

Physical protocol

RS232 protocol with 9600 baud, 8 data bits, 1 stop bit, no parity. Binary value 0 is +5V. Binary value 1 is 0V

Note: The RS232 specification says that a voltage between 3V and 25V is logical 0 and a voltage between -25V and -3V is logical 1. A voltage of 0V is therefore just in the middle of the undetermined region! But the serial port circuitry of a standard PC indeed interprets 0V as 1 and the GARMIN is not damaged by the high voltages from the PC!

Garmin protocol

Every message has the structure:

- Start of message (\$10)
- Type of message
- Number of data bytes
- Data bytes
- Check sum
- End of message (\$1003)

Example:

10 22 0D 40 CD 59 23 00 19 53 06 E8 E8 81 1C 01 68 10 03

Table 1:

HEX value	Meaning
10	Start of message
22	Type of message (\$22 - Track point)
0D	Number of data bytes
40	Data byte 1
CD	Data byte 2
59	Data byte 3
23	Data byte 4
00	Data byte 5
19	Data byte 6
53	Data byte 7
06	Data byte 8
E8	Data byte 9
E8	Data byte 10
81	Data byte 11
1C	Data byte 12
01	Data byte 13

Table 1:

HEX value	Meaning
68	Check sum $22+0D+40+CD+59+23+00+19+53+06+E8+E8+81+1C+01+68 = 500$ (The least significant byte is zero)
1003	End of message

“Stuffing” of the \$10 delimiter is applied starting with byte 3 (Number of data bytes) and ending with the checksum. This means that any \$10 appearing here is duplicated. Such an additional \$10 is not included in the “Number of data bytes” value or in the checksum.

Example:

Instead of

Table 2:

HEX value	Meaning
10	Start of message
22	Type of message (\$22 - Track point)
0D	Number of data bytes
10	Data byte 1
CD	Data byte 2
59	Data byte 3

Table 2:

HEX value	Meaning
23	Data byte 4
10	Data byte 5
10	Data byte 6
53	Data byte 7
06	Data byte 8
E8	Data byte 9
E8	Data byte 10
81	Data byte 11
1C	Data byte 12
01	Data byte 13
91	Check sum $22+0D+10+CD+59+23+10+10+53+06+E8+E8+81+1C+01+91 = 500$
1003	End of message

one would have

Table 3:

HEX value	Meaning
10	Start of message

Table 3:

HEX value	Meaning
22	Type of message (\$22 - Track point)
0D	Number of data bytes
10	Data byte 1
10	Data byte 1 repeated
CD	Data byte 2
59	Data byte 3
23	Data byte 4
10	Data byte 5
10	Data byte 5 repeated
10	Data byte 6
10	Data byte 6 repeated
53	Data byte 7
06	Data byte 8
E8	Data byte 9
E8	Data byte 10
81	Data byte 11
1C	Data byte 12
01	Data byte 13

Table 3:

HEX value	Meaning
91	Check sum $22+0D+10+CD+59+23+10+10+53+06+E8+E8+81+1C+01+91 = 500$
1003	End of message

As no message type has 16 data bytes this “stuffing” can in reality only occur for a data byte or for the check sum.

Each message received should be acknowledged by sending a message of type \$06. Each such “acknowledge message” has 2 data bytes, i.e. it has the form:

Table 4:

HEX value	Meaning
10	Start of message
06	Type of message (\$06 - acknowledge)
02	Number of data bytes
??	Data byte 1
00	Data byte 2, always zero
??	Check sum
1003	End of message

The acknowledgement of a message of type 22 (as above) is consequently:

Table 5:

HEX value	Meaning
10	Start of message
06	Type of message (06 - acknowledge)
02	Number of data bytes
22	Data byte 1
00	Data byte 2, always zero
D6	Check sum (06+02+22+00+D6 = 100)
1003	End of message

Some different message types:

Table 6:

Code1 (HEX)	Code2 (HEX)	Number of data bytes (DEC)	Type of data
0A	n.a.	2	Request that the other GARMIN sends data of a certain type
0C	n.a.	2	Confirmation that all data messages have been sent
1B	n.a.	2	Specification of how many messages with data that will be sent
06	n.a.	2	Message properly received, please send next message ("Acknowledge")

Table 6:

Code1 (HEX)	Code2 (HEX)	Number of data bytes (DEC)	Type of data
21	n.a.	2	Message not properly received, please repeat message ("Not acknowledge")
1F	01	43	Almanac data
13	03	62	Proximity point data
1E	04	21	Route data
22	06	13	Track data
23	07	58	Way point data
3D	18		Configuration data

Code1 is the number put in byte 2 of each message. Code2 is used in the first of the two “Data bytes” of messages 0A and 0C.

The “Not acknowledge” message (21) will in practise never be sent!

If the data transfer is initiated by the GARMIN that should send the data the dialogue starts with a message of type 1B. For this message “Number of data bytes” is 02 and the two data bytes specifies the number of messages that will follow with the least significant byte first.

