Name	Description
STM_LEN	STM length.
OFFSET	Offset of data given at position 1 in this dataset. Number of bytes from start of record to start of data.
VEL_NB	Primary dimension of velocity data is number of beams. Length 0 if correlation data in configuration bit map is false.
VEL_NC	Second dimension of velocity data is number of cells pr beam. Length 0 if correlation data in configuration bit map is false.
VEL_START	Starting byte position of velocity data, which comes after the STM data section.
AMP_NB	Primary dimension of amplitude data is number of beams. Length 0 if correlation data in configuration bit map is false.
AMP_NC	Second dimension of amplitude data is number of cells pr beam. Length 0 if correlation data in configuration bit map is false.
AMP_POS	Correlation data starts after the amplitude data.
CORR_NB	Primary dimension of correlation data is number of beams. Length 0 if correlation data in configuration bit map is false.
CORR_NC	Second dimension of correlation data is number of cells pr beam. Length 0 if correlation data in configuration bit map is false.
CORR_POS	Correlation data starts after the amplitude data.
ALTI_POS	Altimeter data starts after the correlation data.
AST_POS	AST data starts after the altimeter data.
NRS	Number of raw samples given as first element of this object.
ALTIRAW_START	Altimeter raw data starts after the AST data.
AHRS_START	AHRS data follows the altimeter raw data.
PGD_START	Percent good data follows the AHRS data.

SD_START	The standard deviation data follows percent good data.

Object reference: Beams, coordinates and cells

Number of beams, coordinate system and number of cells.

Field	Position Size	Description
Number of beams	15-12 2 bits	Number of Beams (NB) Active beams represented as a 4 char string of 1s and 0s.
Coordinate system	11-10 2 bits	Coordinate system 00 = ENU 01 = XYZ 10 = BEAM 11 = not used
Number of cells	9-0 10 bits	Number of Cells (NC).

7.7 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.

7.8 WaveData

ID: 0x30

This chapter describes the Wave Data Format. In order to enable onboard processed wave data a wave processing license is required. Upon retrieval, the binary processed wave data can be found in the _avgd.ad2cp file. For more information about the processing routine, please refer to the Principles of Operation - Waves manual.

Field	Position Size	Description
Version	0 uint8	Version number of this Data Record Definition. Should be 1.
Offset of data	1 uint8	Number of bytes from start of record to start of non-common data fields. Unit: [bytes]
Included data	2	Bit mask to enable/disable data output.

	2 * 8 bits	Object reference given in table below
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 µs]
Wave counter	16 uint16	Counting from 1
Error	18 4 * 8 bits	Error bitmask Object reference given in table below
Status	22 4 * 8 bits	Wave status bit mask. Object reference given in table below
Spectrum type	26 uint8	Spectrum type used in processing. Values may be: 0: Pressure 1: Velocity 2: Auto depth 3: AST only
Processing method	27 uint8	Processing method for direction. 2 = SUV , 4 = MLMST
Target cell	28 uint8	
Unused	29 uint8	Unused bytes
Number of no detects	30 uint16	Number of ST Bad detects Unit: [#]
Number of bad detects	32 uint16	Number of ST Bad detects Unit: [#]
Cut off frequency	34 float	Cut off frequency, directional estimates are valid for frequencies below this limit

		Unit: [Hz]
Processing time	38 float	Processing time duration Unit: [s]
Number of zero crossings	42 uint16	Number of wave zero crossings
Version string	44 4 bytes	Version as string
Unused	48 54 bytes	56 unused spare bytes
WaveData	OFFSET WD_L bytes	Wave parameters Object reference given in table below
SwellWaves	SWELL_P SWELL_L bytes	Wave Bands Data for swells (lower frequencies). There are two bands separated with BANDSEPFREQ in SETWAVEPROC Object reference given in table below
SeaWaves	SEA_P SEA_L bytes	Wave Bands Data for sea waves (higher bands). There are two bands separated with BANDSEPFREQ Object reference given in table below
EnergySpectrum	WS_P WS_L bytes	Wave energy spectrum Object reference given in table below
FourierCoefficients	WFC_P WFC_L bytes	Wave fourier coefficients data Object reference given in table below
Direction	WDIR_P WDIR_L bytes	Wave direction data Object reference given in table below

Position and size variables:

Name	Description
WD_L	If 'Wave Pars' is true in the 'Included' bit mask, Wave data length (WD_L) is given by 80 bytes (20 floats) and 20 spare bytes.
OFFSET	Offset of data given at position 1 in this dataset. Number of bytes from start of record to start of data.
SWELL_L	If 'Wave band' is true in the 'Included' bit mask, the length of swell band is given by 52 bytes.
SWELL_P	The position of the swell band data are the sum of OFFSET and WD_L.
SEA_L	If 'Wave band' is true in the 'Included' bit mask, the length of sea band is given by 52 bytes.
SEA_P	The position of the sea band data are the sum of OFFSET, WD_L and SWELL_L.
WS_L	If 'energy spectra' is true in the 'Included' bit mask, length is given by 14 bytes (3 floats and a unit16) and 22 spare bytes and 4 times nBins in the

	dataset.
WS_P	Position of wave spectrum data are the sum of OFFSET, WD_L and WB_L.
WFC_L	If 'Fourier spectra' is true in the 'Included' bit mask, the fourier coefficients data Length (FC_L) is given by 14 bytes (3 floats and a unit16) and 22 spare bytes and nBins times 4 coefficients each a float (nBins*4*4).
WFC_P	Position of wave fourier coefficients (FC_P) are the sum of OFFSET, WD_L, WB_L and WS_L.
WDIR_L	If 'Wave direction spectra' is true in the 'Included' bit, wave direction length (WDIR_L) is given by 14 bytes (3 floats and a unit16) and 22 spare bytes.
WDIR_P	Position of wave direction spectra (WDIR_P) is the sum of OFFSET, WD_L, WB_L, WS_L and WFC_L.

Object reference: Included data

Bit mask to enable/disable data output.

Field	Position Size	Description
Wave parameters	0 bit	When 1, wave parameters is included in dataset
Energy spectra	1 bit	When 1, energy spectra is included in dataset
Wave band	2 bit	When 1, wave band is included in dataset
Fourier spectra	3 bit	When 1, fourier spectra is included in output.
Wave direction spectra	4 bit	When 1, Wave direction spectra is included in dataset
Unused	5-16 11 bits	These bits a not used.

