RESON SeaBat™ 7k DATA FORMAT

INTERFACE CONTROL DOCUMENT

Draft Version 0.42

Approval

Notice

The information contained in this document is subject to change without notice. RESON makes no warranty of any kind with regard to this material, including (but not limited to) the implied warranties of merchant liability and fitness for a particular purpose. RESON shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance, or use of this material

Function	Name	Version	Date	Signature
Program Manager	Dan Suchman	1.00		

Protocol Version History

Protocol Version (DRF and NF)	ICD Version
1	0.01 to 0.31
2	0.32 +

Table of Contents

1. INTRODUCTION

This document describes the data format used to log and network-transmit data with the 7k series system. It is a record based protocol that encapsulates data using frames and headers.

A record can hold any type of data. All records have a unique identifier identifying the data type. Each record is built up around a frame wrapping the record. This frame identifies and describes the content of the record. TCP/UDP transmission uses an additional preceding header for better packet handling.

This document defines a set of records for 7k specific data as well as generic sensor data, such as bathymetric data and backscatter imagery data. An optional field is available for non-generic sensor data for each record.

Files has a set of basic records for better parsing performance. The generic format described herein also supports an enhanced data lookup mechanism to improve performance during file random access by utilizing an optional built-in record-indexing scheme. A built in sync pattern combined with the checksum makes it possible to recover records in corrupted files.

The data format defines a set of position, rotation, data types and time conventions for easy handling of the data.

2. CONVENTIONS

This section describes sign conventions, data types and time definition

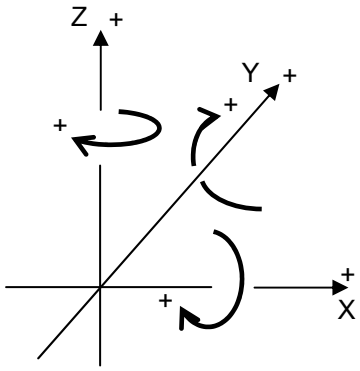
SIGN CONVENTIONS:
Unless otherwise stated, all offset measurements shall be relative to the Vessel Reference Point (VRP). Distances shall be in meters and angles in radians.

The convention used for 3D coordinate rotation is roll, pitch then yaw.

The following sign convention shall be used:

OFFSET	SIGN	DESCRIPTION
X	+ -	Starboard of the VRP Port of the VRP
Y	+ -	Forward of the VRP Astern of the VRP
Z	+ -	Height above the VRP Depth below the VRP
Roll	Port up is positive	Rotation about the along-ship (Y) axis
Pitch	Bow up is positive	Rotation about the across-ship (X) axis
Yaw	Bow right is positive	Rotation about the vertical (Z) axis
Heave	Positive up	
Heading	Positive clockwise	True heading
Altitude	Positive Up	Distance from sea-bottom to the device
Depth	Positive Up	Distance from sea surface to the device
Tide	Positive Up Negative Down	High tide: positive height. Low tide: negative height.

Vessel Axes:



BEAM POSITIONS:
Beam zero / first beam is on the port / left side of the vehicle when the array is installed with the projector facing down and pointing aft.

DATA TYPES DEFINITIONS:
The following data type formats are defined by this document.

- Unsigned values – 'uX' is always an unsigned integer, X bits wide. E.g. u32 = unsigned 32 bits.
- Signed values – 'iX' is always a signed integer, X bits wide. E.g. i16 = signed 16 bits.
- Floating points – They are either f32 or f64 (IEEE 754-1985).

All headers are of static size unless stated otherwise and shall use "struct member alignment" of 1 (one) Bytes (8 bits) in memory. Data shall be represented in little Endian (Intel) byte order unless stated otherwise.

TIME DEFINITION:
Time tags shall be in UTC unless stated otherwise and use the following structure (**7KTIME**):

NAME	SIZE	DESCRIPTION
Year	u16	0 – 65535
Day	u16	1 – 366
Seconds	f32	0.000000 – 59.999999
Hours	u8	0 – 23
Minutes	u8	0 – 59

3. TCP AND UDP

TCP sessions should conform to RFC 793 and UDP to RFC 768.

Unless otherwise stated, TCP connections should not use the Nagle algorithm in order to minimize network latency.

Both source and destination port must be populated with a unique port number for TCP and UDP transmissions.

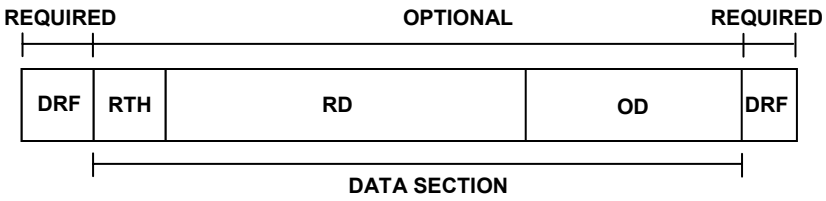
4. RECORD DEFINITION

The 7k record consists of a data record frame (header and checksum), a record type header, an optional record data field and an optional data field for extra information. The optional data field typically holds non-generic sensor specific data.

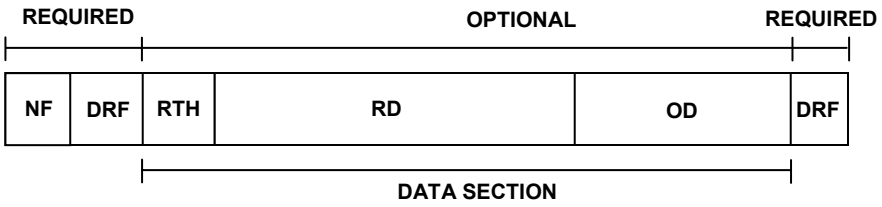
When 7k records are transmitted over a network, a network frame shall precede each record.

7k RECORD

- DRF – Data Record Frame.
- RTH – Record Type Header.
- RD – Record Data.
- OD – Optional Data.



Network prepared with the Network Frame (NF).



5. DATA RECORD FRAME (DRF)

The following data record frame is the generic wrapper in which all records (sensor data or otherwise) shall be embedded. The sync pattern combined with checksum makes it possible to recover records in corrupted files. A record frame always starts with version and offset fields. The frame (version 1) is defined thus:

NAME	SIZE	DESCRIPTION
Version	u16	Version of this frame (e.g.: 1, 2, etc.)
Offset	u16	Offset in bytes from the start of the sync pattern to the start of the DATA SECTION. This allows for expansion of the header whilst maintaining backward compatibility.
Sync pattern	u32	0x0000FFFF
Size	u32	Size in bytes of this record from the start of the version field to the end of the checksum field — that is, it includes the embedded data size.
Optional data offset	u32	Offset in bytes to optional data field from start of record. Zero (0) bytes implies no optional data.
Optional data identifier	u32	Identifier for optional field. Zero (0) if there is no optional field. This identifier is described with each record type.
7KTIME	u8*10	Time tag.
Reserved	u16	Reserved.
Record type identifier	u32	Identifier for record type of embedded data.
Device identifier	u32	Identifier of the device that this data pertains.
Subsystem identifier	u16	Identifier for the device subsystem.
System enumerator	u16	System enumerator for identical systems in one installation. 0 – N.
Data set	u32	Data set number.
Record count	u32	Sequential record counter.
Previous record	i64	Pointer to the previous record of the same type (in bytes from start of file). This is an optional field for files and shall be –1 if not used.
Next record	i64	Pointer to the next record of the same type in bytes from start of file. This is an optional field for files and shall be -1 if not used.
Flags	u16	BIT FIELD Bit 1 – Valid checksum.
Reserved	u16	Reserved.
DATA SECTION	Dynamic	Data Section.
Checksum	u32	Sum of bytes in data section (optional, depends on bit 1 of Flags field). Note: the checksum field should be computed as a 64 bit unsigned integer with the least significant 32 bits used to populate this field — thus ensuring a valid checksum and avoiding an explicit overflow.

6. TCP AND UDP NETWORK FRAME (NF)

In order to facilitate network transport via both the TCP and UDP/IP protocols, records will be packetized using the following header. In this scheme, a series of network packets may contain either a partial record or one or more data records depending upon the boundary size criterion. A series of packets are allowed to of up to a maximum of 128 records.

Each packet shall be less than or equal to 64k bytes including the network header. Packet sizes may not vary in a sequence except for the last packet.

The following header shall prefix the network packet:

NAME	SIZE	DESCRIPTION
Version	u16	Version of this frame (e.g.: 1, 2, …)
Offset	u16	Offset, in bytes, to the start of data from the start of this packet.
Total packets	u32	Number of network packets for set of records transmitted.
Total records	u16	Total number of records in network packets transmitted (helper field for parsing data). Max 128 records per transmission.
Transmission identifier	u16	Transmission identifier (helper field for packet assembly). Must be the same number for each network packet in transmission. Adjacent transmissions in time from one source may not use the same identifier.
Packet size	u32	Size in bytes of this packet including the header and appended data.
Total size	u32	Total size in Bytes of all packets in transmission excluding network frame(s).
Sequence number	u32	Sequential packet number; allows correct ordering during reconstruction. Range: 0 (zero) to N – 1 packets.
Data	Dynamic	Start of data with either a partial or one or more records.

7. LOGGING FILE FORMAT

A valid 7k data file shall be a binary file consisting of a series of data records. Records must be complete and without the network frame. A file header record is recommended as the first record in each file describing the file contents.

8. FILE NOMENCLATURE

It is recommended that file names be based on the UTC date and time that they are created and utilize a ".s7k' extension as follows:

"YYYYMMDD_HHMMSS.s7k"

Where YYYYMMDD represents the date and HHMMSS the time.

For example, 20010516_102852.s7k

Multiple files created at the same time may be differentiated by appending _X to the filename. Here, X is an integer starting at zero and successively incremented for each file.

For example, 20010516_102852_0.s7k and 20010516_102852_1.s7k

9. RECORD TYPES DEFINITIONS

The following table summarizes the allocated record type identifiers.

RECORD TYPE	DESCRIPTION
1000-1999	**Reserved for generic sensor records**
1000	Reference point information
1001	Sensor offset position information
1002	Calibrated sensor offset position information
1003	Position
1004	Attitude
1005	Tide
1006	Altitude
1007	Motion over ground
1008	Depth
1009	Sound Velocity Profile
1010	CTD
1011	Geodesy
2000-2999	**Free range for user-defined records**
3000	EdgeTech FS-DW side scan sonar data record
3001	EdgeTech FS-DW sub-bottom profile data record
3100	Framed BlueFin data frame record
7000-7999	**Reserved for SeaBat™ 7k records**
7000	7k Volatile sonar settings
7001	7k Configuration settings.
7002	7k Match Filter
7004	7k Beam geometry
7005	7k Calibration data
7006	7k Bathymetric data
7007	7k Backscatter image data
7008	7k Beam data
7051	7k System event
7052	7k Data storage status information
7200	7k File header
7300	7k Trigger
7301	7k Trigger sequence setup
7302	7k Trigger sequence done
7400	7k Time message
7500	7k Remote control

7501	7k Remote control acknowledge
7502	7k Remote control not acknowledge
7503	7k Remote control sonar settings
7600	7k Roll
7601	7k Pitch
7610	7k Sound Velocity
7611	7k Absorption loss
7612	7k Spreading loss
11000	Payload Controller — Command
11001	Payload Controller — Command Acknowledge
11002	Payload Controller — Alarm and Status
11200 – 11299	Reserved: Payload Controller — Sensor QC records

RECORD TYPE #1000 `Reference point information'

DESCRIPTION:
Reference point information.

DATA DEFINITION:

DRF	RTH	RD	OD	DRF

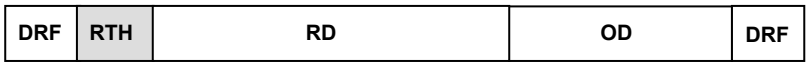
NAME	SIZE	DESCRIPTION
Vehicle's X reference point to Center of Gravity	f32	X offset in meter.
Vehicle's Y reference point to Center of Gravity	f32	Y offset in meter.
Vehicle's Z reference point to Center of Gravity	f32	Z offset in meter.
Water level to Center of Gravity	f32	In meters.

