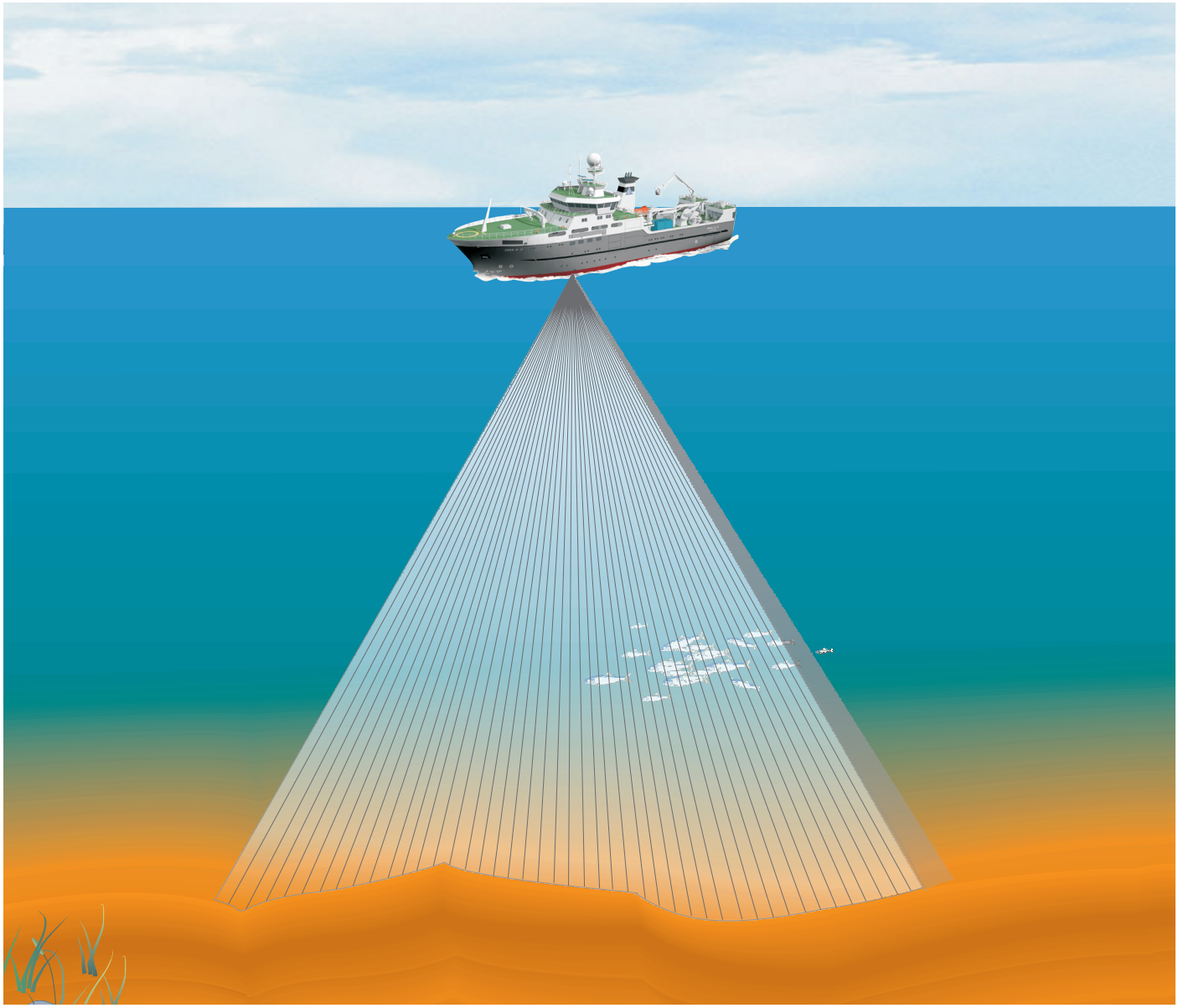
Reference manual

Simrad ME70
Scientific multibeam echo sounder



www.simrad.com

TECHNOLOGY FOR SUSTAINABLE FISHERIES

SIMRAD

KONGSBERG

Simrad ME70

Reference manual

Release 1.2.5

This manual provides you with reference information required to operate and fully understand the commands, menus, presentation modes and options provided by the Simrad ME70 Multibeam scientific echo sounder. For user information in brief, refer to the *Simrad ME70 Operator manual*.