Object reference: Error

Error bitmask

Field	Position Size	Description
No pressure	0 bit	
Low pressure	1	

	bit	
Low amp	2 bit	
White noise	3 bit	
Unreasonable estimation	4 bit	
Never processed	5 bit	
AST out of bound	6 bit	
Direction ambiguity	7 bit	
No pressure peak	8 bit	
Close to clip	9 bit	
AST height loss	10 bit	
High tilt	11 bit	
Correlation	12 bit	
Unused	13-31 19 bits	These bits a not used.

Object reference: Status

Wave status bit mask.

Field	Position Size	Description
Unused	0-15 16 bits	These bits a not used.
Active configuration	16 bit	
Unused	17-32 15 bits	These bits a not used.

Object reference: WaveData

Wave parameters

Field	Position Size	Description
Height 0	0 float	Spectral significant wave height. Unit: [m]
Height 3	4 float	AST significant (33%) wave height. Unit: [m]
Height 10	8 float	AST wave height, top 10%. Unit: [m]
Height max	12 float	AST max wave height in wave ensemble. Unit: [m]
Height mean	16 float	AST mean wave height in wave ensemble. Unit: [m]
Period mean	20 float	Mean period spectrum based. Unit: [s]
Period peak	24 float	Peak period. Unit: [s]
Period Z	28 float	AST mean zero-crossing period. Unit: [s]
Period 1/3	32 float	Mean 1/3 period. Unit: [s]
Period 1/10	36 float	Mean 1/10 period. Unit: [s]
Period max	40 float	Period for maximum wave. Unit: [s]
Period energy	44 float	Wave energy period. Unit: [s]
Direction at peak period	48 float	Direction at peak period. Unit: [deg]
Spreading at peak period	52 float	Spreading at peak period. Unit: [deg]
Wave direction mean	56 float	Mean wave direction. Unit: [deg]
Unidirectivity index	60 float	Unidirectivity index [0-1].
Pressure mean	64 float	Mean pressure during burst. Unit: [m]

Current speed mean	68 float	Mean current speed - wave cells. Unit: [m/sec]
Current direction mean	72 float	Mean current direction - wave cells. Unit: [deg]
AST distance mean	76 float	Mean AST distance during burst. Unit: [m]
Unused	80 20 bytes	20 unused spare bytes

Object reference: SwellWaves

Wave Bands Data for swells (lower frequencies). There are two bands separated with BANDSEPFREQ in SETWAVEPROC

Field	Position Size	Description
Low frequency	0 float	Unit: [Hz]
High frequency	4 float	Unit: [Hz]
Height 0	8 float	Spectral significant wave height [m] Unit: [m]
Period mean	12 float	Mean period spectrum based. Unit: [s]
periodPeak	16 float	Peak period Unit: [s]
Direction at peak period	20 float	Direction at peak period. Unit: [deg]
Wave direction mean	24 float	Mean wave direction. Unit: [deg]
Spreading at peak period	28 float	Spreading at peak period. Unit: [deg]
Unused	32 20 bytes	Unused 20 bytes

Object reference: SeaWaves

Wave Bands Data for sea waves (higher bands). There are two bands separated with BANDSEPFREQ

Field	Position	Description

	Size	
Low frequency	0 float	Unit: [Hz]
High frequency	4 float	Unit: [Hz]
Height 0	8 float	Spectral significant wave height [m] Unit: [m]
Period mean	12 float	Mean period spectrum based. Unit: [s]
periodPeak	16 float	Peak period Unit: [s]
Direction at peak period	20 float	Direction at peak period. Unit: [deg]
Wave direction mean	24 float	Mean wave direction. Unit: [deg]
Spreading at peak period	28 float	Spreading at peak period. Unit: [deg]
Unused	32 20 bytes	Unused 20 bytes

Object reference: EnergySpectrum

Wave energy spectrum

Field	Position Size	Description
Low frequency	0 float	Unit: [Hz]
High frequency	4 float	Unit: [Hz]
Step frequency	8 float	Unit: [Hz]
nBins	12 uint16	Number of spectral bins (default 98)
Unused	14 22 bytes	Unused 20 bytes
Data	36 float * 1 * nBins	Energy spectrum data stored as an array of nBins length. Unit: [cm^2/Hz]

Object reference: FourierCoefficients

Wave fourier coefficients data

Field	Position Size	Description
Low frequency	0 float	Unit: [Hz]
High frequency	4 float	Unit: [Hz]
Step frequency	8 float	Unit: [Hz]
nBins	12 uint16	Number of spectral bins (default 98)
Unused	14 22 bytes	Unused 20 spare bytes
Coefficients	36 float * 4 * nBins	Fourier coefficients stored as 4 x nBins matrix: [A1[nBins] B1[nBins] A2[nBins] B2[nBIns]]. Values range from [-1, 1].

Object reference: Direction

Wave direction data

Field	Position Size	Description
Low frequency	0 float	Unit: [Hz]
High frequency	4 float	Unit: [Hz]
Step frequency	8 float	Unit: [Hz]
nBins	12 uint16	Number of spectral bins (default 98)
Unused	14 22 bytes	Unused spare 20 bytes
Data	36 float * 2 * nBins	Wave direction data as a 2 x nBins matrix. Md[nBins] - Directional spectrum

		Ds[nBins] - Directional spread spectrum.

7.9 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.

8 Data Quality Control

When collecting average current data with a Signature instrument, two data files will be generated and stored to the recorder:

- **[filename].ad2cp:** This file contains raw, single-ping data with no quality control applied.
- **[filename]_avgd.ad2cp:** This file contains averaged current data. The data is averaged over the configured average interval, and internal quality control is applied before averaging.

The internal quality control done to the **[filename]_avgd.ad2cp** file discards all pings with a correlation below 50 % before averaging. This filtering is irreversible and ensures consistent data quality.. For more information about the correlation parameter, please refer to the Correlation - theoretical background.

Percent good

In the averaged current data (found in the **[filename]_avgd.ad2cp** file) there is also a Percentage Good value which indicates the percentage of valid pings retained in each cell during an average interval after the internal quality control.

Example
You have configured your instrument to do 10 pings within each average interval. One ping in the first cell is discarded because its correlation is below 50 %. The resulting Percent Good value will be 90 % for this cell.