Note: for submersible vehicles, since the vertical offset from the COG to the water level is not fixed, the offsets should be set to zero. Typically the offsets to the depth sensor, combined with the reported depth at the sensor and the vehicle attitude would be used to determine the depth of the COG and reference point.

RECORD TYPE #1001 `Sensor offset position information'

DESCRIPTION:
Sensor position offset information data (non-calibrated).

DATA DEFINITION:

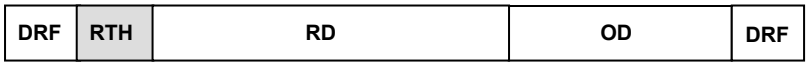


NAME	SIZE	DESCRIPTION
Sensor position X offset	f32	X offset from vehicle reference point in meters.
Sensor position Y offset	f32	Y offset from vehicle reference point in meters.
Sensor position Z offset	f32	Z offset from vehicle reference point in meters.
Sensor roll angle offset	f32	Roll angle offset in radians.
Sensor pitch angle offset	f32	Pitch angle offset in radians.
Sensor yaw angle offset	f32	Yaw angle offset in radians.

RECORD TYPE #1002 `Calibrated sensor offset position information'

DESCRIPTION:
Calibrated sensor position offset information data.

DATA DEFINITION:



NAME	SIZE	DESCRIPTION
Sensor position X offset	f32	X offset from vehicle reference point in meters.
Sensor position Y offset	f32	Y offset from vehicle reference point in meters.
Sensor position Z offset	f32	Z offset from vehicle reference point in meters.
Sensor roll angle offset	f32	Roll angle offset in radians.
Sensor pitch angle offset	f32	Pitch angle offset in radians.
Sensor yaw angle offset	f32	Yaw angle offset in radians.

RECORD TYPE #1003 `Position'

DESCRIPTION:
Position record.

DATA DEFINITION:

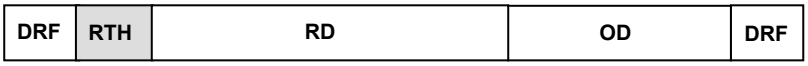
DRF	RTH	RD	OD	DRF

NAME	SIZE	DESCRIPTION
Datum identifier	u32	0 – WGS84. >0 – Reserved.
Latitude	f64	Latitude in radians.
Longitude	f64	Longitude in radians.
Height relative to Datum	f64	In meters.

RECORD TYPE #1004 `Attitude'

DESCRIPTION:
Attitude data record. The length of each data field is dynamic which is based on the field mask.

DATA DEFINITION:



NAME	SIZE	DESCRIPTION
Field mask	u8	BITFIELD 0: Pitch in radians (f32). 1: Roll in radians (f32). 2: Heading in radians (f32). 3: Heave in meters (f32). 7-4: Reserved.
Reserved	u8	Reserved field.
N	u16	Number of fields.
Frequency	f32	Sample rate in samples / second.

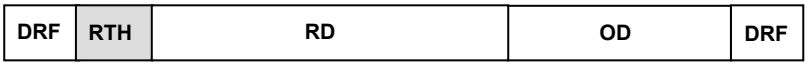
NAME	SIZE	DESCRIPTION
FIELD 0	variable	Sensor data.
…	…	...
FIELD N-1	variable	Sensor data.

RECORD TYPE #1005 `Tide'

DESCRIPTION:

Tide data record.

DATA DEFINITION:

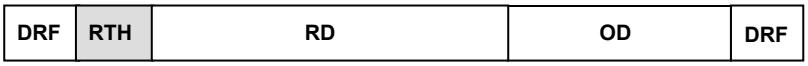


NAME	SIZE	DESCRIPTION
Tide	f32	Height correction above mean sea level in meters.
Source	u16	0 – Table. 1 – Gauge.
Reserved.	u16	Reserved.

RECORD TYPE #1006 `Altitude'

DESCRIPTION:
Altitude data record.

DATA DEFINITION:

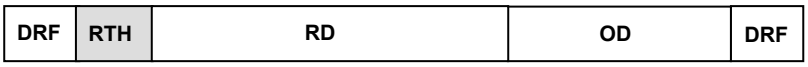


NAME	SIZE	DESCRIPTION
Distance	f32	Distance from seafloor in meters to sensor, positive up (0 at sea bottom).

RECORD TYPE #1007 `Motion over ground'

DESCRIPTION:
Motion over ground record. The length of each data field is dynamic which is based on the field mask.

DATA DEFINITION:



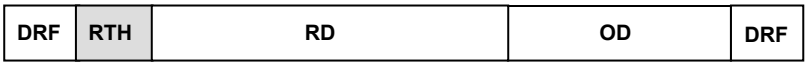
NAME	SIZE	DESCRIPTION
Field mask	u8	BITFIELD 0: X,Y,Z Speed (m/s) (f32,f32,f32). 1: X,Y,Z Acceleration (m/s^2) (f32,f32,f32). 7-2: Reserved.
Reserved	u8	Reserved field.
N	u16	Number of fields.
Frequency	f32	Sample rate in samples / second.

NAME	SIZE	DESCRIPTION
FIELD 0	variable	Sensor data.
…	…	...
FIELD N-1	variable	Sensor data.

RECORD TYPE #1008 `Depth'

DESCRIPTION:
Depth data record.

DATA DEFINITION:

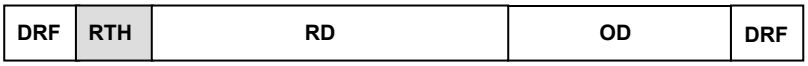


NAME	SIZE	DESCRIPTION
Depth descriptor	u8	0 – Depth to sensor. 1 – Water depth.
Correction flag	u8	0 – RAW depth (as measured). 1 – Corrected depth (relative to mean-sea level).
Reserved.	u16	Reserved.
Depth	f32	The deeper, the bigger (positive) this value becomes.

RECORD TYPE #1009 `Sound velocity profile'

DESCRIPTION:
Sound velocity profile data record.

DATA DEFINITION:



NAME	SIZE	DESCRIPTION
Position flag	u8	0 – Invalid position fields. 1 – Valid position fields.
Reserved.	u8	Reserved.
Reserved.	u16	Reserved.
Latitude	f64	Latitude in radians (WGS84).
Longitude	f64	Longitude in radians (WGS84).
N	u32	Number of samples.

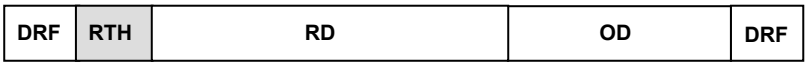
NAME	SIZE	DESCRIPTION
SAMPLE 0 Depth	f32	In meters.
SAMPLE 0 Sound velocity	f32	In meters / second.
…	…	...
SAMPLE N-1 Depth	f32	In meters.
SAMPLE N-1 Sound velocity	f32	In meters / second.

RECORD TYPE #1010 `CTD'

DESCRIPTION:

CTD data record.

DATA DEFINITION:



NAME	SIZE	DESCRIPTION
Sound velocity source flag	u8	0 – Not computed. 1 – CTD. 2 – User computed.
Sound velocity algorithm	u8	0 – Not computed. 1 – Checn Millero. 2 – Delgrosso.
Conductivity flag	u8	0 – Conductivity. 1 – Salinity.
Pressure flag	u8	0 – Pressure. 1 – Depth.
Position flag	u8	0 – Invalid position fields. 1 – Valid position fields.
Reserved.	u8	Reserved.
Reserved.	u16	Reserved.
Latitude	f64	Latitude in radians (WGS84).
Longitude	f64	Longitude in radians (WGS84).
Sample rate	f32	Sample rate.
N	u32	Number of samples.

NAME	SIZE	DESCRIPTION
SAMPLE 0 Conductivity / Salinity	f32	In S/m or ppt.
SAMPLE 0 Water temperature	f32	In Celsius.
SAMPLE 0 Pressure / Depth	f32	In Pascal or meters.
SAMPLE 0 Sound velocity	f32	In meters / seconds.
…	…	...
SAMPLE N-1 Conductivity / Salinity	f32	In S/m or ppt.
SAMPLE N-1 Water temperature	f32	In Celsius.
SAMPLE N-1 Pressure / Depth	f32	In Pascal or meters.
SAMPLE N-1 Sound velocity	f32	In meters / seconds.

RECORD TYPE #1011 `Geodesy'

DESCRIPTION:

The Geodesy data record may be used to define the spheroid, datum and grid definitions for navigational data; each sequentially embedded within the RTH.

There are no dynamic data elements; the optional data portion of the record is used to contain custom projection parameters. The custom identifier field of the RTH identifies the various defined types. Moreover, this identifier may be -1 indicating that the optional data portion of the record contains user specific parameters.

DATA DEFINITION:

DRF	RTH	RD	OD	DRF

NAME	SIZE	DESCRIPTION
Spheroid name	ui8 * 32	A short text description of the spheroid name: e.g., "WGS84".
Semi-major axis	f64	Semi-major axis in meters: e.g., 6378137.0 for WGS84.
Inverse flattening	f64	Inverse flattening in meters: e.g. 298.257223563 for WGS84.
Reserved 1	ui8 * 16	Reserved space; should be zeroed.
Datum name	ui8 * 32	Datum name: e.g., "WGS84"
Data calculation method	ui32	0 – Molodensky 1 – Bursa / Wolfe 2 – DMA MRE 3 – NADCON 4 – HPGN 5 – Canadian National Transformation V2
Number of parameters	ui8	Seven (7) parameter transformation only supported; 9 parameter transformation to be added in later definitions.
DX	f64	X – Shift (m)
DY	f64	Y – Shift (m)
DZ	f64	Z – Shift (m)
RX	f64	X Rotation (degrees)
RY	f64	Y Rotation (degrees)
RZ	f64	Z Rotation (degrees)
Scale	f64	
Reserved 2	ui8 * 35	Reserved for later extension to 9 parameter transformation
Grid name	ui8 * 32	Name of grid system in use: e.g., "UTM"

Grid distance units	ui8	0 – Metres 1 – Feet 2 – Yards 3 – US Survey Feet 4 - Kilometres 5 – Miles 6 – US Survey Miles 7 – Nautical Miles 8 – Chains 9 – Links
Grid angular units	ui8	0 – Radians 1 – Degrees 2 – Degrees, Minutes, seconds 3 – Gradians 4 – Arc-seconds
Latitude of Origin	f64	
Central Meridian	f64	
False Easting	f64	Meters.
False Northing	f64	Meters.
Central Scale Factor	f64	
Custom identifier	i32	Identifier for optional field definition in 7k record. Used to define projection specific parameters. -2 – Custom -1 – Not used
Reserved 3	ui8 * 50	-

Appendix E provides a list of Customer Identifier values currently reserved.

RECORD TYPE #1500 - 1599 Reserved for future QC records

DESCRIPTION:

TBD

DATA DEFINITION:

TBD

RECORD TYPE #3000 `EdgeTech FS-DW side scan data record'

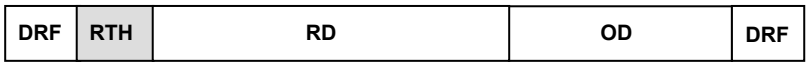
DESCRIPTION:
EdgeTech FS-DW side scan sonar data format.

The subsystem id field of the DRF is used to distinguish between multiple side scans subsystems of differing types such as low frequency and high frequency subsystems.

Each side scan channel imagery channel immediately follows the RTH and is in-term prefixed with its own CHANINFO structure. Port channel appears first followed by starboard.

The received EdgeTech data headers are appended (verbatim) to the end of the imagery data as optional data; sequenced port then starboard. The optional data Identifier field of the DRF determines the format of the EdgeTech header appended: 1 for sidescan data type[1]. Note that the EdgeTech weighting factor is not applied to the data but is present in the appended EdgeTech header.