328588/B

21.12.2012 © Kongsberg Maritime AS

Revision status

Document number: 328588 / Revision B		
Rev.A	01.02.2009	First version of manual. Release for SW version 1.1.1.
Rev.B	21.12.2012	Released for SW version 1.2.5. For details, see record of changes in *About this manual* on page 13

Copyright

©2012 Kongsberg Maritime AS
The information contained in this document remains the sole property of Kongsberg Maritime AS. No part of this document may be copied or reproduced in any form or by any means, and the information contained within it is not to be communicated to a third party, without the prior written consent of Kongsberg Maritime AS. The document, or any part of it, may not be translated to any other language without the written approval from Kongsberg Maritime AS.

Disclaimer

Kongsberg Maritime AS endeavours to ensure that all information in this document is correct and fairly stated, but does not accept liability for any errors or omissions.

Warning

The equipment to which this manual applies must only be used for the purpose for which it was designed. Improper use or maintenance may cause damage to the equipment and/or injury to personnel. All users must be familiar with the contents of the appropriate manuals before attempting to install, operate, maintain or in any other way work on the equipment. Kongsberg Maritime AS disclaims any responsibility for damage or injury caused by improper installation, use or maintenance of the equipment.

Support information

If you require maintenance or repair, contact your local dealer. You can also contact us using the following address: simrad.support@simrad.com. If you need information about our other products, visit http://www.simrad.com. On this website you will also find a list of our dealers and distributors.

Kongsberg Maritime AS
www.kongsberg.com

Raw data file format

The `*.raw` file exported from the ME70 may contain one or more of the following datagram types:

- Configuration
- NMEA
- Annotation
- Sample

Every `*.raw` file begins with two configuration datagrams: *CON0* and *CON1*. NMEA, annotation and sample datagrams constitute the remaining file content. These datagrams are written to the `*.raw` file in the order they are generated by the ME70.

Note that in the code lines, the information following the "//" characters are comments to that line.

Topics

- *Data encapsulation* on page 406
- *Configuration datagram* on page 407
- *NMEA datagrams* on page 408
- *Annotation datagrams* on page 409
- *Sample datagram* on page 409
- *Vessel coordinate system* on page 410
- *Numeric type definition* on page 411

Data encapsulation

A standard encapsulation scheme is used for all data files. Each datagram is preceded by a 4 byte length tag stating the datagram length in bytes. An identical length tag is appended at the end of the datagram.

```
long Length;
struct DatagramHeader
   {
   long DatagramType;
   struct {
      long LowDateTime;
      long HighDateTime;
   } DateTime;
   };
- -
< datagram content
- -
long Length;
};
```

All datagrams use the same header. The datagram type field identifies the type of datagram. ASCII quadruples are used to ease human interpretation and long term maintenance; three characters identify the datagram type and one character identifies the version of the datagram.

The *DateTime* structure contains a 64-bit integer value stating the number of 100 nanosecond intervals since January 1, 1601. This is the internal ”*filetime*” used by the Windows NT operating system. The data part of the datagram contains any number of bytes, and its content is highly datagram dependent.

Common computers fall into two categories:

- Intel based computers write a multibyte number to file starting with the LSB (Least Significant Byte).
- HP, Sun and Motorola do the opposite. They write the MSB (Most Significant Byte) to file first.

The byte order of the length tags and all binary fields within a datagram is always identical to the native byte order of the computer that writes the data file. It is the responsibility of the software that reads the file to perform byte swapping of all multibyte numbers within a datagram if required. Byte swapping is required whenever there is an apparent mismatch between the head and the tail length tags. Hence, the two length tags may be used to identify the byte order of the complete datagram.

The Intel processors allow a multibyte number to be located at any RAM address. However, this may be different on other processors; a short (2 byte) must be located at an even address, a long (4 byte) and a float (4 byte) must be located at addresses that can be divided by four. Hence, the numeric fields within a datagram is specified with this in mind.