If the Percent Good value is lower than 50 % (meaning more than half the pings were discarded due to low correlation), the entire cell is flagged. Flagged data will be represented by extreme placeholder values:

- Velocity: -32.767 m/s
- Speed: 46.34 m/s
- Direction: 225°

8.1 Telemetry Quality Control

When your Signature is configured to output telemetry average current data, correlation filtering is applied before the data is averaged and output. All pings below the correlation threshold are discarded. By default, the lower correlation threshold is set to 50 %, but this can be adjusted using the correlation threshold argument. The threshold can be changed by using the **CORRTH** argument with the **SETTMAVG** command.

If more than 50 % of the pings within one cell and average interval are discarded due to correlation below the threshold, the entire cell will be flagged as invalid in the telemetry output. Flagged data will be represented by extreme placeholder values in the telemetry output:

Velocity: -32.767 m/s
Speed: 46.34 m/s
Direction: 225°

When internal wave processing and telemetry of processed wave data are enabled, invalid data points are flagged using variants of -9 in the telemetry output.

9 Telemetry Data Formats

This section describes the Telemetry Data formats. For more information about the telemetry functionality in general, refer to the Telemetry chapter. Note that all averaged telemetry current data undergo an internal quality control as described in the Telemetry Quality Control chapter.

The checksum calculation is part of the NMEA standard. It is the representation of two hexadecimal characters of an XOR if all characters in the sentence between – but not including – the $ and the * character.

9.1 Averaging Mode

The telemetry of the AVG mode is controlled by the **SET/GETTMAVG** command. The DF parameter of this command sets the data format.

Data format (DF)	Description
3	Binary format as described in the binary data formats chapter.
100	Same NMEA format as previous generations of Aquadopp/AWAC. Note that Signatures produce extra columns for the fourth beam. (NMEA sentences: PNORI, PNORS and PNORC).
101	NMEA format 1 (without tags). (NMEA sentences: PNORI1, PNORS1 and PNORC1).
102	NMEA format 2 (with tags). (NMEA sentences: PNORI2, PNORS2 and PNORC2).
103	NMEA format 3 (with tags). (NMEA sentences: PNORH3, PNORS3 and PNORC3).
104	NMEA format 4 (without tags). (NMEA sentences: PNORH4, PNORS4 and PNORC4).
150	RDI Workhorse PD0 data format. Refer to RDI for documentation.

9.1.1 NMEA Format (DF=100)

Information (configuration): $PNORI

Column	Description	Data format	Example
0	Identifier	"$PNORI"	
1	Instrument type	N 0=Aquadopp 2=Aquadopp Profiler 4=Signature	4
2	Head ID	String	Signature10009 00001
3	Number of beams	N	4
4	Number of cells	N	20
5	Blanking (m)	dd.dd	0.20
6	Cell size (m)	dd.dd	1.00
7	Coordinate system	N 0=ENU 1=XYZ 2=BEAM	0
8	Checksum	*hh	2E

Example (DF=100): `$PNORI,4,Signature1000900001,4,20,0.20,1.00,0*2E`

Sensor data: $PNORS

Colum n	Description	Data format	Example
0	Identifier	"$PNORS"	
1	Date	MMDDYY	102115
2	Time	HHMMSS	090715
3	Error Code (hex)	hh	00000000
4	Status Code (hex)	hh	2A480000
5	Battery Voltage	dd.d	14.4
6	Sound Speed	dddd.d	1523.0
7	Heading	ddd.d	275.9
8	Pitch (deg)	dd.d	15.7
9	Roll (deg)	dd.d	-2.3

10	Pressure (dBar)	ddd.ddd	0.000
11	Temperature (deg C)	dd.dd	22.45
12	Analog input #1	n/a	0
13	Analog input #2	n/a	0
14	Checksum	*hh	1C

Example (DF=100):
`$PNORS,102115,090715,00000000,2A480000,14.4,1523.0,275.9,15.7,2.3,0.000,22.45,0,0*1C`

Current velocity data: $PNORC

Column	Description	Data format	Example
0	Identifier	"$PNORC"	
1	Date	MMDDYY	102115
2	Time	HHMMSS	090715
3	Cell number	N	4
4	Velocity 1 (m/s) (Beam1/X/East)	dd.dd	0.56
5	Velocity 2 (m/s) (Beam2/Y/North)	dd.dd	-0.80
6	Velocity 3 (m/s) (Beam3/Z1/Up1)	dd.dd	-1.99
7	Velocity 4 (m/s) (Beam4/Z2/Up2) - not relevant for three beam systems, will be empty	dd.dd	-1.33
8	Speed (m/s)	dd.dd	0.98
9	Direction (deg)	ddd.d	305.2
10	Amplitude unit	C=Counts	C
11	Amplitude (Beam 1)	N	80
12	Amplitude (Beam 2)	N	88
13	Amplitude (Beam 3)	N	67
14	Amplitude (Beam 4) - not relevant for three beam systems, will be empty	N	78
15	Correlation (%) (Beam1)	N	13
16	Correlation (%) (Beam2)	N	17
17	Correlation (%) (Beam3)	N	10

18	Correlation (%) (Beam4) - not relevant for three beam systems, will be empty	N	18
19	Checksum	*hh	22

Example (DF=100): $PNORC,102115,090715,4,0.56,-0.80,-1.99,-1.33,0.98,305.2,C,80,88,67,78,13,17,10,18*22

9.1.2 NMEA Format 1 and 2 (DF=101/102)

Information Data:

Identifier:
PNORI1 for DF = 101 (without tags)
PNORI2 for DF = 102 (with tags)

Column	Description	Unit	TAG	Data format	Example
1	Instrument type		IT	N 0=Aquadopp 2=Aquadopp Profiler 4=Signature	IT=4
2	Head ID		SN	N	SN=123456
3	Number of Beams		NB	N	NB=4
4	Number of Cells		NC	N	NC=30
5	Blanking Distance	[m]	BD	dd.dd	BD=1.00
6	Cell Size	[m]	CS	dd.dd	CS=5.00
7	Coordinate System (ENU,BEAM,XYZ)		CY	N	CY=BEAM

Example (DF=101):
$PNORI1,4,123456,4,30,1.00,5.00,BEAM*5B
Example (DF=102):
$PNORI2,IT=4,SN=123456,NB=4,NC=30,BD=1.00,CS=5.00,CY=BEAM*68