Example:

10 1B 02 4D 00 96

Table 7:

HEX value	Meaning
10	Start of message
1B	Type of message (\$1B - Specification of how many messages that will follow)
02	Number of data bytes
4D	Number of messages that will follow, least significant byte
00	Number of messages that will follow, most significant byte
96	Check sum ($1B + 02 + 4D + 00 + 96 = 100$)
1003	End of message

The standard acknowledge must be given to this message, i.e.

Table 8:

HEX value	Meaning
10	Start of message
06	Type of message (\$06 - acknowledge)
02	Number of data bytes
1B	Data byte 1 should be the type of message to acknowledge
00	Data byte 2, always zero

Table 8:

HEX value	Meaning
DD	Check sum (06+02+1B+00+DD = 100)
1003	End of message

Then follows the specified number of messages, in the example \$4D, which all will be of the same type (Almanac,...,Configuration). The code number for these data types is in byte 2 in each of the messages in agreement with the general GARMIN message standard. Each message must be acknowledged by the receiver with a message of the “06” type, otherwise the sender will (after having waited a certain time) repeat the last message instead of sending a new message.

Finally, when all data has been transmitted an “end of data” message of type \$0C is sent. The number of data bytes for this message is 2. The first of these data bytes contains “Code2” for the type of data that was sent and the second data byte is always zero. Also this last message (like all messages) should be acknowledged with a message of \$06 type. This then terminates the dialogue. If the data transfer instead is initiated by the GARMIN that should receive data a message of type \$0A is sent to specify what type of data that is wanted. The number of data bytes for this message is 2. The first of these data bytes contains “Code2” for the type of data that is desired and the second data byte is always zero.

The request from one GARMIN to get the track from the other GARMIN is consequently

Table 9:

HEX value	Meaning
10	Start of message

Table 9:

HEX value	Meaning
0A	Type of message (\$0A - Request for data)
02	Number of data bytes
06	Track wanted
00	Data byte 2, always zero
EE	Check sum (0A+02+06+00+EE = 100)
1003	End of message

After having sent this message requesting data (that must be acknowledged by a \$06 message like all messages) the other GARMIN is the talker and the dialogue continues as above with the GARMIN that was requesting data listening only sending “acknowledge” (a “06” message) after each data message received and after having received the “End message” (message code 0C).

Data encoding

GARMIN encodes the longitudes in the range -180 deg +180 deg and the latitudes in the range -90 deg +90 deg as 32 bits signed integers such that -180 deg corresponds to \$80000000 (smallest possible number) and +180 deg corresponds to \$7FFFFFFF (largest possible number). Dates are (unsigned) integers giving the number of second since 31 Dec 1989 at 00:00:00.

The 4 bytes of such a 32 bit integer are given in reverse order. The example above:

10 22 0D 40 CD 59 23 00 19 53 06 E8 E8 81 1C 01 68 10 03

is a track point message

Track point message

Table 10:

HEX value	Meaning
10	Start of message
22	Type of message (\$22 - Track point)
0D	Number of data bytes
40	Data byte 1 (latitude 1)
CD	Data byte 2 (latitude 2)
59	Data byte 3 (latitude 3)
23	Data byte 4 (latitude 4)
00	Data byte 5 (longitude 1)
19	Data byte 6 (longitude 2)
53	Data byte 7 (longitude 3)
06	Data byte 8 (longitude 4)
E8	Data byte 9 (epoch 1)
E8	Data byte 10 (epoch 2)
81	Data byte 11 (epoch 3)

Table 10:

HEX value	Meaning
1C	Data byte 12 (epoch 4)
01	Data byte 13 (switch, either 0 or 1)
68	Check sum $22+0D+40+CD+59+23+00+19+53+06+E8+E8+81+1C+01+68 = 500$
1003	End of message

The meaning is:

latitude: $\$2359CD40 = 593087808 \Rightarrow 180/2147483647 * 593087808 \text{ deg} = 49.712045 \text{ deg}$

longitude: $\$06531900 = 106109184 \Rightarrow 180/2147483647 * 106109184 \text{ deg} = 8.893969 \text{ deg}$

epoch: $\$1C81E8E8 = 478275816 \Rightarrow 478275816 - 315619200 \text{ sec} = 162656616 \text{ sec after 1 Jan 2000 00:00:00 UTC}$ which is 2005/02/25 14:23:36

Data byte 13 is either 01 or 00. It is used to indicate the continuity of the track

Way point message

Table 11:

HEX value	Meaning
10	Start of message
23	Type of message (\$23 - Waypoint)
3A	Number of data bytes
42	Data byte 1 (Name symbol 1)
49	Data byte 2 (Name symbol 2)
4C	Data byte 3 (Name symbol 3)
31	Data byte 4 (Name symbol 4)
20	Data byte 5 (Name symbol 5)
20	Data byte 6 (Name symbol 6)
C0	Data byte 7 (latitude 1)
10	Data byte 8 (latitude 2)
5A	Data byte 9 (latitude 3)
23	Data byte 10 (latitude 4)
40	Data byte 11 (longitude 1)
1F	Data byte 12 (longitude 2)
53	Data byte 13 (longitude 3)
06	Data byte 14 (longitude 4)

