EK80 Raw format

Gavin Macaulay - Institute of Marine Research in Norway

Introduction

The EK80 raw format follows the same main format as the raw format for EK60 and most other Simrad scientific systems. This means that the raw file consists of a set of datagrams. Each datagram has the same encapsulation and tagging convention as existing raw files.

The differences are related to the introduction of new parameters and data implemented by introducing new datagrams which will be described in this document.

In EK80 XML datagrams (XML0) are used more frequently and three new datagrams (FIL1, MRU0, and RAW3) are introduced.

XML datagrams

In EK60 a raw file contains a configuration datagram (CON0), multiple sample datagrams (RAW0), and other datagrams (NME0 and TAG0).

In EK80 the XML datagrams (XML0) contain much of the information found in CON0 and RAW0. XML0 datagrams are introduced for much of the non sample data information as it offers more flexibility than binary datagrams with a fixed structure. As the XML datagrams are not as compressed as fixed binary datagrams these datagrams are not introduced for the actual sample data.

The XML0 datagram is simply an XML text string following the standard XML convention. The length of the XML string can be determined from the length of the datagram given in the datagram encapsulation.

The content of the XML string contains the actual information.

The name of the first node describes the type of data in the XML string. Currently the following information types of data are generated in EK80: Configuration, Environment, and Parameter.

The Configuration XML0 datagram in EK80 replaces the EK60 CON0 datagram.

The Environment XML0 datagram in EK80 replaces the environment information part of the EK60 RAW0 datagram.

The Parameter XML0 datagram in EK80 replaces the parameter information part of the EK60 RAW0 datagram.

Datagram sequence

The datagram sequence in an EK80 raw data file is not fixed and depends of numbers of installed frequency channels but will normally be similar to the following:

One XML0 - Configuration datagram

Two FIL1 - Filter datagrams for each frequency channel

One XML0 - Environment datagram

Then follows sample data for each channel. These data are comprised of two datagrams for each channel for each ping:

One XML0 – Parameter datagram

One RAW3 – Sample datagram

The XML0 - Parameter datagram and the RAW3 – Sample datagram are linked by the time stamp of the datagrams and the ChannelID information in the datagrams.

Asynchronous with the sample data the following datagrams can occur:

XML0 – Environment datagrams

NME0 – NMEA datagrams

TAG0 – Annotation datagrams

MRU0 – Motion datagrams

XML0 - Configuration datagram

The XML0 datagram having Configuration as the name of the first node is the first datagram in the EK80 raw file. The datagram contains the same information as the original CON0 datagram and some extra information.

The ChannelID information is the information which links data from different datagrams in the raw file to a specific frequency channel.

The attached XML file containing Configuration as part of its file name is an example extracted from a raw file and written into a separate file to illustrate the Configuration XML information.

FIL1 – Filter datagrams

The FIL1 datagrams contains filter coefficients used by EK80 for filtering the received signal. The filter coefficients in the FIL1 datagrams are used in combination with information of the transmitted signal to create the matched filter which can be used to create matched filter or pulse compressed echogram data.

The FIL1 datagram has the following content:

Stage:	short	// Filter stage number
Spare[2]:	char	
ChannelID[128]:	char	// Channel identification string
NoOfCoefficients:	short	// Number of complex filter coefficients
DecimationFactor:	short	// Filter decimation factor
Coefficients[]:	float	// Filter coefficients

The filter coefficients are complex values. Thus, the number of values found in Coefficients[] are 2*NoOfCoefficients since each complex filter coefficient consist of one real part and one imaginary part. The complex filter coefficients F(n) are arranged in Coefficients[] as

real(F(1)), imag(F(1)), real(F(2)), imag(F(2)), …. , real(F(NoOfCoefficients)), imag(F(NoOfCoefficients)

XML0 - Environment datagram

In EK60 the RAW0 datagram contains information about the sound speed and the absorption coefficient for the operating frequency. Since the EK80 allow operation covering a continuous frequency band a single value for the absorption coefficient is no longer adequate. In order to address this issue a new XML0 datagram having Environment as the name of the first node is introduced. This datagram contains the environment information from which the sound speed and frequency dependent absorption coefficient can be estimated using appropriate formulas.

The attached XML file containing Environment as part of its file name is an example extracted from a raw file and written into a separate file to illustrate the Environment XML information.

XML0 – Parameter datagram

In EK60 the RAW0 datagram contains information about a set of parameters at ping time needed to process the sample data correctly. Since the EK80 operational capabilities is expanded from just single frequency operation to other pulse forms more parameters to describe each ping are needed. Since these parameters could not be described within the existing RAW0 datagram a new XML0 datagram having Parameter as the name of the first node was introduced. This datagram contains information of all parameters at ping time for each channel.

MRU0 – Motion datagram

In EK60 the RAW0 datagram contains motion data for each ping. In EK80 motion information independent of ping time is supported by introducing a new asynchronous MRU0 motion datagram. In order to provide information also at ping time similar to EK60 currently an MRU0 datagram is generated together with parameter and sample data for each channel for each ping.

The MRU0 datagram has the following structure:

Heave: float

Roll: float

Pitch: float

Heading: float

RAW3 – Sample datagram

In EK60 the RAW0 datagram contains environment data, motion data, parameter data, and sample data for each ping. In EK80 the sample datagram has been reduced to containing only sample data as the remaining information is found in other datagrams.

In EK60 sample data from split beam systems consisted of received sample power and angle values. In EK80 the RAW3 datagram also supports complex sample values from each transducer quadrant/sector. Complex sample values are required for e.g. frequency analysis of data from other pulse forms than normal single frequency operation.

The RAW3 datagram has the following structure:

ChannelID[128]:	char	// Channel identification string
Datatype:	short	// Bit0 = Power, Bit1 = Angle,
		// Bit2 = ComplexFloat16, Bit 3 = ComplexFloat32,
		// Bit 8-10: # of Complex per Samples
Spare[2]:	char	
Offset:	long	// First sample number
Count:	long	// Number of samples
Samples[]:	byte	// Received sample values

The number of values in Samples[] depends on the value of Count and the value (bit values) in Datatype.

As an example a DataType value of 1032 means that Samples[] contains ComplexFloat32 samples and that each sample consists of 4 complex numbers (one from each of the 4 transducer quadrants). In this case Samples[] consists of 4*2*Count values of 32 bit floats, since each sample consists of 4 complex numbers each consisting of one real part and one imaginary part. The sample values S(i,n) are arranged as:

Real(S(1,1)), Imag(S(1,1)), Real(S(2,1)), Imag(S(2,1)), Real(S(3,1)), Imag(S(3,1)), Real(S(4,1)), Imag(S(4,1)), Real(S(1,2)), Imag(S(1,2)), ... , Real(S(4,Count)), Imag(S(4,Count))