Sensors Data:

Identifier:
PNORS1 for DF = 101 (without tags)
PNORS2 for DF = 102 (with tags)

Column	Description	Unit	TAG	Data format	Example
1	Date		DATE	MMDDYY	DATE=083013
2	Time		TIME	HHMMSS	TIME=132455

3	Error Code		EC	N	EC=0
4	Status Code		SC	hhhhhhhh	SC=34000034
5	Battery Voltage	[V]	BV	dd.d	BV=22.9
6	Sound Speed	[m/s]	SS	dddd.d	SS=1500.0
7	Heading Std.Dev.	[deg]	HSD	dd.dd	HSD=0.02
8	Heading	[deg]	H	ddd.d	H=123.4
9	Pitch	[deg]	PI	dd.d	PI=45.6
10	Pitch Std.Dev	[deg]	PISD	dd.dd	PISD=0.02
11	Roll	[deg]	R	dd.d	R=23.4
12	Roll Std.Dev.	[deg]	RSD	dd.dd	RSD=0.02
13	Pressure	[dBar]	P	ddd.ddd	P=123.456
14	Pressure StdDev	[dBar]	PSD	dd.dd	PSD=0.02
15	Temperature	[deg C]	T	dd.dd	T=24.56

Example (DF=101):
$PNORS1,083013,132455,0,34000034,22.9,1500.0,0.02,123.4,45.6,0.02,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=22.9,SS=1500.0,HSD=0.02,H=123.4,PI=45.6,PISD=0.02,R=23.4,RSD=0.02,P=123.456,PSD=0.02,T=24.56*3F

Averaged Data:
Identifier:
PNORC1 for DF = 101 (without tags)
PNORC2 for DF = 102 (with tags)

The averaged data is repeated for each measurement cell.

Column	Description	Unit	TAG	Data format	Example
1	Date		DATE	MMDDYY	DATE=083013
2	Time		TIME	HHMMSS	TIME=132455
3	Cell Number		CN	N	CN=3
4	Cell Position	[m]	CP	dd.d	CP=11.0
5	Velocity East - only included if CY=ENU	[m/s]	VE	dd.ddd	VE=0.332
6	Velocity North - only included if CY=ENU	[m/s]	VN	dd.ddd	VN=0.332

Column	Description	Unit	TAG	Data format	Example
7	Velocity Up - only included if CY=ENU	[m/s]	VU	dd.ddd	VU=0.332
8	Velocity Up 2 - only included if CY=ENU	[m/s]	VU2	dd.ddd0	VU2=0.332
9	Velocity X - only included if CY = XYZ	[m/s]	VX	dd.ddd	VX=0.332
10	Velocity Y - only included if CY = XYZ	[m/s]	VY	dd.ddd	VY=0.332
11	Velocity Z - only included if CY = XYZ	[m/s]	VZ	dd.ddd	VZ=0.332
12	Velocity Z 2 - only included if CY=XYZ	[m/s]	VZ2	dd.ddd	VZ2=0.332
13	Velocity Beam 1 - only included if CY=BEAM	[m/s]	V1	dd.ddd	V1=0.332
14	Velocity Beam 2 - only included if CY=BEAM	[m/s]	V2	dd.ddd	V2=0.332
15	Velocity Beam 3 - only included if CY=BEAM	[m/s]	V3	dd.ddd	V3=0.332
16	Velocity Beam 4 - only included if CY=BEAM	[m/s]	V4	dd.ddd	V4=0.332
17	Amplitude Beam 1	[dB]	A1	ddd.d	A1=78.9
18	Amplitude Beam 2	[dB]	A2	ddd.d	A2=78.9
19	Amplitude Beam 3	[dB]	A3	ddd.d	A3=78.9
20	Amplitude Beam 4	[dB]	A4	ddd.d	A4=78.9
21	Correlation Beam 1	[%]	C1	N	C1=78
22	Correlation Beam 2	[%]	C2	N	C2=78
23	Correlation Beam 3	[%]	C3	N	C3=78
24	Correlation Beam 4	[%]	C4	N	C4=78

Example (DF=101, CY=ENU):
```
$PNORC1,083013,132455,3,11.0,0.332,0.332,0.332,0.332,78.9,78.9,78.9,78.9,78
,78,78,78*46
```
Example (DF=102, CY=ENU):
```
$PNORC2,DATE=083013,TIME=132455,CN=3,CP=11.0,V1=0.332,V2=0.332,V3=-
0.332,V4=-0.332,A1=78.9,A2=78.9,A3=78.9,A4=78.9,C1=78,C2=78,C3=78,C4=78*49
```

9.1.3 NMEA Format 3 and 4 (DF=103/104)

Header Data:

Identifier:

PNORH3 for DF = 103 (with tags)

PNORH4 for DF = 104 (without tags)

Column	Description	Unit	TAG	Data format	Example
1	Date		DATE	YYMMDD	DATE=141112
2	Time		TIME	HHMMSS	TIME=081946
3	Error Code		EC	N	EC=0
4	Status Code		SC	hhhhhhhh	SC=2A4C0000

Example (DF=103):
$PNORH3,DATE=141112,TIME=081946,EC=0,SC=2A4C0000*5F
Example (DF=104):
$PNORH4,141112,083149,0,2A4C0000*4A68

Sensors Data:
Identifier:
PNORS3 for DF = 103 (with tags)
PNORS4 for DF = 104 (without tags)

Column	Description	Unit	TAG	Data format	Example
1	Battery	[V]	BV	dd.d	BV=22.9
2	Sound Speed	[m/s]	SS	dddd.d	SS=1500.0
3	Heading	[deg]	H	ddd.d	H=123.4
4	Pitch	[deg]	PI	dd.d	PI=45.6
5	Roll	[deg]	R	dd.d	R=23.4
6	Pressure	[dBar]	P	ddd.ddd	P=123.456
7	Temperature	[deg C]	T	dd.dd	T=24.56

Example (DF=103):
$PNORS3,BV=22.9,SS=1546.1,H=151.1,PI=-12.0,R=-5.2,P=705.669,T=24.96*7A
Example (DF=104):
$PNORS4,22.9,1546.1,151.2,-11.9,-5.3,705.658,24.95*5A

Averaged Data:
Identifier:
PNORC3 for DF = 103 (with tags)
PNORC4 for DF = 104 (without tags)

The averaged data is repeated for each measurement cell.