DATA DEFINITION:



RECORD TYPE HEADER:

NAME	SIZE	DESCRIPTION
Millisecond time stamp	u32	Relative millisecond timer value.
Ping number	u32	Ping number as received from the EdgeTech subsystem.
Number of channels	u32	Number of imagery channels to follow (typically 2).
Total Bytes of channel data to follow	u32	Total bytes of channel data (and headers) to follow RTH (including optional data).
Data format	u32	Format of data: 0 – Envelope 1 – I and Q (complex)

Following the RTH is the concatenated imagery data each prefixed with a CHANNEL INFO header as follows:

[1] — Refer to appendix B for details of the EdgeTech data fields for the side scan sonar.

CHANNEL INFO

NAME	SIZE	DESCRIPTION
Channel number	u8	Channel number: 0 to Number of channels –1.
Channel type	u8	0 - port 1 - starboard
Data type	u8	0 - slant range 1 - ground range
Polarity	u8	0 – bipolar, 1 - unipolar
Bytes per sample	u8	Bytes per sample of the imagery.
Reserved 1	3 * u8	Reserved for future use.
Number of samples	u32	Number of samples in this channel.
Start time	u32	Start of first sample in microseconds relative to the ping time stamp.
Sample interval	u32	Data sample interval in microseconds.
Range	f32	Slant range or ground range in meters and depends on the data type field above.
Voltage (FSD)	f32	Analogue maximum amplitude. Should be –1 if not used.
Name	16 * i8	Channel name as a zero terminated character array.
Reserved 2	20 * u8	Padding and reserved fields.

RECORD TYPE #3001 `EdgeTech FS-DW Subbottom profiler data record'

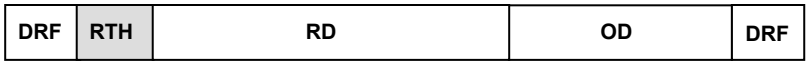
DESCRIPTION:
EdgeTech FS-DW sub-bottom profiler data record

Channel imagery immediately follows the channel data and is prefixed with a CHANINFO structure.

The received EdgeTech SEG-Y header is appended (verbatim) to the end of the imagery data as optional data. The optional data identifier field of the DRF determines the format of the EdgeTech header appended: 2 for SEG-Y data type[2].

Note:

- SBP data will typically be complex (representing I & Q per sample) thus 4 bytes per sample.
- The EdgeTech weighting factor is not applied to the data but is present in the appended EdgeTech header (refer to appendix B).



RECORD TYPE HEADER

NAME	SIZE	DESCRIPTION
Millisecond time stamp	u32	Relative millisecond timer value.
Ping number	u32	Ping number as received from the EdgeTech subsystem.
Number of channels	u32	Number of imagery channels to follow (typically 1).
Total Bytes of channel data to follow	u32	Total bytes of channel data (and headers) to follow this RTH including optional data.
Data format	u32	Sample format: 0 – Envelope 1 – I and Q (complex)

[2] — Refer to appendix B for details of the EdgeTech SEG-Y data format.

CHANNEL INFO

NAME	SIZE	DESCRIPTION
Channel number	u8	Channel number 0 to Number of channels –1.
Channel type	u8	0 – port 1 – starboard
Data type	u8	0 – slant range 1 – ground range
Polarity	u8	0 – bipolar 1 – unipolar
Bytes per sample	u8	Bytes per sample of imagery.
Reserved 1	3 * u8	Reserved for future use.
Number of samples	u32	Number of samples in this channel.
Start time	u32	Start of first sample in micro seconds relative to the ping time stamp.
Sample interval	u32	Data sample interval in microseconds.
Range	f32	Slant range or ground range in meters and depends on the data type field above.
Voltage (FSD)	f32	Analog maximum amplitude. Should be –1 if not used.
Name	16 * i8	Channel name as a zero terminated character array.
Reserved 2	20 * u8	Padding and reserved fields.

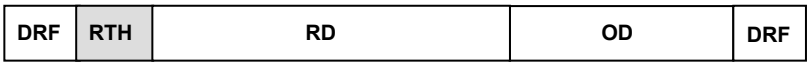
RECORD TYPE #3100 Framed BlueFin data frame record

DESCRIPTION:
This is a general container for various BlueFin data frames — each identified by the data format field of the RTH as defined below.

DATA DEFINITION:
The embedded data consists of a RTH, followed by a variable number of BlueFin defined data frames. For high record rates, multiple frames will be concatenated into a single record; the number will be rate dependent and typically represent 1 second of data.

For example, the navigation frames are output at nominal rate of 25 Hz — thus 25 frames representing 1 second of data will typically be contained by a single record.

Refer to appendix C for a definition of the record data frames.



RECORD TYPE HEADER

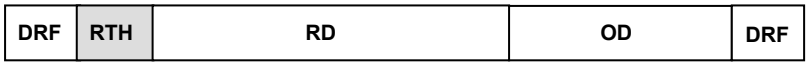
NAME	SIZE	DESCRIPTION
Time stamp of first frame	u32	Millisecond time stamp of first frame in record
Number of frames	u32	Number of frames embedded in this record.
Frame size	u32	Embedded frame size in bytes
Data format	u32	Data type identifier 0 – Navigation data 1 – Environment data
Reserved	16 * u8	Reserved.

RECORD TYPE #7000 `7k Volatile sonar settings'

DESCRIPTION:

SeaBat™ 7k volatile system settings for bathymetric, backscatter imagery and beam data outputs. This record is transmitted each ping.

DATA DEFINITION:



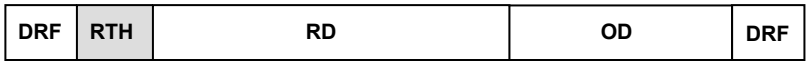
NAME	SIZE	DESCRIPTION
Sonar Id	u64	Serial number.
Ping number	u32	Sequential number.
Frequency	f32	Transmit frequency in Hertz.
Sample rate	f32	Sample in Hertz
Receiver bandwidth	f32	In Hertz.
Pulse width	f32	Seconds of pulse.
Pulse type	u32	0 - Rectangular.
Pulse reserved	u32	Additional pulse information.
Ping period	f32	Seconds since last ping.
Range selection	f32	Range selection in meters.
Power selection	f32	Power selection in dB//μPa
Gain selection	f32	Gain selection in dB.
Projector steering angle X	f32	In radians.
Projector steering angle Y	f32	In radians.
Projector –3dB beam width X	f32	In radians.
Projector –3dB beam width Y	f32	In radians.
Projector focal point	f32	In meters.

Control flags	u32	BITFIELD 3-0: Auto range method. 7-4: Auto bottom detect filter method. 8: Bottom detect range filter. 9: Bottom detect depth filter. 14-10: Auto receiver gain method. 31-15: Reserved.
Projector's magic number	u32	Projector selection.
Transmit flags	u32	BITFIELD 3-0: Pitch stabilization method. 7-4: Yaw stabilization method. 31-8: Reserved.
Hydrophone's magic number	u32	Hydrophone selection.
Receive flags	u32	BITFIELD 3-0: Roll stabilization method. 7-4: Dynamic focusing method. 11-8: Doppler compensation method. 15-12: Match filtering method. 19-16: TVG method. 31-20: Reserved.
Bottom detection filter info	f32	Min range (if range filter active).
Bottom detection filter info	f32	Max range (if range filter active).
Bottom detection filter info	f32	Min depth (if depth filter active).
Bottom detection filter info	f32	Max depth (if depth filter active).
Absorption	f32	Absorption in dB/km.
Sound velocity	f32	Sound Velocity in m/s
Spreading	f32	Spreading loss in dB.

RECORD TYPE #7001 `7k Configuration'

DESCRIPTION:
SeaBat™ 7k configuration record. Request only.

DATA DEFINITION:



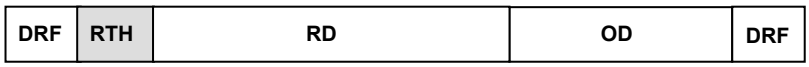
NAME	SIZE	DESCRIPTION
Sonar Id	u64	Sonar serial number
N	u32	Number of devices.

NAME	SIZE	DESCRIPTION
Module 0 magic number	u32	Unique identifier number.
Module 0 description	u8*64	ASCII string.
Module 0 serial number	u64	
Module 0 Info length	u32	In Bytes.
Module 0 info	dymanic	Varies with device type.
…	…	…
Module N-1 magic number	u32	Unique identifier number.
Module N-1 description	u8*64	ASCII string.
Module N-1 serial number	u64	
Module N-1 Info length	u32	In Bytes.
Module N-1 info	dymanic	Varies with device type.

RECORD TYPE #7002 `7k Match filter'

DESCRIPTION:
SeaBat™ 7k match filter settings record. Request only.

DATA DEFINITION:



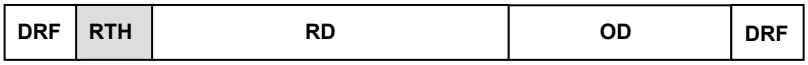
NAME	SIZE	DESCRIPTION
Sonar Id	u64	Sonar serial number
Ping number	u32	Sequential number.
TBD	TBD	TBD

RECORD TYPE #7004 `7k Beam geometry'

DESCRIPTION:

SeaBat™ 7k beam position and size data excluding backscatter imagery beams. X represent across track beams and Y along track beams. This record is sent on update, request.

DATA DEFINITION:



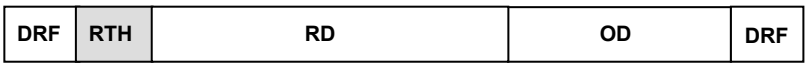
NAME	SIZE	DESCRIPTION
Sonar Id	u64	Sonar serial number
Rx	u32	Number of receiver beams.

NAME	SIZE	DESCRIPTION
Beam X direction angle[Rx]	f32*Rx	Angle in radians.
Beam Y direction angle[Rx]	f32*Rx	Angle in radians.
-3dB Beam width X[Rx]	f32*Rx	Angle in radians.
-3dB Beam width Y[Rx]	f32*Rx	Angle in radians.

RECORD TYPE #7005 `7k Calibration data'

DESCRIPTION:
SeaBat™ 7k calibration record. Request only.

DATA DEFINITION:



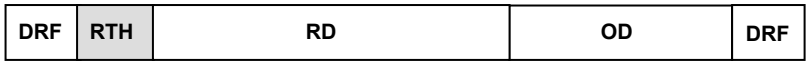
NAME	SIZE	DESCRIPTION
Sonar Id	u64	Sonar serial number.
Receivers	u16	Number of hydrophone receiver channels.

NAME	SIZE	DESCRIPTION
Receiver gain[receivers]	f32	Relative to a nominal gain of 1.0.
Receiver phase[receivers]	f32	Relative to a nominal phase of 0.0 radians.

RECORD TYPE #7006 `7k Bathymetric data'

DESCRIPTION:
SeaBat™ 7k multibeam Bathymetric output data. This record is transmitted each ping.

DATA DEFINITION:



NAME	SIZE	DESCRIPTION
Sonar Id	u64	Serial number.
Ping number	u32	Sequential number.
Rx	u16	Number of receiver beams.

NAME	SIZE	DESCRIPTION
Range[Rx]	f32*Rx	Two way travel time in seconds.
Quality[Rx]	u8*Rx	BITFIELD 3-0: Quality value (0 = bad 15 = best). 7-4: Reserved.
Intensity[Rx]	f32*Rx	Signal strength dB//μPa.

RECORD TYPE #7007 `7k Backscatter imagery data'

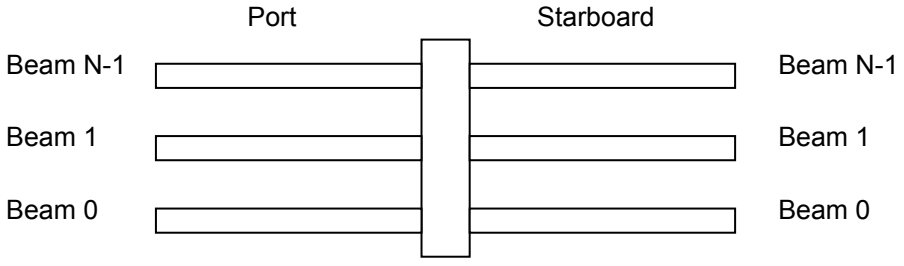
DESCRIPTION:

SeaBat™ backscatter imagery output data. One record holds one beam per side, i.e. a system with multiple beams per side sends multiple records. This record is transmitted for each beam pairs, every ping.