Configuration datagram

All character strings are zero terminated.

“CON0” datagram

```
struct ConfigurationDatagram
{
   DatagramHeader DgHeader // ”CON0”
   ConfigurationHeader ConfigHeader;
   ConfigurationBeam Beam[];
};
{
   char cSurveyName[128]; // File name prefix
   char cTransectName[128]; // Not used
   char cSounderName[128]; // "MBES"
   char cVersion[30]; //
   short nMultiplexing; // Always 0
   long lTimeBias; // difference between UTC
      // and local time [minutes]
   float fSoundVelocityAverage; // [m/s]
   float fSoundVelocityTransducer; // [m/s]
   char cSpare[84];
   long lTransducerCount; // 1,2,3..
};
struct ConfigurationBeam
{
   char ChannelId[128];
   long BeamType; // See note 1 below
   float Frequency; // [Hz]
   float Reserved1; // Not used
   float EquivalentBeamAngle; // [dB]
   float BeamWidthAlongship; // [degree]
   float BeamWidthAthwartship; // [degree]
   float AngleSensitivityAlongship;
   float AngleSensitivityAthwartship;
   float AngleOffsetAlongship; // [degree]
   float AngleOffsetAthwartship; // [degree]
   float PosX; // future use
   float PosY; // future use
   float PosZ; // future use
   float BeamSteeringAngleAlongship; // [degree]
      // See note 2 below
   float BeamSteeringAngleAthwarthship; // [degree]
      // See note 3 below
   float BeamSteeringAngleUnused; // Not used
   float PulseLength; // [seconds]
   float Reserved2; // Not used in ME70
   char Spare1[8]; // future use
   float Gain; // [dB]
   float Reserved3; // Not used in ME70
   char Spare2[8]; // future use
   float SaCorrection; // [dB]
   float Reserved4; // Not used in ME70
   char Spare3[8];
   char GPTSoftwareVersion [16];
   char Spare4[28];
};
```

1 **BeamType**: This is a 32–bit integer, but only the last three bits are used.
   - Bit 0: 0 = Single beam, 1 = Split beam
   - Bit 1: 0 = Beam in fan, 1 = Reference beam
   - Bit 2: 0 = Reference beam “A”, 1 = Reference beam “B”

2 **BeamSteeringAngleAlongship**: Clockwise rotation around the Y axis.

   See *Vessel coordinate system* on page 410.

3 **BeamSteeringAngleAthwarthship**: Counter clockwise rotation around the X axis.

   See *Vessel coordinate system* on page 410.

“CON1” datagram

The *CON1* datagram contain the description of the beam mode used during the recording of the file. The description is in terms of an XML text string containing the information from the beam configuration XML formats.

```
struct ConfigurationDatagram
{
   DatagramHeader DgHeader; // "CON1"
   char BeamConfiguration[]; // xml string
};
```

Related topics

- *Beam configuration XML formats* on page 461

NMEA datagrams

The NMEA datagrams contain original NMEA 0183 input sentences.

```
struct TextDatagram{
   DatagramHeader DgHeader; // “NME0”
   char Text[]; // ”$GPGLL,5713.213,N.....”
};
```

The size of the datagram depends on the sentence length.

Related topics

- *About the NMEA telegram format* on page 383

Annotation datagrams

The annotation datagram contains comment text.

```
struct TextDatagram{
   DatagramHeader DgHeader; // “TAG0”
   char  Text[]; // ”Dangerous wreck”
};
```

The text string is zero terminated. The size of the complete datagram depends on the annotation length (maximum 80 characters).