Column	Description	Unit	TAG	Data format	Example
1	Cell position	[m]	CP	dd.d	CP=2.5
2	Speed	[m/s]	SP	dd.ddd	SP=0.751
3	Direction	[deg]	DIR	ddd.d	DIR=170.1
4	Averaged Correlation		AC	N	AC=5
5	Averaged Amplitude		AA	N	AA=28

Example (DF=103):
`$PNORC3,CP=4.5,SP=3.519,DIR=110.9,AC=6,AA=28*3B`
Example (DF=104):
`$PNORC4,27.5,1.815,322.6,4,28*70`

9.2 Burst

The telemetry of the BURST mode is controlled by the **SET/GETTMBURST** command. The DF parameter of this command sets the data format.

Data format (DF)	Description
3	Binary format as described in 'Data Record Definition (version 3)'.
101	NMEA format 1 (without tags). (NMEA sentences: PNORI1, PNORS1 and PNORC1).
102	NMEA format 2 (with tags). (NMEA sentences: PNORI2, PNORS2 and PNORC2).
103	NMEA format 3 (with tags). (NMEA sentences: PNORH3, PNORS3 and PNORC3).
104	NMEA format 4 (without tags). (NMEA sentences: PNORH4, PNORS4 and PNORC4).

9.2.1 NMEA Format 1 and 2 (DF=101/102)

Information Data:

Identifier:

PNORI1 for DF = 101 (without tags)

PNORI2 for DF = 102 (with tags)

Column	Description	Unit	TAG	Data format	Example
1	Instrument type		IT	N 0=Aquadopp 2=Aquadopp Profiler 4=Signature	IT=4
2	Head ID		SN	N	SN=123456
3	Number of Beams		NB	N	NB=4
4	Number of Cells		NC	N	NC=30
5	Blanking Distance	[m]	BD	dd.dd	BD=1.00
6	Cell Size	[m]	CS	dd.dd	CS=5.00
7	Coordinate System (ENU,BEAM,XYZ)		CY	N	CY=BEAM

Example (DF=101):

$PNORI1,4,123456,4,30,1.00,5.00,BEAM*5B

Example (DF=102):

$PNORI2,IT=4,SN=123456,NB=4,NC=30,BD=1.00,CS=5.00,CY=BEAM*68

Sensors Data:

Identifier:

PNORS1 for DF = 101 (without tags)
PNORS2 for DF = 102 (with tags)

Column	Description	Unit	TAG	Data format	Example
1	Date		DATE	MMDDYY	DATE=083013
2	Time		TIME	HHMMSS	TIME=132455
3	Error Code		EC	N	EC=0
4	Status Code		SC	hhhhhhhh	SC=34000034
5	Battery Voltage	[V]	BV	dd.d	BV=22.9
6	Sound Speed	[m/s]	SS	dddd.d	SS=1500.0
7	Heading Std.Dev.	[deg]	HSD	dd.dd	HSD=0.02
8	Heading	[deg]	H	ddd.d	H=123.4
9	Pitch	[deg]	PI	dd.d	PI=45.6
10	Pitch Std.Dev	[deg]	PISD	dd.dd	PISD=0.02
11	Roll	[deg]	R	dd.d	R=23.4
12	Roll Std.Dev.	[deg]	RSD	dd.dd	RSD=0.02
13	Pressure	[dBar]	P	ddd.ddd	P=123.456
14	Pressure StdDev	[dBar]	PSD	dd.dd	PSD=0.02
15	Temperature	[deg C]	T	dd.dd	T=24.56

Example (DF=101):
$PNORS1,083013,132455,0,34000034,22.9,1500.0,0.02,123.4,45.6,0.02,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=22.9,SS=1500.0,HSD=0.02,H=123.4,PI=45.6,PISD=0.02,R=23.4,RSD=0.02,P=123.456,PSD=0.02,T=24.56*3F

Averaged Data:
Identifier:
PNORC1 for DF = 101 (without tags)
PNORC2 for DF = 102 (with tags)

The averaged data is repeated for each measurement cell.

Column	Description	Unit	TAG	Data format	Example
1	Date		DATE	MMDDYY	DATE=083013
2	Time		TIME	HHMMSS	TIME=132455
3	Cell Number		CN	N	CN=3
4	Cell Position	[m]	CP	dd.d	CP=11.0
5	Velocity East - only included if CY=ENU	[m/s]	VE	dd.ddd	VE=0.332
6	Velocity North - only included if CY=ENU	[m/s]	VN	dd.ddd	VN=0.332
7	Velocity Up - only included if CY=ENU	[m/s]	VU	dd.ddd	VU=0.332
8	Velocity Up 2 - only included if CY=ENU	[m/s]	VU2	dd.ddd0	VU2=0.332
9	Velocity X - only included if CY = XYZ	[m/s]	VX	dd.ddd	VX=0.332
10	Velocity Y - only included if CY = XYZ	[m/s]	VY	dd.ddd	VY=0.332
11	Velocity Z - only included if CY = XYZ	[m/s]	VZ	dd.ddd	VZ=0.332
12	Velocity Z 2 - only included if CY=XYZ	[m/s]	VZ2	dd.ddd	VZ2=0.332
13	Velocity Beam 1 - only included if CY=BEAM	[m/s]	V1	dd.ddd	V1=0.332
14	Velocity Beam 2 - only included if CY=BEAM	[m/s]	V2	dd.ddd	V2=0.332
15	Velocity Beam 3 - only included if CY=BEAM	[m/s]	V3	dd.ddd	V3=0.332
16	Velocity Beam 4 - only included if CY=BEAM	[m/s]	V4	dd.ddd	V4=0.332
17	Amplitude Beam 1	[dB]	A1	ddd.d	A1=78.9
18	Amplitude Beam 2	[dB]	A2	ddd.d	A2=78.9
19	Amplitude Beam 3	[dB]	A3	ddd.d	A3=78.9
20	Amplitude Beam 4	[dB]	A4	ddd.d	A4=78.9
21	Correlation Beam 1	[%]	C1	N	C1=78
22	Correlation Beam 2	[%]	C2	N	C2=78
23	Correlation Beam 3	[%]	C3	N	C3=78
24	Correlation Beam 4	[%]	C4	N	C4=78

Example (DF=101, CY=ENU):

