Garmin Recorded Sonar Data (RSD)

File Format

Herbert Oppmann

herby@memotech.franken.de

http://www.memotech.franken.de/FileFormats/

2024-03-19

Content

Garmin Recorded Sonar Data (RSD) File Format

Filename extension *.rsd

These files are created by Garmin ECHOMAP and some GPSMAP devices, and can be opened and viewed with HomePort [1].

This documentation is based on own research and the sources listed in the references section.

Basic data types

All values are serialized in little-endian byte order (least significant byte first).

Type	Length	Description
bool	1	0= false and 1= true
byte	1	8 bit unsigned integer (range 0 .. 255)
ushort	2	16 bit unsigned integer (range 0 .. 65535)
int	4	32 bit signed integer (range -2147483648 .. 2147483647)
uint	4	32 bit unsigned integer (range 0 .. 4294967295)
ulong	8	64 bit unsigned integer (range 0 .. 18446744073709551615)
float	4	32 bit floating point value according to [4].

Version:
A byte or ushort where the value is calculated as major version x 100 + minor Version.
E.g. 0x0064 = 100 = V1.0

GDate:
An uint, where the values 0 or 0xFFFFFFFF both mean "no value given". All other values have to be interpreted as unix time (see [3]), but with a starting value (epoch) of 1989-12-31 instead of 1970-01-01.
Note: While the number counts seconds, unix time is not defined as "number of seconds since the epoch", but "number of days since the epoch" x 24 x 60 x 60 + "number of seconds since midnight". The difference between the two is that unix time ignores leap seconds. The difference between unix time and UTC is already more than 20 seconds.

VarInt:
This is the same serialization as with Google Protocol Buffers. Note that the variable sized integer serialization format used in the Garmin Point of Interest (GPI) file format (described in my document Garmin_GPI_Format.pdf) is different.

VarUInt32:
Encoded as VarInt, internally stored in a uint (32 bit) and therefore limited to the respective value range.

VarInt32:
Zig-zag-encoded to a uint, then encoded as VarInt. Internally stored in a int (32 bit) and therefore limited to the respective value range.

CRC:
32 bit unsigned integer CRC value. The CRC algorithm is close to CRC-32, a.k.a. PKZIP or ITU-T Recommendation V.42 [5].

Polynomial:	0x04c11db7	
Reflect data:	true	
Reflect remainder:	true	
Final XOR value:	0xffffffff	
Initial remainder:	0	← Different! Normally 0xffffffff is used

MapUnit:
An int, whose value must be multiplied with 360.0° / (1 << 32) to get a coordinate (latitude or longitude).

General file structure

The file is divided into two areas:

Offset	Content
0x0000 - 0x4FFF	Area for serialization of the file header
0x5000 - end of file	Area for serialization of the list of sonar records

The serialization is very, very similar to Google Protocol Buffers (see [2] and [6]), but has some differences.

It is really a serialization, in the sense that there are no fixed offsets. The output is a linear stream of bytes. It consists of a compact sequence of data fields, where a lot of them are of variable size (like VarInt or string). Some fields are even optional and are just left out if not given. That means you can't start reading somewhere in the middle; you always have to read from the start. That's why HomePort, when opening an RSD file, reads and parses the complete file and creates an index file (see file format below) in the users home directory for enabling subsequent efficient random access to sonar records.

Serialization rules:

Serialization of a fixed structure

Type	Content
Number of fields x Field (Type according to the value)	Just the serialized field values in fixed order. There is no indication for the type of the field values within the file format. The application has to know the type and length for each field in advance.
optional:	
CRC	CRC over the complete serialization of the structure.

Serialization of a variable structure

Type	Content
VarUInt32	Number of fields that are following
Number of fields x Field	The serialized fields In theory, the fields could be in any order, because they can be identified by their field number. But usually the fields are in ascending order.

Type	Content
optional:	
CRC	CRC over the complete serialization of the structure.