Sample datagram

The sample datagram contains data from just one beam channel. It can contain power sample data, or it can contain both power and angle sample data.

```
struct SampleDatagram
   {
   DatagramHeaderDgHeader; // “RAW0”
   short Channel; // Channel number
   short Mode;
         // Datatype: 1 = Power, 2 = Angle, 3 = Both
   float TransducerDepth; // [m]
   float Frequency; // [Hz]
   float TransmitPower; // [W]
   float PulseLength; // [seconds]
   float BandWidth; // [Hz]
   float SampleInterval; // [s]
   float SoundVelocity; // [m/s]
   float AbsorptionCoefficient; // [dB/m]
   float Heave; // [m]
   float Roll; // [degree]
   float Pitch; // [degree]
   float Temperature; // [deg C]
   float Heading; // [degree]
   short TransmitMode;
      // 0 = Active, 1 = Passive
      // 2 = Test, -1 = Unknown
   char Spare[6];
   long Offset; // First sample
   long Count; // Number of samples
   short Power[]; // Compressed format - See below!
   short Angle[]; // See below!
   };
```

1 **Power**: The power data contained in the sample datagram is compressed. In order to restore the correct value(s), you must decompress the value according to the equation below.

$$y = x \frac{10 \log(2)}{256}$$

where:

x = power value derived from the datagram

y = converted value (in dB)

2 **Angle**: The fore-and-aft (alongship) and athwartship electrical angles are output as one 16-bit word. The alongship angle is the most significant byte while the athwartship angle is the least significant byte. Angle data is expressed in 2's complement format. The electrical angle must be multiplied with the angle sensitivity to get the target position relative to beam centre. Positive numbers denotes the fore and starboard directions.

Vessel coordinate system

The ME70 uses a right hand coordinate system with positive **X** forward, positive **Y** to starboard and positive **Z** down. This means that depth values are positive.

Angles are defined as clockwise rotation in the positive direction of the axis. 0 degrees is defined to be at the horizontal plane for angles around the X and Y axis (roll and pitch). For example, roll is around the x-axis, and a positive roll means that starboard side goes down.

There are no restrictions to where the coordinate system's origo is located in the **X** and **Y** directions.

Important

In the **Z** direction, the origo must be located on the water line.

Note

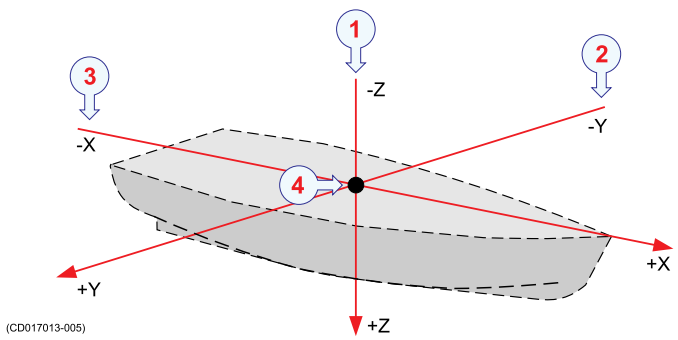
The sea surface with the vessel in normal trim defines the horizontal (X-Y) plane. The water line must therefore be marked on the hull with the vessel in normal trim before any dry docking.

1 *Downward (Z-axis)*

2 *Starboard (Y-azis)*

3 *Forward (X-axis)*

4 *Reference point (origo)*



Separate *Array* and *Vessel* coordinate systems are defined, but coincide with each other. Deviation due to inaccuracies during installation will be represented by minor differences between these two coordinate systems.

The centre beam of the ME70 fan is at 0 degrees athwart beam steering, beams towards the port side are located at angles of less than 0 degrees (negative values), while beams towards starboard side are located at angles higher than 0 degrees (positive values). A

vertical beam has alongship beam steering of 0 degrees. Beams pointing to the aft are located at angles of less than 0 degrees (negative values), while beams towards the bow are located at angles higher than 0 degrees (positive values).

Numeric type definition

In order to describe the data formats, common **C** structures are used to represent individual data blocks. The size of the various **C** types are:

- **char** = 8–bit integer
- **uchar** = 8-bit unsigned integer
- **short** = 16-bit integer
- **ushort** = 16-bit unsigned integer
- **WORD** = 16-bit unsigned integer
- **int** = 32-bit integer
- **long** = 32-bit integer
- **ulong** = 32-bit unsigned integer
- **DWORDLONG** = 64-bit unsigned integer
- **float** = 32-bit floating point (IEEE 754)
- **double** = 64-bit floating point (IEEE 754)

ISBN-13: -
©2012 Kongsberg Maritime AS