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

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

9.2.2 NMEA Format 3 and 4 (DF=103/104)

Header Data:

Identifier:
PNORH3 for DF = 103 (with tags)
PNORH4 for DF = 104 (without tags)

Column	Description	Unit	TAG	Data format	Example
1	Date		DATE	YYMMDD	DATE=141112
2	Time		TIME	HHMMSS	TIME=081946
3	Error Code		EC	N	EC=0
4	Status Code		SC	hhhhhhhh	SC=2A4C0000

Example (DF=103):
$PNORH3,DATE=141112,TIME=081946,EC=0,SC=2A4C0000*5F
Example (DF=104):
$PNORH4,141112,083149,0,2A4C0000*4A68

Sensors Data:

Identifier:
PNORS3 for DF = 103 (with tags)
PNORS4 for DF = 104 (without tags)

Column	Description	Unit	TAG	Data format	Example
1	Battery	[V]	BV	dd.d	BV=22.9
2	Sound Speed	[m/s]	SS	dddd.d	SS=1500.0
3	Heading	[deg]	H	ddd.d	H=123.4
4	Pitch	[deg]	PI	dd.d	PI=45.6
5	Roll	[deg]	R	dd.d	R=23.4
6	Pressure	[dBar]	P	ddd.ddd	P=123.456
7	Temperature	[deg C]	T	dd.dd	T=24.56

Example (DF=103):
$PNORS3,BV=22.9,SS=1546.1,H=151.1,PI=-12.0,R=-5.2,P=705.669,T=24.96*7A

Example (DF=104):
$PNORS4,22.9,1546.1,151.2,-11.9,-5.3,705.658,24.95*5A

Averaged Data:
Identifier:
PNORC3 for DF = 103 (with tags)
PNORC4 for DF = 104 (without tags)

The averaged data is repeated for each measurement cell.

Column	Description	Unit	TAG	Data format	Example
1	Cell position	[m]	CP	dd.d	CP=2.5
2	Speed	[m/s]	SP	dd.ddd	SP=0.751
3	Direction	[deg]	DIR	ddd.d	DIR=170.1
4	Averaged Correlation		AC	N	AC=5
5	Averaged Amplitude		AA	N	AA=28

Example (DF=103):
`$PNORC3,CP=4.5,SP=3.519,DIR=110.9,AC=6,AA=28*3B`
Example (DF=104):
`$PNORC4,27.5,1.815,322.6,4,28*70`

9.3 Altimeter

The telemetry for the Altimeter is controlled by the **SET/GETTMALTI** command. The DF parameter of this command sets the data format. Please note that telemetered altimeter data is always derived from the Leading Edge algorithm.

Data format (DF)	Description
200	NMEA (PNORA) format without Tags.
201	NMEA (PNORA) format with Tags.

Column	Description	Unit	TAG	Data format	Example
1	Date		DATE	YYMMDD	DATE=130830
2	Time		TIME	hhmmss	TIME=132455
3	Pressure	[dBar]	P	ddd.ddd	P=123.456
4	Altimeter Distance (Leading Edge)	[m]	A	ddd.ddd	A=112.233
5	Quality Parameter		Q	N	Q=13068
6	Status		ST	XX	ST=00
7	Pitch	[deg]	PI	d.d	PI=2.3
8	Roll	[deg]	R	d.d	R=1.3

Example (DF=200):
$PNORA,190902,122341,0.000,24.274,13068,08,-2.6,-0.8*7E
Example (DF=201):
$PNORA,DATE=190902,TIME=122341,P=0.000,A=24.274,Q=13068,ST=08,PI=-2.6,R=-0.8*72

9.4 Waves

The telemetry of internally processed wave data is controlled by the **SET/GETTMWAVE** command. The DF parameter of this command sets the data format.

Data format (DF)	Description
500	Binary format as described in the data format chapter.
501	NMEA format (NMEA sentences: PNORW, PNORB, PNORE, PNORF, PNORWD)
502	Legacy binary format, same as for the previous generation of AWACs. Described in Integrators Guide - Classic

9.4.1 NMEA Waves DF501

• Data with variants of -9 (-9.00, -999…) are invalid data.
• Empty fields are unused.
• The checksum calculation is part of the NMEA standard. It is the representation of two hexadecimal characters of an XOR if all characters in the sentence between – but not including – the $ and the * characters.

Wave parameters

Column	Description	Unit	Data format
0	Identifier		“$PNORW”
1	Date		MMDDYY
2	Time		hhmmss
3	Spectrum basis type		N 0 = Pressure 1 = Velocity 3 = AST
4	Processing method		N 1 = PUV 2 = SUV 3= MLM 4 = MLMST
5	Hm0	[m]	dd.dd
6	H3	[m]	dd.dd
7	H10	[m]	dd.dd
8	Hmax	[m]	dd.dd
9	Tm02	[s]	dd.dd

10	Tp	[s]	dd.dd
11	Tz	[s]	dd.dd
12	DirTp	[deg]	ddd.dd
13	SprTp	[deg]	ddd.dd
14	Main direction	[deg]	ddd.dd
15	Unidirectivity index		dd.dd
16	Mean pressure	[dbar]	dd.dd
17	Number of no detects		N
18	Number of bad detects		N
19	Near surface current speed	[m/s]	dd.dd
20	Near surface current direction	[deg]	ddd.dd
21	Wave error code		hhhh
22	Checksum		*hh

Example (DF=501):
$PNORW,120720,093150,0,1,0.89,-9.00,1.13,1.49,1.41,1.03,-9.00,190.03,80.67,113.52,0.54,0.00,1024,0,1.19,144.11,0D8B*7B

Wave band parameters

Column	Description	Unit	Data format
0	Identifier		“$PNORB”
1	Date		MMDDYY
2	Time		hhmmss
3	Spectrum basis type		N 0 = Pressure 1 = Velocity 3 = AST
4	Processing method		N 1 = PUV 2 = SUV 3= MLM 4 = MLMST
5	Frequency Low		d.dd
6	Frequency High		d.dd
7	Hmo	[m]	dd.dd

8	Tm02	[s]	dd.dd
9	Tp	[s]	dd.dd
10	DirTp	[deg]	ddd.dd
11	SprTp	[deg]	ddd.dd
12	Main direction	[deg]	ddd.dd
13	Wave error code		hhhh
14	Checksum		*hh