Table 11:

HEX value	Meaning
00	Data byte 15 (Always zero)
00	Data byte 16 (Always zero)
00	Data byte 17 (Always zero)
00	Data byte 18 (Always zero)
32	Data byte 19 (Byte 1 of comment)
35	Data byte 20 (Byte 2 of comment)
2D	Data byte 21 (Byte 3 of comment)
46	Data byte 22 (Byte 4 of comment)
45	Data byte 23 (Byte 5 of comment)
42	Data byte 24 (Byte 6 of comment)
2D	Data byte 25 (Byte 7 of comment)
30	Data byte 26 (Byte 8 of comment)
35	Data byte 27 (Byte 9 of comment)
20	Data byte 28 (Byte 10 of comment)
31	Data byte 29 (Byte 11 of comment)
34	Data byte 30 (Byte 12 of comment)
3A	Data byte 31 (Byte 13 of comment)
32	Data byte 32 (Byte 14 of comment)
30	Data byte 33 (Byte 15 of comment)

Table 11:

HEX value	Meaning
00	Data byte 34 (Byte 16 of comment)
20	Data byte 35 (Byte 17 of comment)
	etc.....
20	Data byte 58 (Byte 40 of comment)
42	Check sum
1003	End of message

The meaning is:

Name: \$42494C312020 - “BIL1” (ASCII)

Latitude: \$235A10C0 = 593105088 => $180/2147483647 * 593105088 \text{ deg} = 49.713494 \text{ deg}$

Longitude: \$06531F40 = 106109184 => $180/2147483647 * 106109184 \text{ deg} = 8.894103 \text{ deg}$

Comment: \$32352D4645422D30352031343A3230 - “25-FEB-05 14:20” (ASCII)

Route message

Table 12:

HEX value	Meaning
10	Start of message
1D	Type of message (\$1D - Route)

Table 12:

HEX value	Meaning
15	Number of data bytes
00	Data byte 1 (Route number)
57	Data byte 2 (Byte 1 of ASCII comment)
	etc.....
20	Data byte 21 (Byte 20 of ASCII comment)
42	Check sum
1003	End of message

Proximity point message

This message is identical to waypoint message except that a 4 byte floating point value specifying the critical distance is added at the end. The type is \$13.

Table 13:

HEX value	Meaning
10	Start of message
13	Type of message (\$13 - Proximity point)

Table 13:

HEX value	Meaning
3E	Number of data bytes
42	Data byte 1 (Name symbol 1)
45	Data byte 2 (Name symbol 2)
41	Data byte 3 (Name symbol 3)
52	Data byte 4 (Name symbol 4)
20	Data byte 5 (Name symbol 5)
20	Data byte 6 (Name symbol 6)
80	Data byte 7 (latitude 1)
FE	Data byte 8 (latitude 2)
57	Data byte 9 (latitude 3)
23	Data byte 10 (latitude 4)
BF	Data byte 11 (longitude 1)
25	Data byte 12 (longitude 2)
55	Data byte 13 (longitude 3)
06	Data byte 14 (longitude 4)
00	Data byte 15 (Always zero)
00	Data byte 16 (Always zero)
00	Data byte 17 (Always zero)
00	Data byte 18 (Always zero)

Table 13:

HEX value	Meaning
32	Data byte 19 (Byte 1 of comment)
	etc.....
20	Data byte 58 (Byte 40 of comment)
00	Data byte 59 (Distance 1)
00	Data byte 60 (Distance 2)
C8	Data byte 61 (Distance 3)
43	Data byte 62 (Distance 4)
E0	Check sum
1003	End of message

The meaning is:

Name: \$424541522020 => "BEAR" (ASCII)

Latitude: \$2357FE80 = 592969344 => $180/2147483647 * 592969344$ deg = 49.702116 deg

Longitude: \$065525BF = 106243519 => $180/2147483647 * 106243519$ deg = 8.905229 deg

Distance: \$43C80000 => 400 (Distance in meter, IEEE floating point representation)

Software consideration

The most obvious use/need for this interface is to dump the track points to a PC . The simplest and most straightforward way to do this is to initiate this transfer of track (or any other data) from the GARMIN and to program the PC to simply send “acknowledge” at the appropriate times. For any message the GARMIN sends byte 2 is the type and byte 3 is the number of data bytes. From the latter one computes the number of bytes in the message ($6 + \text{“the number of data bytes”}$) and when the full message has been sent an 8 byte acknowledge message is sent to the GARMIN. The type of the message that was recieved (byte 2) shall be inserted as byte 4 of the acknowledge message and the resulting check sum should be put in byte 6. The other 6 bytes of this acknowledge message are fixed (Table 4). When this 8 byte “Acknowledge” message has been received by the GARMIN it continues to send the next message