DATA DEFINITION:

DRF	RTH	RD	OD	DRF

NAME	SIZE	DESCRIPTION
Sonar Id	u64	Serial number.
Ping number	u32	Sequential number.
Beam position	f32	Meters forward from position of beam 0.
Control flags	u32	BITFIELD 3-0: Yaw stabilization method. 7-4: Beamforming method. 31-8: Reserved.
S	u32	Samples.
Port -3dB beam width Y	f32	In radians (typically a large angle).
Port -3dB beam width Z	f32	In radians (typically a small angle).
Starboard -3dB beam width Y	f32	In radians (typically a large angle).
Starboard -3dB beam width Z	f32	In radians (typically a small angle).
Port beam steering angle Y	f32	In radians (typically slightly positive).
Port beam steering angle Z	f32	In radians (typically pi).
Starboard beam steering angle Y	f32	In radians (typically slightly positive).
Starboard beam steering angle Z	f32	In radians (typically zero).
N	u16	Number of beams per side.
Current beam number	u16	Beam number of this record's data (0 to N-1).
W	u8	Number of bytes per sample.
Data types	u8	BITFIELD 0: Amplitude 1: Phase



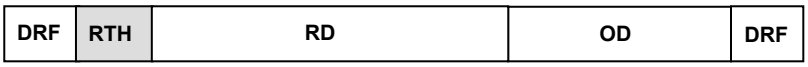
NAME	SIZE	DESCRIPTION
Port beam	W * S	Amplitude/Phase series. First sample represents range 0 meters.
Starboard beam	W * S	Amplitude/Phase series. First sample represents range 0 meters.

RECORD TYPE #7008 `7k Beam data'

DESCRIPTION:

SeaBat™ 7k beam data record is used for beamformed data output. This record is used for snippet output as well. Beams and samples are numbered from 0. First beam to last beam fields are always enumerated from low to high numbers. This record is transmitted each ping.

DATA DEFINITION:



NAME	SIZE	DESCRIPTION
Sonar Id	u64	Sonar serial number.
Ping number	u32	Sequential number.
Beams	u16	Total number of beams in record.
Reserved	u16	Reserved.
Samples	u32	Samples in ping. Only valid if all beams and samples are in record.
Record subset flag	u8	0 – All beams and samples in ping. 1 – Beam and / or sample ping subset.
Row column flag	u8	0 – Beam followed by samples. 1 – Sample follows by beams.
Reserved.	u16	Reserved.
Data sample type(s)	u32	(Lowest bits set is positioned at first position in data sample, etc.) BIT 1-4 Amplitude: 1 – Amplitude (8 bits) 2 – Amplitude (16 bits) BIT 5-8 Phase 1 – Phase (8 bits) 2 – Phase (16 bits) BIT 9-12 I and Q 1 – signed 16 bit I and signed 16 bit Q

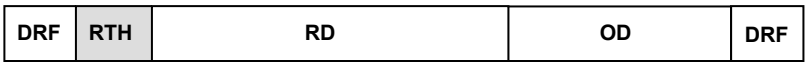
NAME	SIZE	DESCRIPTION
Beam descriptor	u16	Beam number.
Begin sample descriptor	u32	First sample in beam from transmitter and outward.
End sample descriptor	u32	Last sample in beam from transmitter and outward.
…	…	…
Beam descriptor	u16	Beam number.
Begin sample descriptor	u32	First sample in beam from transmitter and outward
End sample descriptor	u32	Last sample in beam from transmitter and outward.
First column / row	dynamic	Amplitude/Phase series. Array is populated with samples from transmitter and outward, or beams from low beam number and increasing.
…	…	…
Last column / row	dynamic	Amplitude/Phase series. Array is populated with samples from transmitter and outward, or beams from low beam number and increasing.

RECORD TYPE #7011 `7k Image data'

DESCRIPTION:

SeaBat™ 7k beam image record.

DATA DEFINITION:



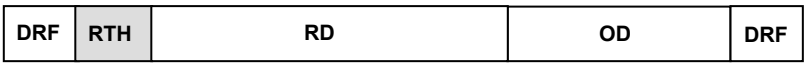
NAME	SIZE	DESCRIPTION
W	u32	Image width in pixels.
H	u32	Image height in pixels.
Color depth	u16	Color depth per pixel.
Width height flag	u16	0 – Width followed by height. 1 – Height followed by width.

NAME	SIZE	DESCRIPTION
First row / column	dynamic	Populated from left to right or from top to bottom.
…	…	…
Last row / column	dynamic	Populated from left to right or from top to bottom.

RECORD TYPE #7051 `7k System event'

DESCRIPTION:
SeaBat™ 7k alarm and status information.

DATA DEFINITION:



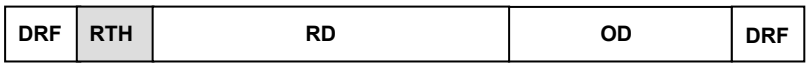
NAME	SIZE	DESCRIPTION
Sonar Id	u64	Sonar serial number.
Event Id	u16	0 – Success. 1 – Information. 2 – Warning. 3 – Error.
Message length	u16	Message length in Bytes.

NAME	SIZE	DESCRIPTION
Event message	dynamic	Null terminated string.

RECORD TYPE #7052 `7k Data storage status information'

DESCRIPTION:
SeaBat™ 7k data storage status information.

DATA DEFINITION:

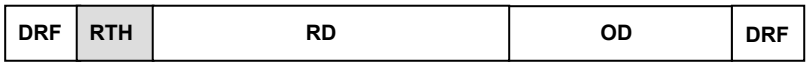


NAME	SIZE	DESCRIPTION
TBD	TBD	TBD

RECORD TYPE #7200 `7k File Header'

DESCRIPTION:
Optional data field can contain any customer specific information necessary to describe the file further.

DATA DEFINITION:



NAME	SIZE	DESCRIPTION
File identifier	u128	0xF3302F43CFB04d6fA93E2AEC33DF577D
Version number	u16	File format version number (1 – N).
Reserved	u16	Reserved.
Session identifier	u128	User defined session identifier. Used to associate multiple files for a given session.
Record data size	u32	Size of record data. 0 if not present.
N	u32	Number of subsystems. 0 if not set.

NAME	SIZE	DESCRIPTION
Recording name	u8*64	Null terminated US-ASCII string.
Recording program version number	u8*16	Null terminated US-ASCII string.
User defined name	u8*64	Null terminated US-ASCII string.
Notes	u8*128	Null terminated US-ASCII string.
Device Identifier 1	u32	Identifier for record type of embedded data.
Subsystem Identifier 1	u16	Identifier of the device that this data pertains.
System enumerator 1	u16	Identifier for the device subsystem.
...	...	...
Device Identifier N	u32	Identifier for record type of embedded data.
Subsystem Identifier N	u16	Identifier of the device that this data pertains.
System enumerator N	u16	Identifier for the device subsystem.

SEABAT 7k TRIGGER RECORDS HAS A RESERVED NUMBER RANGE FROM 7300 TO 7399.
TBD: Description.

RECORD TYPE #7300 `7k trigger'
DESCRIPTION:
SeaBat™ 7k software trigger record.

DATA DEFINITION:

DRF	RTH	RD	OD	DRF

NAME	SIZE	DESCRIPTION
Trigger Identifier	u16	Identifier for trigger input / output that shall be executed. See types 7301 and 7302 for details.

RECORD TYPE #7301 `7k trigger device configuration'

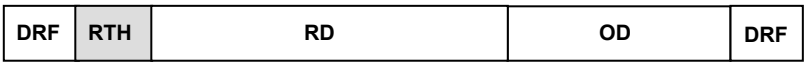
DESCRIPTION:

SeaBat™ 7k trigger device / software identifier device configuration. Use this record to map virtual trigger identifiers with hardware and software triggers.

DATA DEFINITION:

TRIGGER DEFINITION (TRIGGER_N see below)

NAME	SIZE	DESCRIPTION
Trigger type	u8	0 – In. 1 – Out. 2 – Neither.
Trigger category	u8	0 – Hardware. 1 – Software network. 2 – Software UI.
Trigger port	u16	Hardware – "BNC connector" TBD Software network – Out TCP port. Software UI – Reserved.
Trigger address	u32	Hardware – Pulse Type identifier TBD Software network – Out IP address. Software UI – Reserved.
Identifier	u16	Identifier used in sequence setup.



NAME	SIZE	DESCRIPTION
Triggers	u16	Number of triggers (0 – N).

NAME	SIZE	DESCRIPTION
TRIGGER_0	u64	See "TRIGGER DEFINITION"
…	…	…
TRIGGER_N	u64	See "TRIGGER DEFINITION"

RECORD TYPE #7302 `7k trigger steps sequence setup'

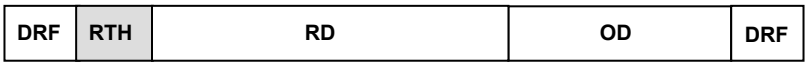
DESCRIPTION:

SeaBat™ 7k trigger steps sequence setup record. Use this record to define your trigger sequence. A Wait step takes the sequence into sleep mode where is waits for next input trigger to occur. An Execute step triggers an output (either hardware or software).

DATA DEFINITION:

TRIGGER STEP DEFINITION (STEP_N see below)

NAME	SIZE	DESCRIPTION
Type	u16	0 – (W) Wait step (Forces the sequence to wait for an input trigger). 1 – (E) Execute step (Output triggers).
Identifier	u16	Wait step – Input trigger identifier. Execute step – Output identifier (either HW or SW trigger).
Delay	f32	Delay in seconds since last step or input trigger.



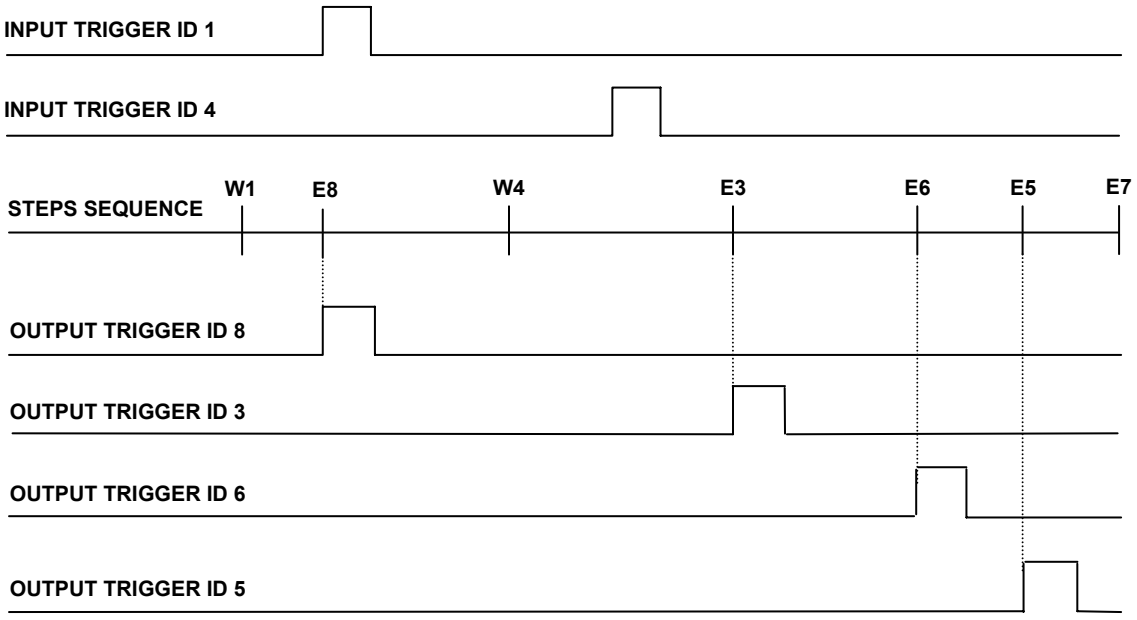
NAME	SIZE	DESCRIPTION
Steps	u16	Number of steps in sequence (0 – N).