Example (DF=501):
$PNORB,120720,093150,1,4,0.02,0.20,0.27,7.54,12.00,82.42,75.46,82.10,0000*67
$PNORB,120720,093150,1,4,0.21,0.99,0.83,1.36,1.03,45.00,0.00,172.16,0000*5C

Wave energy density spectrum

Column	Description	Unit	Data format
0	Identifier		"$PNORE"
1	Date		MMDDYY
2	Time		hhmmss
3	Spectrum basis type		N 0 = Pressure 1 = Velocity 3 = AST
4	Start Frequency	[Hz]	d.dd
5	Step Frequency	[Hz]	d.dd
6	Number of Frequencies - N		nnn
7	Energy density (frequency 1)	[cm^2/Hz]	dddd.ddd
8	Energy density (frequency 2)	[cm^2/Hz]	dddd.ddd
N+6	Energy density (frequency N)	[cm^2/Hz]	dddd.ddd
N+7	Checksum		*hh

Example (DF=501):
$PNORE,120720,093150,1,0.02,0.01,98,0.000,0.000,0.000,0.000,0.003,0.012,0.046,0.039,0.041,0.039,0.036,0.039,0.041,0.034,0.034,0.031,0.026,0.027,0.025,0.024,0.023,0.025,0.023,0.020,0.020,0.025,0.023,0.027,0.029,0.033,0.029,0.033,0.028,0.032,0.031,0.033,0.029,0.032,0.032,0.031,0.041,0.038,0.043,0.050,0.048,0.042,0.034,0.030,0.033,0.039,0.036,0.035,0.042,0.039,0.038,0.044,0.042,0.054,0.065,0.064,0.054,0.051,0.064,0.062,0.051,0.049,0.066,0.068,0.073,0.062,0.064,0.062,0.063,0.061,0.062,0.059,0.060,0.051,0.049,0.059,0.075,0.096,0.093,0.084,0.084,0.074,0.081,0.076,0.103,0.098,0.114,0.103,0.117,0.125,0.131,0.144,0.143,0.129*71

Fourier coefficient spectra

Column	Description	Unit	Data format
0	Identifier		“$PNORF”
1	Fourier coefficient flag (A1/B1/A2/B2)		"CC"
2	Date		MMDDYY
3	Time		hhmmss
4	Spectrum basis type		N 0 = Pressure 1 = Velocity 3 = AST
5	Start Frequency	[Hz]	d.dd
6	Step Frequency	[Hz]	d.dd
7	Number of Frequencies - N		nnn
8	Fourier coefficient CC (frequency 1)		dddd.ddd
9	Fourier coefficient CC (frequency 2)		dddd.ddd
N+7	Fourier coefficient CC (frequency N)		dddd.ddd
N+8	Checksum		*hh

Example (DF=501):

$PNORF,A1,120720,093150,1,0.02,0.01,98,0.0348,0.0958,0.1372,0.1049,-0.0215,-
0.0143,0.0358,0.0903,0.0350,0.0465,-0.0097,0.0549,-0.0507,-0.0071,-0.0737,0.0459,-0.0164,0.0275,-
0.0190,-0.0327,-0.0324,-0.0364,-0.0255,-0.0140,-9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-
9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-
9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-
9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-
9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-
9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-
9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-9.0000*0D
$PNORF,B1,120720,093150,1,0.02,0.01,98,-0.0230,-
0.0431,0.0282,0.0151,0.0136,0.0465,0.1317,0.1310,0.0500,0.0571,0.0168,0.0713,-0.0002,0.0164,-
0.0315,0.0656,-0.0046,0.0364,-0.0058,0.0227,0.0014,0.0077,0.0017,0.0041,-9.0000,-9.0000,-9.0000,-
9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-
9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-
9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-
9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-
9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-
9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-9.0000*2F
$PNORF,A2,120720,093150,1,0.02,0.01,98,-0.3609,-0.0617,0.0441,0.0812,-0.0956,-0.1695,-0.3085,-
0.2760,-0.2235,-0.1159,-0.0956,-0.0421,-0.0474,0.0119,0.0079,-0.0578,-0.1210,-0.1411,-0.0939,-0.1063,-
0.1158,-0.1201,-0.1393,-0.1556,-9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-
9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-

9.0000,-9.0000*26
$PNORF,B2,120720,093150,1,0.02,0.01,98,0.6465,0.3908,0.3669,0.3364,0.6169,0.6358,0.6473,0.6038,0.5338,0.4258,0.3862,0.3817,0.3692,0.2823,0.1669,0.1052,0.0019,-0.1209,-0.2095,-0.2144,-0.2109,-0.2509,-0.2809,-0.3491,-9.0000*07

Wave directional spectra

Column	Description	Unit	Data format
0	Identifier		“$PNORWD”
1	Main direction/directional spread (MD/DS)		"CC"
2	Date		MMDDYY
3	Time		hhmmss
4	Spectrum basis type		N 0 = Pressure 1 = Velocity 3 = AST
5	Start Frequency	[Hz]	d.dd
6	Step Frequency	[Hz]	d.dd
7	Number of Frequencies - N		nnn
8	Direction/spread (frequency 1)	[deg]	dddd.ddd
9	Direction/spread (frequency 2)	[deg]	dddd.ddd
N+7	Direction/spread (frequency N)	[deg]	dddd.ddd
N+8	Checksum		*hh

Example (DF=501):
$PNORWD,MD,120720,093150,1,0.02,0.01,98,326.5016,335.7948,11.6072,8.1730,147.6098,107.1336,74.8001,55.4424,55.0203,50.8304,120.0490,52.4414,180.2204,113.3304,203.1034,55.0302,195.6657,52.9780,196.9988,145.2517,177.5576,168.0439,176.1304,163.7607,-9.0000,-

```
9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-
9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-9.0000*05
$PNORWD,DS,120720,093150,1,0.02,0.01,98,79.3190,76.6542,75.1406,76.6127,79.9920,79.0342,75.29
61,74.3028,78.5193,77.9860,80.2380,77.2964,78.9473,80.3010,77.7126,77.7154,80.3341,79.1574,80.22
08,79.4005,79.7031,79.5054,79.9868,80.4341,-9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-
9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-
9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-
9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-
9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-
9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-
9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-9.0000,-9.0000*16
```

10 Appendices

10.1 Echosounder equations

F = frequency in kHz

F_c = center frequency in kHz, given by the FREQ parameter for the last beam, typically BEAMCFGLIST,BEAM=5,FREQ