Serialization of a variable structure field

Type	Content
VarUInt32	Key: (field_number << 3) + value_length The least significant three bits are value_length: 0: not used 1-6: the respective number of bytes 7: the length is >= 7, and the exact length is given in an additional VarUInt32 immediately following This is different to Google Protocol Buffers, where these bits are the wire_type. The upper bits are the field_number. This is important so the application knows which optional fields are left out. Also important if the format is enhanced, so that later added fields can be ignored.
VarUInt32	Length in bytes. Only present if value_length has the value 7.
Type according to the value	Value There is no indication for the type of the value within the file format. The application has to know the type for each field number in advance. Only the length of fields is known, so fields with unknown field numbers can be skipped. The type can also be a complex type (structure or array).

Serialization of an array

Type	Content
VarUInt32	Number of array elements
Number of array elements x value	Value There is no indication for the type here. The application has to know the type for the array elements. The type can also be a complex type (structure or array).

Serialization of an variable array (varray)

Type	Content
different	Number of array elements
VarUInt32	Overall length of all values in bytes.
Number of array elements x value	Value There is no indication for the type here. The application has to know the type for the array elements. The type can also be a complex type (structure or array).

File header area

Structure of the header area:

Content
Header structure
Filler, byte 0x00 up to end of header area

Header (variable structure with CRC)

Field nr.	Name	Optional?	Type	Content
0	magic_number	no	uint	= 0xD9264B7C
1	format_version	no	ushort	Seen value 0
2	channel_count	no	uint	Seen values 1, 4 and 6
3	max_channel_count	no	byte	Seen value 16
5	file_information	no	FileInformation structure	see below
6	channel_information_array	no	array[channel_count] of ChannelInformation structure	see below

FileInformation (variable structure)

Field nr.	Name	Optional?	Type	Content
0	unit_software_version	no	Version (ushort)	Seen values 3.10, 3.30, 4.40, 4.60, 5.60, 24.20
1	unit_id_type	no	uint	
2	unit_product_number	no	ushort	HWID as described in my document Garmin_BIN_Format.pdf
3	date_time_of_recording	no	GDate	

ChannelInformation (variable structure)

Field nr.	Name	Optional?	Type	Content
0	data_info	no	varray[VarUInt32] of DataInfo element	Array length seen: 1
1	first_chunk_offset	no	long	Absolute file offset to first real sonar record of this channel
2	prop_chan_info	no	varray[VarUInt32] of DpsChannelInformation structure	Array length seen: 1

DataInfo

Name	Type	Content
channel_id	VarUInt32	Values seen 0, 1, 2, 25, 33, 35, 38, 40, 41, 43, 45, 46, 47, 48, 49, 51, 54, 56, 81, 83, 86, 88, 95, 97, 98, 99

DpsChannelInformation (variable structure)

Field nr.	Name	Optional?	Type	Content
0	transducer_port	no	VarUInt32	Seen values 0 and 3
1	transducer_freq	no	TransducerFreq structure	
2	channel_capabilities	no	VarUInt32	Seen values 4, 0x3E, 0x74, 0x7E, 0x16BE, 0x16FE
3	gain_table	yes	array[] of int	Array length seen: 256 Negative values, steadily increasing, with 0 as the last value.

TransducerFreq (variable structure)

Field nr.	Name	Optional?	Type	Content
0	mode	no	VarUInt32	Seen values 1 and 3
1	start_freq	no	VarUInt32	Frequency in Hz Seen values see next line
2	end_freq	no	VarUInt32	Frequency in Hz Seen values for start_freq and end_freq: 140 – 240 kHz, 200 – 200 kHz, 245 – 275 kHz, 445 – 465 kHz, 770 – 840 kHz, 1000 – 1120 kHz

Example:

Byte(s) Meaning

Header Structure

06 VarUInt32 number of fields in header structure
04 VarUInt32 field 0 "magic_number", length 4
7c 4b 26 d9 uint
0a VarUInt32 field 1 "format_version", length 2
00 00 ushort
14 VarUInt32 field 2 "channel_count", length 4
01 00 00 00 uint
19 VarUInt32 field 3 "max_channel_count", length 1
10 byte
2f VarUInt32 field 5 "file_information", length 7 (actual length follows)
11 VarUInt32 length = 17
04 VarUInt32 number of fields in FileInformation structure
02 VarUInt32 field 0 "unit_software_version", length 2
4a 01 = ushort Version 330
0c VarUInt32 field 1 "unit_id_type", length 4

```
        25 1e 65 e7     uint
        12      VarUInt32 field 2 "unit_product_number", length 2
        82 06           ushort 1666 = echoMAP 50s/50dv or 70s/70dv
        1c      VarUInt32 field 3 "date_time_of_recording", length 4
        c6 15 4a 2e     GDate 2014-08-10 12:12:54
37      VarUInt32 field 6 "channel_information_array", length 7 (actual length follows)
        ad 08   VarUInt32 length = 1069
        01      VarUInt32 number of array elements
        array element 0 of ChannelInformation structure
        03      VarUInt32 number of fields
        03      VarUInt32 field 0 "data_info", length 3
                01      VarUInt32 number of array elements
                        array element 0 of DataInfo structure
                01      VarUInt32 length = 1
                01      VarUInt32 "channel_id"  = 1
        0f      VarUInt32 field 1 "first_chunk_offset", length 7 (actual length follows)
                08      VarUInt32 length 8
                31 50 00 00 00 00 00 00 long, points to the second record,
                i.e. the first record that has a body
        17      VarUInt32 field 2 "prop_chan_info", length 7 (actual length follows)
                9a 08   VarUInt32 length = 1050
                01      VarUInt32 number of array elements
                97 08   VarUInt32 length = 1047
                array element 0
                04      VarUInt32 number of fields of DpsChannelInformation structure
                01      VarUInt32 field 0 "transducer_port", length 1
                03      VarUInt32 = 3
                0f      VarUInt32 field 1 "transducer_freq", length 7 (actual length follows)
                        0b      VarUInt32 length = 11
                        03      VarUInt32 number of fields
                        01      VarUInt32 field 0 "mode", length 1
                        01              VarUInt32 = 1
                        0b      VarUInt32 field 1 "start_freq", length 3
                        c0 9a 0c        VarUInt32 = 200000
                        13      VarUInt32 field 2 "end_freq", length 3
                        c0 9a 0c        VarUInt32 = 200000
                11      VarUInt32 field 2 "channel_capabilities", length 1
                04              VarUInt32 = 4
                1f      VarUInt32 field 3 "gain_table", length 7 (actual length follows)
                82 08   VarUInt32 length = 0x402 = 1026
                80 02   VarUInt32 number of elements = 0x100 = 256 x int
                78 77 FF FF     int = -34952
```

02 78 FF FF int = -34814
…
72 FF FF FF int = -142
00 00 00 00 int = 0
AA BF 99 BA uint CRC over header structure

Filler

Bytes with value 0x00 until 0x4FFF

List of sonar records

Sonar records are in sequence without gaps.

Structure of each sonar record:

Content
Record header structure
Optional record body, consisting of - Record body structure - Sonar data
Record trailer structure

The first record of each channel/beam and optionally the last record have sequence_count and data_size=0 and therefore no body. All others do have a body.

Record header (variable structure with CRC)

Field nr.	Name	Optional?	Type	Content
0	magic_number	no	uint	= 0xB7E9DA86
1	state_data_info	no	StateData structure	see below
2	sequence_count	no	uint	For each channel/beam the counting starts with 0 and increases with each record. The last record may also have 0 here.
3	data_crc	no	uint	CRC over the record body (this is the body header and the sonar data). Is 0xFFFFFFFF (initial value) if sequence_count=0.
4	data_size	no	ushort	Size of the record body in byte (this is the body header and the sonar data). Is 0 if sequence_count=0.
5	recording_time_ms	no	uint	

StateData (variable structure)