NAME	SIZE	DESCRIPTION
STEP_0	u64	See "TRIGGER STEP DEFINITION"
…	…	
STEP_N	u64	See "TRIGGER STEP DEFINITION"

EXAMPLE

Wx – Wait step where x is the input trigger identifier.
Ex – Execute step where x is the output trigger identifier.

Step 1: Delay 0.0s, then wait for input #1.
Step 2: Delay 0.0s, then output trigger #8.
Step 3: Delay 0.1s, then wait for input #4.
Step 4: Delay 0.2s, then output trigger #3.
Step 5: Delay 0.3s, then output trigger #6.
Step 6: Delay 0.2s, then output trigger #5.
Step 7: Delay 0.2s, then goto step 1 (Trigger type 2 – "Neither")



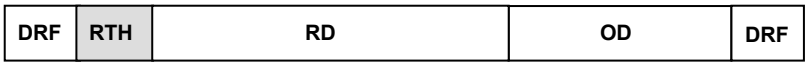
SEABAT 7k TIME RECORDS HAS A RESERVED NUMBER RANGE FROM 7400 TO 7499.

RECORD TYPE #7400 `Time message'

DESCRIPTION:

The time (7KTIME) in Data Record Frame represent the new time. This message can be used together with a PPS or equivalent. The leap second offset field can be used to flag for leap second inserts ahead of time.

DATA DEFINITION:



NAME	SIZE	DESCRIPTION
Leap second offset	i8	-1, 0 or +1 second for midnight 31 Dec.
Pulse flag	u8	0 – Message is not associated with hardware pulse. 1 – Message preceding hardware pulse. 2 – Message following hardware pulse.
Port identifier	u16	Port number identifier for pulse.
Reserved	u32	Reserved.
Reserved	u64	Reserved.

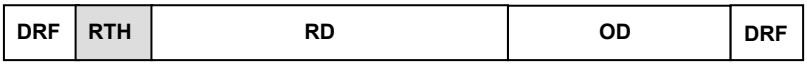
SEABAT 7k REMOTE CONTROL RECORDS HAS A RESERVED NUMBER RANGE FROM 7500 TO 7550.

RECORD TYPE #7500 `7k Remote control'

DESCRIPTION:

SeaBat™ 7k remote control record.

DATA DEFINITION:



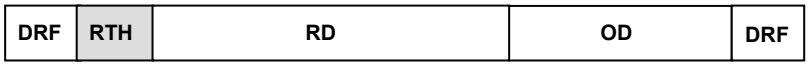
NAME	SIZE	DESCRIPTION
Remote control ID	u32	See separate remote control table for details. See Appendix D.
Ticket	u32	Ticker number. Incrementing number for control packet tracking.

NAME	SIZE	DESCRIPTION
Remote control data	dynamic	Value(s).

RECORD TYPE #7501 `7k Remote control acknowledge'

DESCRIPTION:
SeaBat™ 7k remote control acknowledge record.

DATA DEFINITION:



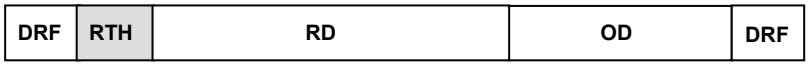
NAME	SIZE	DESCRIPTION
Ticket	u32	Ticker number in record 7500.

RECORD TYPE #7502 `7k Remote control not acknowledge'

DESCRIPTION:

SeaBat™ 7k remote control not acknowledge record.

DATA DEFINITION:



NAME	SIZE	DESCRIPTION
Ticket	u32	Ticker number in record 7500.
Error code.	u32	Error code, see table below.

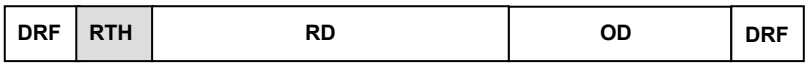
ERROR CODES:

NUMBER	DESCRIPTION
0	Reserved.
1	Rejected command.
2	Unknown command.

RECORD TYPE #7503 `7k Remote control sonar settings'

DESCRIPTION:
SeaBat™ 7k sonar control settings record.

DATA DEFINITION:



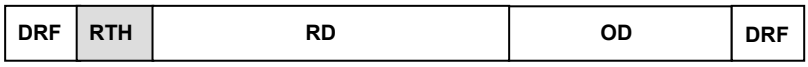
NAME	SIZE	DESCRIPTION
Frequency	f32	Transmit frequency in Hertz.
Sample rate	f32	Sample in Hertz
Pulse width	f32	Seconds of pulse.
Pulse type	u32	0 - Rectangular.
Pulse reserved	u32	Additional pulse information.
Ping period	f32	Seconds since last ping.
Range selection	f32	Range selection in meters.
Power selection	f32	Power selection in dB//μPa
Gain selection	f32	Gain selection in dB.
Projector steering angle X	f32	In radians.
Projector steering angle Y	f32	In radians.
Projector focal point	f32	In meters.
Control flags	u32	BITFIELD 3-0: Auto range method. 7-4: Auto bottom detect filter method. 8: Bottom detect range filter. 9: Bottom detect depth filter. 14-10: Auto receiver gain method. 31-15: Reserved.
Projector's magic number	u32	Projector selection.
Transmit flags	u32	BITFIELD 3-0: Pitch stabilization method. 7-4: Yaw stabilization method. 31-8: Reserved.
Hydrophone's magic number	u32	Hydrophone selection.
Receive flags	u32	BITFIELD 3-0: Roll stabilization method. 7-4: Dynamic focusing method. 11-8: Doppler compensation method. 15-12: Match filtering method. 19-16: TVG method. 31-20: Reserved.
Bottom detection filter info	f32	Min range (if range filter active).
Bottom detection filter info	f32	Max range (if range filter active).
Bottom detection filter info	f32	Min depth (if depth filter active).
Bottom detection filter info	f32	Max depth (if depth filter active).

RECORD TYPE #7600 `7k Roll'

DESCRIPTION:

SeaBat™ 7k roll value. Use this record to set system roll value.

DATA DEFINITION:



NAME	SIZE	DESCRIPTION
Roll	f32	In radians.

RECORD TYPE #7601 `7k Pitch'

DESCRIPTION:
SeaBat™ 7k pitch value. Use this record to set system pitch value.

DATA DEFINITION:

DRF	RTH	RD	OD	DRF

NAME	SIZE	DESCRIPTION
Pitch	f32	In radians.

RECORD TYPE #7610 `7k Sound Velocity'

DESCRIPTION:

SeaBat™ 7k sound velocity. Use this record to set system sound velocity value.

DATA DEFINITION:

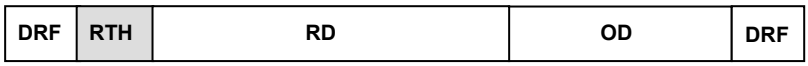
DRF	RTH	RD	OD	DRF

NAME	SIZE	DESCRIPTION
Sound velocity	f32	In meters / second.

RECORD TYPE #7611 `7k absorption loss'

DESCRIPTION:
SeaBat™ 7k absorption loss. Use this record to set system absorption loss value.

DATA DEFINITION:



NAME	SIZE	DESCRIPTION
Absorption loss	f32	In dB / km.

RECORD TYPE #7612 `7k spreading loss'

DESCRIPTION:

SeaBat™ 7k absorption loss. Use this record to set system spreading loss value. This coefficient value is used in conjunction with the absorption loss value to re-compute the TVG curve that will be applied to the returned signal.

DATA DEFINITION:

DRF	RTH	RD	OD	DRF

NAME	SIZE	DESCRIPTION
Spreading loss	f32	In dB (0.0 – 60.0).

RECORD TYPE # 11000 'Payload Controller Command'

DESCRIPTION:

This record type is used to send commands to the Payload Controller (PLC) or specified subordinate sensors for specific command and control purposes. Commands are routed using the appropriate sensor index identifiers.

Subordinate sensor commands and optional PLC command fields use an embedded ASCII variable length command data field; parameters are comma delimited fields. Floating point parameters shall use an ASCII "." to delimit the fractional portion of a number. Messages are case sensitive.

Command and message definitions are provided in appendix F of this document.

ASCII COMMAND MESSAGE SYNTAX

<COMMAND>[[,<PARAMETER 1>][,<PARAMETER_2>]… [,<PARAMETER_N>]]

DATA DEFINITION:

DRF	RTH	RD	OD	DRF

NAME	SIZE	DESCRIPTION
Sensor index	u32	Identifies the target for this command: -1 – PLC only >= 0 – Specified sensor identifier.
Command ID	u32	Command identifier (see RESON 6046 documentation).
Action	u32	Specifies whether the command is setting or retrieving state information: 0 – Unspecified 1 – Set 2 – Get (typically solicits a response) Note: Typically for a retrieve operation the optional ASCII COMMAND DATA field is not required.
Command bytes to follow	u32	Number of bytes in the command to follow and may be zero.
ASCII COMMAND DATA	Dynamic	Optional field specifying PLC or sensor ASCII command.

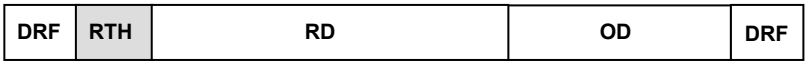
Note that the Sensor index field is a relative index used to identify the target of a specific command. The index is assigned by the PLC and will be a unique sequential number for each active sensor in the suite.

RECORD TYPE # 11001 'Payload Controller Command Acknowledge'

DESCRIPTION:

In verbose mode, the Payload Controller (PLC) will reply to all commands received from its remote host(s) in order to acknowledge command receipt and to indicate its success or failure. Conversely, the PLC will not reply to receive commands in quite mode. The message format is as follows:

DATA DEFINITION:



NAME	SIZE	DESCRIPTION
Command ID	u32	Command ID of the previously received command.
Sensor index	u32	Sensor index identifying the target sensor for the previous command (c.f. record #11000) received.
Command status	u32	Last command receive status: 0 – Command rejected 1 – Command accepted 2 – Unknown command.

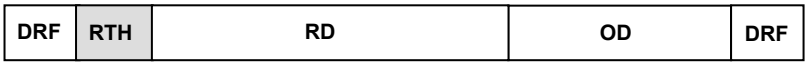
RECORD TYPE # 11002 'Payload Controller Alarm or Status'

DESCRIPTION:

This message will be sent to connected hosts either in response to a solicited enquiry or an unsolicited message due to a significant change in the system state such as an alarm condition.

Appendix F of this document gives the relevant status and alarm message identifiers and formats.

DATA DEFINITION:



NAME	SIZE	DESCRIPTION
Type	i32	Type of message: 0 – Status, 1 – Alarm
Alarm mode	i32	Alarm mode (relevant only to alarms): -1 = Unspecified, 0 = Alarm Clear 1 = Alarm Active
Identifier	i32	Message or alarm identifier (c.f. appendix F).
Message bytes	u32	Bytes to follow in dynamic portion (RD) of this message.

10. DEVICE IDENTIFIERS

IDENTIFIER	VENDOR	DESCRIPTION
1001	TrueTime	PCISG
2000	CDC	SMCG
2001	CDC	SPG
2002	Empire Magnetics	YS2000 Rotator
4013	Reson	TC4013
7001	Reson	SeaBat™ 7001
7012	Reson	SeaBat™ 7012
7100	Reson	SeaBat™ 7100
7101	Reson	SeaBat™ 7101
7102	Reson	SeaBat™ 7102
7112	Reson	SeaBat™ 7112
7123	Reson	SeaBat™ 7123 Subsystem identifiers: 0 – LFFDS 1 – MFFDS 2 – FCS
7125	Reson	SeaBat™ 7125
7128	Reson	SeaBat™ 7128
7150	Reson	SeaBat™ 7150
7160	Reson	SeaBat™ 7160
10000	TSS	DMS 05
10001	TSS	335B
10002	TSS	332B
10010	SeaBird	SeaBird SBE37
10020	Litton	Litton 200
11000	EdgeTech	EdgeTech FS-DW subsystems
11100	BlueFin	BlueFin vehicle controller
12000	Simrad	Simrad RPT319

11. APPENDIX A — WIN32 C/C++ API

TBD.