MajorAxis3dBBeamAngle:

GETUSERECHOMAJORANGLE values when EN=1:

$$\text{Major angle}_F = P0 + P1(F - F_c) + P2(F - F_c)^2 + P3(F - F_c)^3 + P4(F - F_c)^4 + P5(F - F_c)^5 \text{ [degrees]}$$

MajorAxis3dBBeamAngleOffset:

GETUSERECHOMAJOROFFSET values when EN=1:

$$\text{Major offset}_F = P0 + P1(F - F_c) + P2(F - F_c)^2 + \cdots + P5(F - F_c)^5 \text{ [degrees]}$$

MinorAxis3dBBeamAngle:

GETUSERECHOMINORANGLE values when EN=1:

$$\text{Minor angle}_F = P0 + P1(F - F_c) + P2(F - F_c)^2 + \cdots + P5(F - F_c)^5 \text{ [degrees]}$$

MinorAxis3dBBeamAngleOffset:

GETUSERECHOMINOROFFSET values when EN=1:

$$\text{Minor offset}_F = P0 + P1(F - F_c) + P2(F - F_c)^2 + \cdots + P5(F - F_c)^5 \text{ [degrees]}$$

TwoWayBeamAngle:

GETUSERECHOTWOWAYANGLE values when EN=1:

$$\text{Two way angle}_F = P0 + P1(F - F_c) + P2(F - F_c)^2 + \cdots + P5(F - F_c)^5 \text{ [dB steradians]}$$

Gain:

GETUSERECHOGAIN polynomial values when EN=1. When pulse compression is enabled the polynomial expression shall define the gain over the bandwidth given by:

$$\text{Gain}_F = P0 + P1(F - F_c) + P2(F - F_c)^2 + \cdots + P5(F - F_c)^5 \text{ [dB]}$$

The bandwidth is given in % by the BW parameter for the last beam, typically BEAMCFGLIST,BEAM=5,BW

For monochromatic echograms the gains at the monochromatic frequency specified in GETECHO,FREQ is given directly for up to three frequencies:

$$\text{Gain}_{FreqA} = \text{GAINFA [dB]}$$
$$\text{Gain}_{FreqB} = \text{GAINFB [dB]}$$
$$\text{Gain}_{FreqC} = \text{GAINFC [dB]}$$

10.2 Raw echosounder data parsing

The raw echosounder data includes samples received during the transmit of the pulse and the blanking distance to facilitate more flexible user processing of the raw data, i.e. the receiver starts at the same time as the transmitter. Part of the raw echosounder data structure is the field "startSampleIndex". This is the number of samples passed from when the transmission starts to the time when the firmware starts processing samples to create the echogram. Data received during the blanking duration is ignored in the firmware echogram processing. They are discarded since they either are part of a reflection of a partial pulse or from reflections within the blanking distance. A formula may be derived:

$$i_{startSample} = \frac{t_{XMIT1} + t_{2BD}}{t_{sample}} = \frac{t_{XMIT1} + 2BD}{v_{sound}} f_{rawSamples}$$

Note: i is the notation for index, t for time, v for velocity and f for frequency.

- XMIT1 = duration of the transmit pulse (ms)
- BD = blanking distance (m)

Example:

GETECHO,XMIT1 = 1 m/s

GETECHO,BD = 2 m

v_{sound} = 1500 m/s

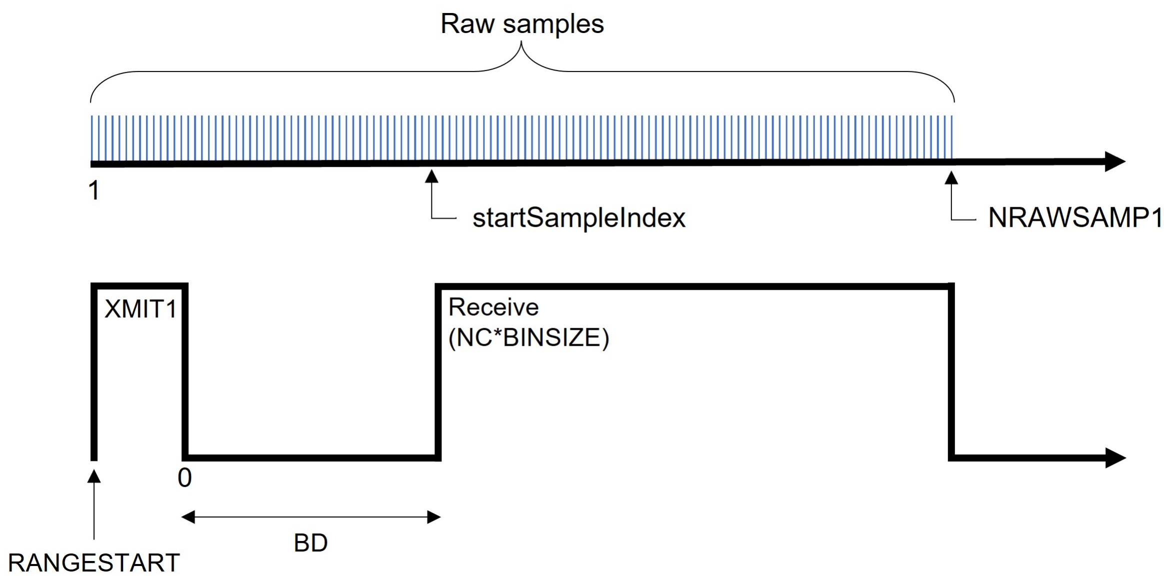
$f_{rawSamples}$ ≈ 4464 Hz

Thus:

$i_{startSample}$ = 16,368 ~ 16

The diagram below illustrates how the raw echosounder sampling works. The raw samples are transmitted and received from the start of the transmit pulse (XMIT). After the blanking distance and at startSampleIndex, the firmware will start to process the received samples (Receive) until the full range has been achieved (NC*BINSIZE). This endpoint defines the full number of samples in the raw echosounder profiler (NRAWSAMP1).

XMIT1, NC (number of cells), BINSIZE and BD can be found using the GETECHO command. RANGESTART and NRAWSAMP1 can be found using the READECHO command. As discussed, startSampleIndex is found post-measurement in the RawEchosounderData structure.



10.3 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;
}
```