Field nr.	Name	Optional?	Type	Content
0	state	no	VarUInt32	=1 for the first and last record which have no body =2 for the records inbetween which have a body
1	data_info	no	varray[VarUInt32] of VarUInt32 channel_id	Array length seen: 1

Body (variable structure)

Field nr.	Name	Optional?	Type	Content
0	channel_id	no	VarUInt32	
1	bottom_depth	no	VarInt32	probably in mm
2	drawn_bottom_depth	no	VarInt32	"
3	first_sample_depth	no	VarInt32	"
4	last_sample_depth	no	VarInt32	"
5	gain	no	byte	Offset in gain table?
6	sample_status	no	VarUInt32	Seen value 2
7	sample_cnt	no	uint	Seen values 745, 867, 947, 1002, 1004, 1106, 1221, 1284, 1396, 1477, 1612, 1647, 1653, 1717, 1817, 1930, 2048 and lots more
8	shade_avail	no	bool	Seen values 0 and 1
9	scposn_lat	no	int	In map units
10	scposn_lng	no	int	In map units
11	water_temperature	no	float	In degrees celsius
12	beam	yes	VarUInt32	Seen values 1-4
13	beam_info	yes	BeamInfo structure	
14	interrogation_id	yes	VarUInt32	
15	?	yes	StructUnknown1	

BeamInfo (variable structure)

Field nr.	Name	Optional?	Type	Content
0	port_starboard_beam_angle_deg	no	byte	
1	fore_aft_beam_angle_deg	no	byte	
2	port_starboard_element_angle_deg	no	byte	
3	fore_aft_element_angle_deg	no	byte	
5	?	yes	StructUnknown2	
6	?	yes	StructUnknown3	

StructUnknown1 (variable structure)

Field nr.	Name	Optional?	Type	Content
0	?	?	float	Seen 0
1	?	?	float	Seen 1
2	?	?	float	Seen values 7.999582, 11.99967, 15.99977, 29.70248, 31.79441, 109.5105
3	?	?	float	Seen values 0.7, 0.8936226, 0.8955613

StructUnknown2 (variable structure)

Field nr.	Name	Optional?	Type	Content
0	?	?	float	Seen -55, 55
1	?	?	float	Seen 0

StructUnknown3 (variable structure)

Field nr.	Name	Optional?	Type	Content
0	?	?	byte	Seen 0
1	?	?	float	Seen 52
2	?	?	float	Seen 0.5
3	?	?	float	Seen 52
4	?	?	float	Seen 52
5	?	?	float	Seen 0.5
6	?	?	float	Seen 0.5

Sonar data

Any structure in it?

Record trailer (fixed structure with CRC)

Type		Content
uint	magic_number	= 0xF98EACBC
uint	chunk_size	Total length of record in byte. From beginning of Record header to end of Record trailer. This allows backward navigation from one record to the previous one.

Example:

Byte(s) Meaning

Record header structure

06 VarUInt32 number of fields in record header structure

04 VarUInt32 field 0 "magic_number", length 4
86 da e9 b7 uint
0f VarUInt32 field 1 "state_data_info", length 7 (actual length follows)
07 VarUInt32 length = 7
02 VarUInt32 number of fields
01 VarUInt32 field 0 "state", length 1
02 VarUInt32 = 2
0b VarUInt32 field 1 "data_info", length 3
01 VarUInt32 number of array elements
array element 0 of DataInfo structure
01 VarUInt32 length = 1
01 VarUInt32 "channel_id" = 1
14 VarUInt32 field 2 "sequence_count", length 4
01 00 00 00 uint
1c VarUInt32 field 3 "data_crc", length 4
ba d7 b3 a1 uint
22 VarUInt32 field 4 "data_size", length 2
2d 18 ushort
2c VarUInt32 field 5 "recording_time_ms", length 4
04 03 00 00 uint
e5 1d e3 22 uint CRC over record header structure