12. APPENDIX B — EDGETECH FS-DW FORMAT HEADERS

The following are the EdgeTech defined channel header structure definitions for the FS-AU series of sonars that detail the channel data formats pertaining to the SEG-Y and Sidescan formats (c.f.: record type #3000 and 3001, Format identifier field).

```
/* ------------------------------------------------------------------------ */
/* SidescanDefs.h                                                           */
/* ------------------------------------------------------------------------ */
/*                                                                          */
/* (c) Copyright 1999  EdgeTech,                                            */
/*                                                                          */
/* This file contains proprietary information, and trade secrets of         */
/* EdgeTech, and may not be disclosed or reproduced without the prior       */
/* written consent of EdgeTech.                                             */
/*                                                                          */
/* EdgeTech is not responsible for the consequences of the use or misuse    */
/* of this software, even if they result from defects in it.                */
/*                                                                          */
/* ------------------------------------------------------------------------ */
/*                                                                          */
/* EdgeTech sidescan data format description.                               */
/*                                                                          */
/* ------------------------------------------------------------------------ */

#ifndef __SIDESCANDEFS_H__
#define __SIDESCANDEFS_H__


/* ------------------------------------------------------------------------ */
/* includes                                                                 */
/* ------------------------------------------------------------------------ */

#include "SegyDefs.h"


/* ------------------------------------------------------------------------ */
/* Each record of data has a 80 byte header.  The content of which is       */
/* defined below.                                                           */
/*                                                                          */
/* Unused fields should be set to 0                                         */
/* ------------------------------------------------------------------------ */

typedef struct
{
  /*   0 -   1 : Subsystem (0 .. n)                                        */
  unsigned short int subsystem;

  /*   2 -   3 : Channel Number (0 .. n)                                   */
  unsigned short int channelNum;

  /*   4 -   7 : Ping number (increments with ping)                        */
  unsigned long int pingNum;

  /*   8 -   9 : Packet number (1..n) Each ping starts with packet 1       */
  unsigned short int packetNum;

  /*  10 -  11 : TriggerSource (0 = internal, 1 = external)               */
  unsigned short int trigSource;

  /*  12 -  15 : Samples in this packet                                    */
```

```
 unsigned long int samples;

/*  16 -  19 : Sample interval in ns of stored data                    */
unsigned long int sampleInterval;

/*  20 -  23 : starting Depth (window offset) in samples               */
unsigned long int startDepth;

/*  24 -  25 : -- defined as 2 -N volts for lsb                        */
short int weightingFactor;

/*  26 -  27 : Gain factor of ADC                                      */
unsigned short int ADCGain;

/*  28 -  29 : Maximum absolute value for ADC samples for this packet  */
unsigned short int ADCMax;

/*  30 -  31 : Range Setting (meters X 10)                             */
unsigned short int rangeSetting;

/*  32 -  33 : Unique pulse identifier                                 */
unsigned short int pulseID;

/*  34 -  35 : Mark Number (0 = no mark)                               */
unsigned short int markNumber;

/*  36 -  37 : Data format                                             */
/*   0 = 1 short  per sample  - envelope data                          */
/*   1 = 2 shorts per sample  - stored as real(1), imag(1),            */
/*   2 = 1 short  per sample  - before matched filter (raw)            */
/*   3 = 1 short  per sample  - real part analytic signal              */
/*   NOTE: For type = 1, the total number of bytes of data to follow is */
/*   4 * samples.  For all other types the total bytes is 2 * samples.  */
unsigned short int dataFormat;

/*  38 -  39 : Reserved field to round up to a 32-bit word boundary    */
unsigned short int reserved;


/* ------------------------------------------------------------------- */
/* computer date / time data acquired                                  */
/* ------------------------------------------------------------------- */

/*  40 -  43 : Millieconds today                                       */
unsigned long int millisecondsToday;

/*  44 -  45 : Year                                                    */
short int year;

/*  46 -  47 : Day of year (1 - 366)                                   */
unsigned short int day;

/*  48 -  49 : Hour of day (0 - 23)                                    */
unsigned short int hour;

/*  50 -  51 : Minute (0 - 59)                                         */
unsigned short int minute;

/*  52 -  53 : Second (0 - 59)                                         */
unsigned short int second;


/* ------------------------------------------------------------------- */
/* Auxillary sensor information                                        */
/* ------------------------------------------------------------------- */
```

```
    /*  54 -  55 : Compass heading (minutes)                                */
    short int heading;

  /*  56 -  57 : Pitch (minutes)                                            */
  short int pitch;

  /*  58 -  59 : Roll (minutes)                                             */
  short int roll;

  /*  60 -  61 : Heave (centimeters)                                        */
  short int heave;

  /*  62 -  63 : Yaw (minutes)                                              */
  short int yaw;

  /*  64 -  67 : Vehicle depth (centimeters)                                */
  unsigned long int depth;

  /*  68 -  69 : Temperature (degrees Celsius X 10)                         */
  short int temperature;

  /*  70 -  79 : Reserved for future use                                    */
  char reserved2[10];


  /* ---------------------------------------------------------------------- */
  /* Data area begins here                                                  */
  /* ---------------------------------------------------------------------- */
  /* Data begins at byte 80.                                                */
  /* short int data[];                                                      */
  /* ---------------------------------------------------------------------- */

} SidescanHeaderType;


/* ---------------------------------------------------------------------- */

#endif  /* end Not __SIDESCANDEFS_H__ */

/* ---------------------------------------------------------------------- */
/*                          end SidescanDefs.h                            */
/* ---------------------------------------------------------------------- */
```

```
 /* ------------------------------------------------------------------- */
 /* SegyDefs.h                                                          */
/* ------------------------------------------------------------------- */
/*                                                                     */
/* (c) Copyright 1997, 1998, 2000  EdgeTech,                           */
/*                                                                     */
/* This file contains proprietary information, and trade secrets of    */
/* EdgeTech, and may not be disclosed or reproduced without the prior  */
/* written consent of EdgeTech.                                        */
/*                                                                     */
/* EdgeTech is not responsible for the consequences of the use or misuse */
/* of this software, even if they result from defects in it.           */
/*                                                                     */
/* ------------------------------------------------------------------- */
/*                                                                     */
/* EdgeTech SEG-Y subbottom data format description.                   */
/*                                                                     */
/* ------------------------------------------------------------------- */

#ifndef __SEGYDEFS_H__
#define __SEGYDEFS_H__


/* ------------------------------------------------------------------- */
/* Values for trigger type field                                       */
/* ------------------------------------------------------------------- */

typedef enum
{
  TRIGGER_INTERNAL,           /* Internal trigger (0)                  */
  TRIGGER_EXTERNAL,           /* External trigger (1)                  */
} TriggerSourceType;


/* ------------------------------------------------------------------- */
/* Each record of data has a 240 byte header, the content of which is  */
/* defined below.  The structure is intended to be compatible with most */
/* implemented SEG-Y formats, as well as the original standard.        */
/*                                                                     */
/* Unused fields should be set to 0                                    */
/*                                                                     */
/* SEG-Y defined fields (** -> Highly Recommended in SEG-Y)            */
/* ------------------------------------------------------------------- */

typedef struct
{
  /* 0-3 : Trace Sequence Number (always 0) **                         */
  long int sequenceNumber;

  /* 4-7 : Starting depth (window offset) in samples.                  */
  unsigned long int startDepth;

  /* 8-11: Ping number (increments with ping) **                       */
  unsigned long int pingNum;

  /* 12-15 : Channel Number (0 .. n) **                                */
  unsigned long int channelNum;

  /* 16-27                                                             */
  short int unused1[6];

  /* 28-29 : ID Code (always 1 => seismic data) **                     */
  short int traceIDCode;

  /* 30-33                                                             */
  short int unused2[2];
```

```
 /* 34-35 : DataFormatType                                              */
/*   0 = 1 short  per sample  - envelope data                          */
/*   1 = 2 shorts per sample, - stored as real(1), imag(1),            */
/*   2 = 1 short  per sample  - before matched filter                  */
/*   3 = 1 short  per sample  - real part analytic signal              */
/*   4 = 1 short  per sample  - pixel data / ceros data                */
short int dataFormat;

/* 36-37 : Distance from towfish to antennae in cm                     */
short int NMEAantennaeR;

/* 38-39 : Distance to antennae starboard direction in cm              */
short int NMEAantennaeO;

/* 40-71 : Reserved for RS232 data - TBD                               */
char RS232[32];

/* ------------------------------------------------------------------- */
/* Navigation data :                                                   */
/* If the coorUnits are seconds(2), the x values represent longitude   */
/* and the y values represent latitude.  A positive value designates   */
/* the number of seconds east of Greenwich Meridian or north of the    */
/* equator.                                                            */
/* ------------------------------------------------------------------- */

/* 72-75 : Meters or Seconds of Arc                                    */
long int sourceCoordX;

/* 76-79 : Meters or Seconds of Arc                                    */
long int sourceCoordY;

/* 80-83 : mm or 10000 * (Minutes of Arc)                              */
long int groupCoordX;

/* 84-87 : mm or 10000 * (Minutes of Arc)                              */
long int groupCoordY;

/* 88-89 : Units of coordinates - 1->length (x /y), 2->seconds of arc  */
short int coordUnits;

/* 90-113 : Annotation string                                          */
char annotation[24];

/* 114-115 : Samples in this packet **                                 */
/* Note:  Large sample sizes require multiple packets.                 */
unsigned short int samples;

/* 116-119 : Sample interval in ns of stored data **                   */
unsigned long int sampleInterval;

/* 120-121 : Gain factor of ADC                                        */
unsigned short int ADCGain;

/* 122-123 : user pulse power setting (0 - 100) percent                */
short int pulsePower;

/* 124-125 : correlated data 1 - No, 2 - Yes                           */
short int correlated;

/* 126-127 : Starting frequency in 10 * Hz                             */
unsigned short int startFreq;

/* 128-129 : Ending frequency in 10 * Hz                               */
unsigned short int endFreq;
```

```
  /* 130-131 : Sweep length in ms                                        */
  unsigned short int sweepLength;

/* 132-139                                                              */
short int unused7[4];

/* 140-141 : alias Frequency (sample frequency / 2)                     */
unsigned short int aliasFreq;

/* 142-143 : Unique pulse identifier                                    */
unsigned short int pulseID;

/* 144-155                                                              */
short int unused8[6];

/* 156-157 : Year data recorded (CPU time)                              */
short int year;

/* 158-159 : day                                                        */
short int day;

/* 160-161 : hour                                                       */
short int hour;

/* 162-163 : minute                                                     */
short int minute;

/* 164-165 : second                                                     */
short int second;

/* 166-167 : Always 3 (other not specified by standard)                 */
short int timeBasis;

/* 168-169 :  weighting factor for block floating point expansion       */
/*            -- defined as 2 -N volts for lsb                          */
short int weightingFactor;

/* 170-171 :                                                            */
short int unused9;

/* -------------------------------------------------------------------- */
/* From pitch/roll/temp/heading sensor                                  */
/* -------------------------------------------------------------------- */

/* 172-173 : Compass heading (100 * degrees) -180.00 to 180.00 degrees  */
short int heading;

/* 174-175 : Pitch                                                      */
short int pitch;

/* 176-177 : Roll                                                       */
short int roll;

/* 178-179 : Temperature (10 * degrees C)                               */
short int temperature;

/* -------------------------------------------------------------------- */
/* User defined area from 180-239                                       */
/* -------------------------------------------------------------------- */

/* 180-181 : Heave compensation offset (samples)                        */
short int heaveCompensation;

/* 182-183 : TriggerSource (0 = internal, 1 = external)                 */
short int trigSource;
```

```
   /* 184-185 : Mark Number (0 = no mark)                                */
   unsigned short int markNumber;

  /* 186-187 : Hour                                                      */
  short int NMEAHour;

  /* 188-189 : Minutes                                                   */
  short int NMEAMinutes;

  /* 190-191 : Seconds                                                   */
  short int NMEASeconds;

  /* 192-193 : Course                                                    */
  short int NMEACourse;

  /* 194-195 : Speed                                                     */
  short int NMEASpeed;

  /* 196-197 : Day                                                       */
  short int NMEADay;

  /* 198-199 : Year                                                      */
  short int NMEAYear;

  /* 200-203 : Millieconds today                                         */
  unsigned long int millisecondsToday;

  /* 204-205 : Maximum absolute value for ADC samples for this packet    */
  unsigned short int ADCMax;

  /* 206-207 : System constant in tenths of a dB                         */
  short int calConst;

  /* 208-209 : Vehicle ID                                                */
  short int vehicleID;

  /* 210-215 : Software version number                                   */
  char softwareVersion[6];

  /* Following items are not in X-Star                                   */

  /* 216-219 : Initial spherical correction factor (useful for multiping /*/
  /* deep application) * 100                                             */
  long int sphericalCorrection;

  /* 220-221 : Packet number (1 - N) (Each ping starts with packet 1)    */
  unsigned short int packetNum;

  /* 222-223 : A/D decimation before FFT                                 */
  short int ADCDecimation;

  /* 224-225 : Decimation factor after FFT                               */
  short int decimation;

  /* 226-239                                                             */
  short int unuseda[7];

  /* ------------------------------------------------------------------- */
  /* Data area begins here                                               */
  /* ------------------------------------------------------------------- */
  /* Data begins at byte 240, has this.samples points in it              */
  /* short int data[];                                                   */
  /* ------------------------------------------------------------------- */
} SegyDataType;
```

```
  /* ------------------------------------------------------------------------ */

#endif  /* end Not __SEGYDEFS_H__ */

/* ------------------------------------------------------------------------ */
/*                              end SegyDefs.h                              */
/* ------------------------------------------------------------------------ */
```

