Danish Geodata Agency

FAU Format Specifications

Version 1 - 18 November 2021

Table of Contents

FAU v1 - Introduction

The FAU (*FArvandsvæsenets UTM*) is a hydrographic binary file format.

The format was originally designed together with a sonar manufacturer to store data collected by one of the first commercial multibeam system. That system operated by simultaneously transmitting 4 beams (each separated by a fixed angle), shifting and repeating the transmission to create a whole swath of beams. This influenced the adopted data structure.

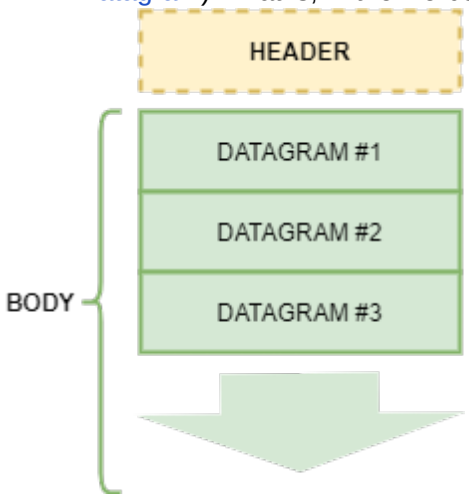
The optional header was introduced by geodesist Jørgen Eeg to facilitate communication among editing, visualization and analysis programs (i.e., Vise, Semi and Semantics).

Relevant contributions have also been provided by the EIVA software vendor.

File Layout

The FAU file is structured with:

- An optional header (see FAU v1 - Header).
- A body (see FAU v1 - Body), containing a number of repeated datagrams (see FAU v1 - Body Datagram). That is, in the file body there is a datagram for each depth measurement.



Data Organization

A FAU file can have one of the two following data organization:

- **Structured FAU** The file follows a strict ping-beam structure. The usual use case is the storage of multibeam raw soundings. For instance, if a multibeam collects 128 beams for each ping, then a FAU file storing 100 pings will have 128x100 = 12,800 datagrams. This rigid structure provides a mechanism to jump directly to a required beam of a ping.
- **Unstructured FAU** The file is simply a container of depth measurements (a point cloud). An example of use case for such a solution is the storage of a thinned/decimated version of all the raw soundings stored in a structured FAU file.

The format provides support for:

- **Depth Flagging** Identify depth measurements that have been flagged (i.e., selected) for potential rejection by manual or automated filtering. A depth measurement is still considered valid unless it is also marked as rejected.
- **Depth Rejection** Mark depth measurements that have been actually rejected and, thus, should not be used for derived products.

Format License

FAU v1 - Data Structure

Data Element Type

The following table describes the element types in use:

Type Symbol	Type Description
U8	Unsigned Char
S8	Signed Char
U16	Unsigned Short
S16	Signed short
U32	Unsigned Integer
S32	Signed Integer
U64	Unsigned Long
S64	Signed Long
F32	Float
F64	Double

FAU v1 - Header

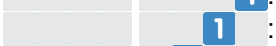
When present, the header of a FAU file is composed of the following fields:

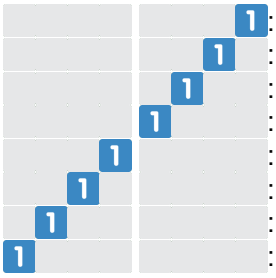
Field Name	Field Type	Field Description
`identity`	8 x S8	Bytes used to detect the endianness adopted by the file: • Little Endian: 'fau__uaf' • Big Endian: ‘_uaffau_’

`minilab`	20 x S8	A ‘minilabel’ containing the geospatial reference system in use:
		• *#[Projection]N[Horizontal Datum]*
		  • The first character is always '#'.
		  • Despite the format name, it is also possible to store data in projection different from UTM. For example:
		  • *mrc* Mercator
		  • *npstg* Polar stereographic
		  • *estg* Equatorial Stereographic
		  • *itm* Gauss Krueger
		  • *upsn* Universal Polar Stereographic North
		  • *upss* Universal Polar Stereographic South
		  • *dlmb* Lamberts Conical Two
		  • *lmb* Lamberts Conical One
		  • *sbf* System SBF (Denmark)
		  • *dks* System DKS (Denmark/Sweden)
		  • The eighth character sets the adopted Z-convention. The 'N' indicates “normal heights” (depths) with positive down.
		Example of mini-label:
		• '*#utm22nNwgs84*' identifies:
		  • *utm22n* UTM projection, Zone 22 North.
		  • *N* Depths (positive Z is down)
		  • *wgs84* WGS84 datum.