Record body structure

0d VarUInt32 number of fields in record body structure
01 VarUInt32 field 0 "channel_id", length 1
01 VarUInt32 = 1
0b VarUInt32 field 1, "bottom_depth", length 3
b0 c8 03 VarInt32 = 58416
13 VarUInt32 field 2, "drawn_bottom_depth", length 3
c8 c1 03 VarInt32 = 57544
19 VarUInt32 field 3, "first_sample_depth", length 1
00 VarInt32 = 0
23 VarUInt32 field 4, "last_sample_depth", length 3
ba 89 05 VarInt32 = 83130
29 VarUInt32 field 5, "gain", length 1
f0 byte = 240
31 VarUInt32 field 6, "sample_status", length 1
02 VarUInt32 = 2
3c VarUInt32 field 7, "sample_cnt", length 4
00 08 00 00 uint = 2048
41 VarUInt32 field 8, "shade_avail", length 1
01 bool = true
4c VarUInt32 field 9, "scposn_lat", length 4

4b 36 68 20 int (map units) = 45.5724540632218°
54 VarUInt32 field 10, "scposn_lon", length 4
b5 4a 82 07 int (map units) = 10.5594643671066°
5c VarUInt32 field 11, "water_temperature", length 4
27 4e be 41 float = 23.78816 °C
61 VarUInt32 field 12, "beam", length 1
01 VarUInt32 = 1

Fields 13 "beam_info" and 14 "interrogation_id" are not present here.

Sonar data

cf 71 cf 71 cf 71 cf 71 cf 71 cf 71 cf 71 cf 71 cf 71 cf 71 cf 71 cf 71 cf 71 cf 71 cf 71 cf 71 cf 71 cf 71 cf 71 cf 71 cf 71 cf 71 cf 71 cf 71 cf 71 cf 71 cf 71 cf 71 78 1c 98 18 42 17 2a 14 c5 16 78 ...

Record trailer structure

bc ac 8e f9 uint magic value
0C 18 00 00 uint length = 6156 total length of record including the header and trailer
d0 48 d4 a6 uint CRC over record trailer structure

Garmin Format of Sonar Index File

For each RSD file, an index file is created in C:\Users\<Name of user>\AppData\Local\Garmin\HomePort\SonarIndex.

The filename is the filename of the RSD file (without the extension) + "_" + the value of the header field "date_time_of_recording", in decimal representation. The file does not have any filename extension.

General file structure

Header
List of entries, one entry for each record in the RSD file

Header

Type		Content	
uint		Seen value 7 Magic?	
uint		Number of channels Seen values 1, 4 and 6	
For each channel:			
	uint	channel_id	
	double	Maximum depth for this channel, in meter.	Probably used for scaling the graphical display.
	double	Minimum depth for this channel, in meter.	
	ushort	Channel number. Seen value 1-4	
byte		Seen value 1	
uint		Maximum length of a record body	
byte		Seen value 1	
3 x byte		Different with almost each run. No reasonable content, does not correlate to date/time either.	

Entry for record

Offset	Type	Content
0x00	uint	channel_id
0x04	uint	Byte offset of the record body in RSD file
0x08	uint	recording_time_ms

References

Used sources of information

[1] Recording Sonar Data with a Garmin Marine Device https://support.garmin.com/en-US/?faq=28FiszKefp6WDJredHQFx9

[2] Google Protocol Buffers https://en.wikipedia.org/wiki/Protocol_Buffers

[3] Unix time https://en.wikipedia.org/wiki/Unix_time

Standards and specifications

[4] ISO/IEC/IEEE 60559:2011 Information technology – Microprocessor Systems – Floating-Point arithmetic (or IEEE Std 754-2019)

[5] ITU-T Recommendation V.42: Error-correcting procedures for DCEs using asynchronous-to-synchronous conversion https://www.itu.int/rec/T-REC-V.42/en

[6] Google Protocol Buffers https://developers.google.com/protocol-buffers/

Sources of sample files

[7] https://share.phys.ethz.ch/~alps/idea/Sonar_Recordings/

[8] https://www.dropbox.com/s/kfdya038azy2cy4/Sonar001.RSD