13. APPENDIX C — BLUEFIN DATA FRAME DEFINITIONS

The following are the BlueFin defined data message types, relevant to record number 3100. Each message type below is follows the network / data frame specified below.

Network/Data Frame:

NAME	SIZE	DESCRIPTION
Packet Size	u32	Size in bytes of this packet including the header and appended data.
Version	u16	Version of this frame (e.g.: 1, 2, etc.)
Offset	u16	Offset, in bytes, to the start of data from the start of this packet.
Data Type	u32	Data Description. 1 = Navigation message 2 = Environment message
Data Size	u32	Size of data, in bytes.
Time Stamp	u8*10	7K Time Format, UTC.
Checksum	u32	CRC32 checksum for all bytes in record.
Reserved	u16	Reserved.
Data	Dynamic	Start of data section with either a partial or one or more records.

Navigation Message:

NAME	SIZE	DESCRIPTION
Quality Metrics	u32	NA
Latitude	f64	Radians
Longitude	f64	Radians
Speed	f32	Meters / Second
Depth	f64	Meters
Altitude	f64	Meters
Roll	f32	Radians
Pitch	f32	Radians
Yaw	f32	Radians
Northing Rate	f32	Meters / Second
Easting Rate	f32	Meters / Second
Depth Rate	f32	Meters / Second
Altitude Rate	f32	Meters / Second
Roll Rate	f32	Radians / Second
Pitch Rate	f32	Radians / Second
Yaw Rate	f32	Radians / Second
Position Time	f64	Seconds
Atitude Time	f64	Seconds

Navigation Message Quality Metrics:

NAME	BIT NUMBER	DESCRIPTION
Latitude	00 (LSB)	0 – Invalid, 1 – Valid
Longitude	01	0 – Invalid, 1 – Valid
Speed	02	0 – Invalid, 1 – Valid
Depth	03	0 – Invalid, 1 – Valid
Altitude	04	0 – Invalid, 1 – Valid
Roll	05	0 – Invalid, 1 – Valid
Pitch	06	0 – Invalid, 1 – Valid
Yaw	07	0 – Invalid, 1 – Valid
Northing Rate	08	0 – Invalid, 1 – Valid
Easting Rate	09	0 – Invalid, 1 – Valid
Depth Rate	10	0 – Invalid, 1 – Valid
Altitude Rate	11	0 – Invalid, 1 – Valid
Roll Rate	12	0 – Invalid, 1 – Valid
Pitch Rate	13	0 – Invalid, 1 – Valid
Yaw Rate	14	0 – Invalid, 1 – Valid
Position Time	15	0 – Invalid, 1 – Valid
Attitude Time	16	0 – Invalid, 1 – Valid
Undefined	17 – 31	NA

Environment Message:

NAME	SIZE	DESCRIPTION
Quality metrics	u32	See bit flag definition below.
Sound velocity	f32	Sound speed in m/s.
Reserved	u8 * 88	Padding and reserved space.

Environment Message Quality Metrics:

NAME	BIT NUMBER	DESCRIPTION
Sound velocity	0 (LSB)	0 – Invalid, 1 – Valid
Undefined	1 to 31	NA

14. APPENDIX D — 7K REMOTE CONTROL DEFINITIONS

SeaBat™ 7k remote control definitions.

IDENTIFIER	DESCRIPTION	POSSIBLE RETURN RECORDS
1000	Shutdown	7501, 7502
1001	Reboot	7501, 7502
1002	Calibrate	7501, 7502, 7005
1003	Range	7501, 7502
1004	Max ping rate	7501, 7502
1005	Transmit power	7501, 7502
1006	Transmit pulse width	7501, 7502
1007	Transmit pulse type	7501, 7502
1008	Receiver gain	7501, 7502
1009	Bottom detection method mask	7501, 7502
1010	Bottom detection filter info	7501, 7502
1011	Projector selection	7501, 7502
1012	Projector stabilization	7501, 7502
1013	Transmitter stabilization	7501, 7502
1014	Auto range	7501, 7502
1015	Hydrophone selection.	7501, 7502
1016	Projector steering	7501, 7502
1017	Receiver gain type	7501, 7502
1018	TVG coefficients	7501, 7502
1019	Auto receiver gain	7501, 7502
1050	Single record request	7501, 7502, 7000, 7001, 7002, 7004, 7005, 7006, 7007, 7008, 7051, 7052
1051	Volatile data feed	7501, 7502, 7000, 7001, 7002, 7004, 7005, 7006, 7007, 7008, 7051, 7052
1052	Stop volatile data feed.	7501, 7502
1053	Persistent data feed	7501, 7502, 7000, 7001, 7002, 7004, 7005, 7006, 7007, 7008, 7051, 7052
1054	Stop persistent data feed.	7501, 7502
1100	Start system	7501, 7502
1101	Stop system	7501, 7502

IDENTIFIER:
1000
NAME:
Shutdown
POSSIBLE RETURN RECORDS:
7501, 7502
DESCRIPTION:
Software and firmware halt followed by power shutdown to dry and wet hardware.
PARAMETERS:
None.

IDENTIFIER:
1001
NAME:
Reboot
POSSIBLE RETURN RECORDS:
7501, 7502
DESCRIPTION:
Software and firmware restart.
PARAMETERS:
None.

IDENTIFIER:
1002
NAME:
Calibrate
POSSIBLE RETURN RECORDS:
7501, 7502, 7005
DESCRIPTION:
Initiate system calibration. Record 7005 is returned upon successful calibration.
PARAMETERS:
None.

IDENTIFIER:
1003
NAME:
Range
POSSIBLE RETURN RECORDS:
7501, 7502
DESCRIPTION:
System range setting.
PARAMETERS:

NAME	SIZE	DESCRIPTION
Range	f32	Range setting in meters.

IDENTIFIER:
1004
NAME:
Max ping rate
POSSIBLE RETURN RECORDS:
7501, 7502
DESCRIPTION:
Max ping setting.
PARAMETERS:

NAME	SIZE	DESCRIPTION
Max ping rate	f32	Max ping rate setting in ping per second.

IDENTIFIER:
1005
NAME:
Transmit power
POSSIBLE RETURN RECORDS:
7501, 7502
DESCRIPTION:
System transmit power setting.
PARAMETERS:

NAME	SIZE	DESCRIPTION
Transmit power	f32	Transmit power in dB//uPa.

IDENTIFIER:
1006
NAME:
Pulse width
POSSIBLE RETURN RECORDS:
7501, 7502
DESCRIPTION:
System transmit pulse width setting.
PARAMETERS:

NAME	SIZE	DESCRIPTION
Transmit pulse width	f32	Transmit pulse width in seconds.

IDENTIFIER:
1007
NAME:
Pulse type
POSSIBLE RETURN RECORDS:
7501, 7502
DESCRIPTION:
System transmit pulse type.
PARAMETERS:

NAME	SIZE	DESCRIPTION
Transmit pulse type	u32	0 - Rectangular.

IDENTIFIER:
1008
NAME:
Receiver gain
POSSIBLE RETURN RECORDS:
7501, 7502
DESCRIPTION:
System receiver gain.
PARAMETERS:

NAME	SIZE	DESCRIPTION
Receiver gain	f32	Gain selection in dB.

IDENTIFIER:
1009
NAME:
Bottom detection mask
POSSIBLE RETURN RECORDS:
7501, 7502
DESCRIPTION:
System bottom detection mask.
PARAMETERS:

NAME	SIZE	DESCRIPTION
Bottom detection flag	u32	BITFIELD 3-0: Reserved. 7-4: Bottom detection method. 8: Range filter (0n / Off). 9: Depth filter (0n / Off). 31-10: Reserved.

IDENTIFIER:
1010
NAME:
Bottom detection filter info.
POSSIBLE RETURN RECORDS:
7501, 7502
DESCRIPTION:
System bottom detection filter info.
PARAMETERS:

NAME	SIZE	DESCRIPTION
Bottom detection filter info	f32	Min range (if range filter active).
Bottom detection filter info	f32	Max range (if range filter active).
Bottom detection filter info	f32	Min depth (if depth filter active).
Bottom detection filter info	f32	Max depth (if depth filter active).

IDENTIFIER:
1011
NAME:
Projector selection.
POSSIBLE RETURN RECORDS:
7501, 7502
DESCRIPTION:
System projector selection.
PARAMETERS:

NAME	SIZE	DESCRIPTION
Projector's magic number	u32	Projector selection.

IDENTIFIER:
1012
NAME:
Projector stabilization.
POSSIBLE RETURN RECORDS:
7501, 7502
DESCRIPTION:
System projector stabilization setting.
PARAMETERS:

NAME	SIZE	DESCRIPTION
Projector mask	u32	BITFIELD: 3-0: Pitch stabilization method. 7-4: Yaw stabilization method. 31-8: Reserved.

IDENTIFIER:
1013
NAME:
Transmit stabilization.
POSSIBLE RETURN RECORDS:
7501, 7502
DESCRIPTION:
System transmit stabilization setting.
PARAMETERS:

NAME	SIZE	DESCRIPTION
Transmit mask	u32	BITFIELD: 3-0: Roll stabilization method. 31-4: Reserved.

IDENTIFIER:
1014
NAME:
Auto range.
POSSIBLE RETURN RECORDS:
7501, 7502
DESCRIPTION:
System automatic range method setting.
PARAMETERS:

NAME	SIZE	DESCRIPTION
Auto range mask	u32	BITFIELD: 3-0: Auto range method. 31-4: Reserved.

IDENTIFIER:
1015
NAME:
Hydrophone selection.
POSSIBLE RETURN RECORDS:
7501, 7502
DESCRIPTION:
System hydrophone selection.
PARAMETERS:

NAME	SIZE	DESCRIPTION
Hydrophone's magic number	u32	Hydrophone selection.

IDENTIFIER:
1016
NAME:
Projector steering.
POSSIBLE RETURN RECORDS:
7501, 7502
DESCRIPTION:
System projector steering.
PARAMETERS:

NAME	SIZE	DESCRIPTION
Steering angle X	f32	In radians
Steering angle Y	f32	In radians.

IDENTIFIER:
1017
NAME:
Receiver gain type.
POSSIBLE RETURN RECORDS:
7501, 7502
DESCRIPTION:
System receiver gain type setting.
PARAMETERS:

NAME	SIZE	DESCRIPTION
Receiver gain type	u32	TVG method.