`version`	32 x S8	The version of the program used for converting to FAU format.
`conversion_time`	S32	The UNIX time in seconds when the file was generated.
`length`	S32	The size of the header in bytes. The value is always 768 (3x 256 bytes, and 32x 24 bytes - the size of the Body Datagram).
`ping_number`	U64	The ping number of the first ping converted into a FAU file. If not used/applicable, use 0.
`source`	S32	The source of the depth measurements: • 1 Database. • 2 SBD. • 3 SEMI. • 4 XYZ.
`kind`	S32	The type of depth measurements: • 1 Multibeam, with DGPS navigation. • 2 Multibeam, with RTK navigation. • 4 Singlebeam. • 8 Thinned. • 16 Not the primary detection layer.
`tide`	S32	A collection of bit fields for describing the approach adopted for vertical reduction: •  : 1 Data is corrected for tide. •  : 2 Delayed heave was applied. • : 4 Cross-section offset was applied.

`roll_offset`	F32	Calibration value for static roll in decimal degrees If not used/applicable, use 0.
`pitch_offset`	F32	Calibration value for static pitch in decimal degrees. If not used/applicable, use 0.
`heading_offset`	F32	Calibration value for static heading in decimal degrees. If not used/applicable, use 0.
`time_offset`	S32	Calibration value for time in milliseconds. If not used/applicable, use 0.
`edited_sensors`	S32	A collection of bit fields for describing the sensors with edited values: •  : 1 Roll. •  : 2 Pitch. •  : 4 Gyro. • : 8 Heave.
`sv_sensors`	S32	A collection of bit fields for describing the type of sound speed sensor: •  : 1 Transducer sensor. •  : 2 Sound speed profiler. •  : 4 Scanfish equipment. • : 8 Transducer sensor not working.
`sv_name`	512 x S8	The sound speed filename used for the stored data, with extension. If not used/applicable, the field is left empty.
`nr_of_beams`	S32	The number of beams for each ping/swath. It is mandatory for `kind=1` and `kind=2`; otherwise, use 0.
`nr_of_pings`	S32	The total number of pings/swaths in the file. If not used/applicable (e.g., unstructured FAU), use 0.
`bb_max_n`	S32	The maximum northing coordinate among the valid depth measurements, in centimeters.
`bb_min_n`	S32	The minimum northing coordinate among the valid depth measurements, in centimeters.
`bb_max_e`	S32	The maximum easting coordinate among the valid depth measurements, in centimeters.
`bb_min_e`	S32	The minimum easting coordinate among the valid depth measurements, in centimeters.
`bb_max_h`	S32	The maximum depth value among the valid depth measurements, in centimeters.
`bb_min_h`	S32	The minimum depth value among the valid depth measurements, in centimeters.
`track_heading`	F32	The average heading of the trackline, in decimal degrees. If not used/applicable, use 0.0.

`speed`	F32	The average speed of the trackline, in meter per second. If not used/applicable, use 0.0.
`roll_95`	F32	The interval in decimal degrees containing the 95% of the roll values. If not used/applicable, use 0.0.
`pitch_95`	F32	The interval in decimal degrees containing the 95% of the pitch values. If not used/applicable, use 0.0.
`heave_95`	S32	The interval in centimeters containing the 95% of the heave values. If not used/applicable, use 0.
`max_time_gap`	S32	This value and `ping_nr_max_time_gap` provides information about the correct functioning of the multibeam system: does it deliver the requested number of pings per second? The maximum time gap in stored in centiseconds. If not used/applicable, use 0.
`ping_nr_max_time_gap`	S32	This value and `max_time_gap` provides information about the correct functioning of the multibeam system: does it deliver the requested number of pings per second? If not used/applicable, use 0.
`ping_nr_pos_jump`	S32	Number of jumps between individual pings due to, for example, unstable positioning system. If not used/applicable, use 0.
`max_non_linearity`	S32	The largest numerical difference (counted in number of swaths) between any swath in the file and the idealized swaths in the bounding box. If not used/applicable, use 0.
`major`	S8	The major version number for Vise/MapSpikes.
`minor`	S8	The minor version number for Vise/MapSpikes.
`auto_flags`	S8	Field indicating whether an automatic flagging was used. • 0 No automatic flagging. • 1 An automatic flagging was used.
`rot_rect_valid`	S8	A collection of bit fields for describing the validity of the rotated bounding box: •  : 1 Valid rotated bounding box. •  : 2 Maximum non-linearity valid. •  : 4 Valid transducer depth.
`bb_tilt_x`	F64	The x-coordinate of the rotated bounding box, in centimeters.
`bb_tilt_y`	F64	The y-coordinate of the rotated bounding box, in centimeters.
`bb_tilt_w`	F64	The width of the rotated bounding box, in centimeters.
`bb_tilt_h`	F64	The height of the rotated bounding box, in centimeters.

`bb_tilt_ang`	F64	The rotation angle of the rotated bounding box, in decimal degrees.
`transducer_depth`	S32	The depth of the transducer, in centimeters. If not used/applicable, use 0.
`transmit_beam_width`	F32	Along-track TX beam width, in decimal degrees. If not used/applicable, use 0.0.
`swath_angle`	F32	The aperture of the swath, in decimal degrees. If not used/applicable, use 0.
`normalization`	S32	The UNIX time in seconds of the last performed channel normalization. A feature available with RESON systems for correcting the output of each analogue receiver channel for minor variations in amplitude and phase.
`bit_field`	S32	A collection of bit fields for describing the data in the FAU file:  • : 1 Roll stabilized. • : 2 Snippets. • : 4 Equiangle. • : 8 Equidistant. • : 16 Intermediate. • : 32 RESON Flex Mode. • : 64 Continuous Wave. • : 128 Frequency Modulated.
`frequency`	S16	The sonar frequency in KHz.
`database_id`	S64	An identifier for the source database. The Block Id in EIVA's NaviEdit database.
`spare`	10 x S8	Currently unused.

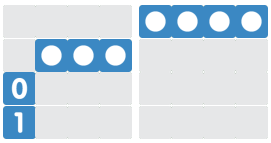
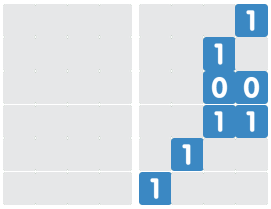
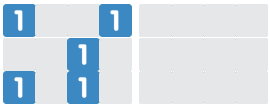
FAU v1 - Body

The body of a FAU file is composed of several FAU v1.0 - Body Datagrams, one for each depth measurement.

FAU v1 - Body Datagram

Each body datagram is composed of the following 11 fields:

Field Name	Field Type	Field Description
n	S32	The northing coordinate, in centimeters.
e	S32	The easting coordinate, in centimeters.
depth	S32	The depth value, in centimeters.
sec	S32	The UNIX time, in seconds.
angle	S16	The beam angle, in 0.01 degrees. The angle is positive at starboard.
heave	S8	The heave value, in 0.02 meters. The value is positive down.

`roll`	S8	The roll value, in 0.1 degrees. The angle is positive when the starboard side is down.
`quality`	U8	A value representing the quality of the depth measurement. Each bit in the value has a specific meaning:  • Values from 0 to 15 are quality indicators. • Bits reserved for flagging. • Valid. • (128) Rejected. Examples of usage: *EIVA implementation of the quality indicators for RESON echosounders:*  • *(1) Amplitude detection.* • *(2) Phase detection.* • *Not detected (neither amplitude nor phase).* • *Combined amplitude and phase detection.* • *(4) Passed Brightness Test.* • *(8) Passed Collinearity Test.* *Vise 14.3 implementation of the flagging/rejection:*  • *(144) Rejected by angle (e.g., >60 degrees).* • *(32) Only flagged by MapSpikes.* • *(160) Rejected based on MapSpikes.*
`amplitude`	S8	The signal amplitude associated with the depth measurement. The unit of measure is unspecified.
`pitch`	S8	The pitch value, in 0.1 degrees. The value is positive when the bow is up.
`centisec`	U8	The number of centiseconds to be added to the `sec` field.

The total size of a body datagram is always 24 bytes.

FAU v1 - Additional Historical Information

Before the release of ‘FAU Format Specifications v1’, there have been a number of variations on how the data have been stored in the FAU Format.

This section aims to describe the most common ones, without any insurance of completeness.

> The description of the variations is ONLY informative. Their use for newly collected data is discouraged.

Format Variants

This is a list of known format variants:

Format Extension	Description
.fag /.fal	The ‘g' in the extension indicates that the location of the measurements is in geographic coordinates. Similarly, the ‘l' was for 'Latitude/Longitude'.
.fas	The ‘s' in the extension indicates that the location of the measurements is in stereographic coordinates.

.fu2	The format differs from a regular .fau as follows:
	• It does not have a header (see FAU v1 - Header).
	• The intepretation of FAU v1 - Body Datagram differs as follows:
	  • The implicit roll multiplier is 0.2 (rather than 0.1).
	  • The 2-byte `Beam Angle` field is substituted by:
	  • 1-byte `Beam Number` (value stored as a `char`; thus its range is between -128 and 127).
	  • 1-byte `Mean Error of the Depth Relative to its Neighborhood` (value stored as an `unsigned char`; thus its range is between 0 and 255).

Field Variations

This is a list of observed field variations:

Field	Section	Description
`roll`	body	The implicit roll multiplier is 0.2 (rather than 0.1).