IDENTIFIER:
1018
NAME:
TVG coefficients.
POSSIBLE RETURN RECORDS:
7501, 7502
DESCRIPTION:
System TVG coefficients setting.
PARAMETERS:

NAME	SIZE	DESCRIPTION
TBD	TBD	TBD

IDENTIFIER:
1019
NAME:
Auto receiver gain.
POSSIBLE RETURN RECORDS:
7501, 7502
DESCRIPTION:
System automatic receiver gain setting.
PARAMETERS:

NAME	SIZE	DESCRIPTION
Auto receiver gain flag	u32	0 – Off. 1 – On.

IDENTIFIER:
1050
NAME:
Single record request.
POSSIBLE RETURN RECORDS:
7501, 7502, 7000, 7001, 7002, 7004, 7005, 7006, 7007, 7008, 7051 and 7052.
DESCRIPTION:
Request latest record.
PARAMETERS:

NAME	SIZE	DESCRIPTION
Record type	u32	Record number: 7000, 7001, 7002, 7004, 7005, 7006, 7007, 7008, 7051 or 7052.

IDENTIFIER:
1051
NAME:
Volatile data feed.
POSSIBLE RETURN RECORDS:
7501, 7502, 7000, 7001, 7002, 7004, 7005, 7006, 7007, 7008, 7051 and 7052.
DESCRIPTION:
Create volatile data feed.
PARAMETERS:

NAME	SIZE	DESCRIPTION
N	u32	Number of records.
Array of records.	N*u32	Record numbers: 7000, 7001, 7002, 7004, 7005, 7006, 7007, 7008, 7051 or 7052.

IDENTIFIER:
1052
NAME:
Stop volatile data feed.
POSSIBLE RETURN RECORDS:
7501, 7502
DESCRIPTION:
Stop volatile data feed.
PARAMETERS:
None.

IDENTIFIER:
1053
NAME:
Persistent data feed.
POSSIBLE RETURN RECORDS:
7501, 7502, 7000, 7001, 7002, 7004, 7005, 7006, 7007, 7008, 7051 and 7052.
DESCRIPTION:
Create persistent data feed.
PARAMETERS:

NAME	SIZE	DESCRIPTION
Destination IP	u32	Destination IP address.
Destination port number.	u16	UDP or TCP port.
Protocol flag.	u16	0 – UDP. 1 – TCP.
N	u32	Number of records.
Array of records.	N*u32	Record numbers: 7000, 7001, 7002, 7004, 7005, 7006, 7007, 7008, 7051 or 7052.

IDENTIFIER:
1054
NAME:
Stop persistent data feed.
POSSIBLE RETURN RECORDS:
7501, 7502
DESCRIPTION:
Stop persistent data feed.
PARAMETERS:

NAME	SIZE	DESCRIPTION
Destination IP	u32	Destination IP address.
Destination port number.	u16	UDP or TCP port.
Protocol flag.	u16	0 – UDP. 1 – TCP.

IDENTIFIER:
1100
NAME:
Start system.
POSSIBLE RETURN RECORDS:
7501, 7502
DESCRIPTION:
Starting the system enables pinging and the data flow.
PARAMETERS:
N/A

IDENTIFIER:
1101
NAME:
Stop system.
POSSIBLE RETURN RECORDS:
7501, 7502
DESCRIPTION:
Stopping the system disable pinging and the data flow.
PARAMETERS:

IDENTIFIER:
1102
NAME:
Load parameters.
POSSIBLE RETURN RECORDS:
7501, 7502
DESCRIPTION:
Loads system parameters from disk.
PARAMETERS:
N/A.

15. APPENDIX E — PROJECTION IDENTIFIERS AND DESCRIPTORS

The following table defines the reserved values for the custom identifier field of the Geodesy record (record number 1011). Definitions of projection specific parameters are TBD.

CUSTOM IDENTIFIER	PROJECTION
-1	Not used.
0	Universal Transverse Mercator (UTM)
1	Albers Equal-Area Conic
2	Azimuthal Equal Area
3	Azimuthal Equidistant
4	Bonne
5	Cassini
6	Double Stereographic
7	Equal-Area Cylindrical
8	Equidistant Conic
9	Equidistant Cylindrical
10	European Stereographic
11	Gnomic
12	Oblique Mercator (Rectified Skew Orthomorphic - with Skew Angle parameter)
13	Hotine
14	Hungarian National System (EOV)
15	Hungarian National System (EOV)
16	IMW Polyconic
17	Lambert Conformal Conic (1 parallel)
18	Lambert Conformal Conic (2 parallel)
19	Mercator
20	Miller Cylindrical
21	Mollweide
22	Orthographic
23	Polar Azimuthal
24	Equal Area
25	Polar Azimuthal Equidistant
26	Polar Stereographic
27	Polyconic
28	Robinson
29	Sinusiodal
30	Space Oblique Mercator
31	Stereographic
32	Stereographic 70
33	Transverse Mercator (Gauss-Kruger)
34	Two-Point Fit (polynomial projection)
35	Van der Grinten 1

16. APPENDIX F — RESON PLC COMMANDS AND MESSAGES

Table F.1 pertains to record type 11000 — the RESON Payload Controller remote command set.

Tables F.2 and F.3, pertain to record type 11002 — the RESON Payload Controller's status and alarm messages respectively. Here, the message type is determined by the "Type" field of the RTH of record number 11002. The tabulated ID field corresponds to the "Identifier" field of the RTH.

Note also that the format specifier field given in the following tables use a pseudo C programming language printf style format and are advisory only.

Table F.1 — RESON PLC Remote Commands.

Command	ID	Format	Description
Generic	0	%s	<command> - generic command.
Logging	1	%d	<flag> - logging enable (≠ 0) disable (0).
Switch	2	%s	<name> - changes the logging file automatically ("Auto") based on size or to the specified name otherwise.
Path	3	%s	Sets the logged data path to the specified folder.
Version	4	-	Solicits a PLC version message.
Load	5	-	Load the PLC run list from stored 6046.sav file.
Save	6	-	Save current run list to the 6046.sav file.
Subscribe	7	%d, %lu	<sensor id, record type> - allows a client to subscribe to a particular 7k data message for a given sensor id on a per client basis.
Unsubscribe	8	%d, %lu	<sensor id, record type> - withdraws subscription to a 7k data message for the specified sensor and for the given client.
Alarm	9	%d, %d	<alarm id, action> - disable (0) or enable (1) a given alarm on a per client basis.
Status	10	%d	<sensor id> - solicits a subsequent status message.
Verbose	11	%d	<flag> - disables (0) or enables (1) verbose mode. Disabled is synonymous with quiet mode.
Reset	12	%d	<sensor id> - resets the given sensor back to its default state. This command is reserved for future use.
Shutdown	13	%d	<shutdown type> - issues a shutdown request to the PLC (any) and optionally the OS (1) on which the PLC is running. When 1, shutdown will also be issued to subordinate sensors where they are capable of doing so.
Modules	14	-	Solicits the list of currently supported sensor modules.

Table F.1 — RESON PLC Remote Commands (continued)

Run list	15	-	Solicits the list of sensors that are currently loaded.
Add	16	%d, %d, %d, %s	<module number, subsystem id, port number, sensor name> - adds a sensor to the run list and executes the sensor's startup procedure.
Remove	17	%d	Removes a sensor from the run list and executes the sensor's shutdown procedure.
Report logging status	18	-	Solicits PLC's logging status message (file details, table F.2, id 3).
Reserved	19	-	Reserved for future use.
Time sync	20	%04u/%02u/%02u,%02u:%02u:%8.5f	<year, month, day, hour, minute, seconds> - grossly disciplines the PLC's clock to the specified time.
Health	21	-	Solicits a health message as defined by identifier 6 of table F.2.
Auto start	22	%lu	<mode> - disables (0) or enables (1) logging and active sensors on start up.
Auto health check	23	%d, %lu	<enable, period> - enables (1) or disables (0) the internal PLC auto health check of subordinate sensors at the specified period (ms).
Route Messages	24	%d	<sensor id> - sets the sensor for which non-PLC based 7k messages will be routed when sent to the PLC by a remote client.
Report Alarms	25	-	Reports active alarms to the requesting client. Individual alarms are reported as successive 7k records (c.f. record # 11002 of [1]).
Logging setup	26	%lu, %lu or none	<file size, overlap> - used to either set or retrieve the PLC logging setup. When setting, the maximum file size and record overlap between successive log files are specified. Retrieval solicits the logging setup status record and does not require any supplied parameters.
Reserved	27	-	Reserved for RESON internal use.

Table F.2 — RESON PLC Status Messages

Status	ID	Format	Description
Version	0	%0.2f	<version>
Modules	1	Modules: %d N x (, Module = %d, %s, %.2lf	<N, N x (module #, file name, version)> - Number of modules (N) and details of each module supported; includes the module number, name and version number for each. Note: the format specifier is not printf compliant and advisory only.
Run list	2	Sensors: %d N x (,Sensor = %d, %d, %d, %d, %d, %d, %s)	<N, N x (sensor index, sensor id, sensor type, port, media, subsystem id)> - specifies the active run list for the sensor suite. Note: the format specifier is not printf compliant and advisory only.
File details	3	Logging = %d, %s, %.3lf	<enable flag, file name, file size> - specifies the current state of logging (0 – inactive, 1 – active) and, if active, the current fully qualified file name on the PLC system and its size in Mbytes.
Sensor reply	4	PulseFiles: %d, N x (, %s)	<N, N x (pulse name)> - Provides a list of the pulse names. In the case of the EdgeTech FS-DW, the pulse names are the pulse files. Note: the format specifier is not printf compliant and advisory only.
Logging setup	5	LoggingSetup = %lu, %lu, %s	
Health	6	Sensors: %lu N x (, %d, %d)	<number of sensors, N x (sensor id, health flag)> - health status for each sensor is provided. The health flag is either healthy (1) or unhealthy (0).

In addition to the above identifiers, the following are reserved for future use:

0 to 5,000 — General system status messages

Table F.3 — RESON PLC Alarm Messages

Alarm	ID	Format	Description
All clear	0	-	No system alarms are active at this time.
Disk space critical	1	%s	PLC system's disk is at or below critical remaining available space (usually ≤ 10% of capacity).
Sensor not responding	2	%d, %s	<sensor id, description> - a given sensor has not responded for a predefined period.
Ping dropped	3	%d, %ld, %ld	<sensor id, start ping number, stop ping number> - specified ping number or range of pings have been dropped by the subsystem. Note: -1 if field is invalid. E.g., "0, 200, -1": sensor (id 0) reports that ping 200 was dropped.
Sensor reported failure	4	%d, %d, %d, %s	<sensor id, cause, internal code, description> - an internal sensor error has been detected. An internal code of -1 means that the sensor code is unknown. Causes are: 0 — Unknown or unspecified 1 — Internal error code 2 — Service required 3 — Temperature out of range 4 — Humidity out of range
Sensor unhealthy	5	%d, %s	<sensor id, description> — the specified sensor has failed its health check.
Sensor failed to start	6	%d, %s	<sensor id, description> — the specified sensor has failed to correctly start up.
Sensor failed to shutdown	7	%d, %s	<sensor id, description> — the specified sensor has failed to correctly shut down.
Sensor failed to load data	8	%d, %s	<sensor id, description> — the specified sensor has failed to provide acquired data to PLC for logging.
Sensor failed to set time	9	%d, %s	<sensor id, description> — the specified sensor has failed to set the system time.
Sensor failed to route 7k message	10	%d, %d, %s	<sensor id, client index, description> — the specified client failed to route non-PLC 7k message to the specified sensor.
Sensor command failed	11	%d, %d, %s	<sensor id, client id, description> — the command addressed to the specified sensor by the given client failed to be invoked.
Sensor status retrieval failed	12	%d, %s	<sensor id, description> — the PLC failed to retrieve status information for specified sensor.
Reset failed	13	%d, %s	<sensor id, description> — the PLC failed to reset the specified sensor.
Invalid command	15	%d, %d, %s	<client index, command id, description> — the specified command from the given client was invalid thus ignored.

In addition to the above identifiers, the following are reserved for future use:

0 to 5,000 — General alarm messages
10,000 to 11,000 — QC-based alarm messages