Inertial Sensor Interface

IMU, MRU, AHRS & INS

Firmware Reference Manual

Document SBGFWM.2.2
Revision 2.2 - Nov 20, 2020

Support
EMEA
support@sbg-systems.com
+33 1 80 88 43 70

Americas
support@sbg-systems.com
+1 (657) 549-5807

SBG SYSTEMS

Revision history

Rev	Date	Author	Information
2.2	Oct 29, 2020	RS	[SBGFWM-126] Now ELLIPSE-N and D can accept RTCM corrections over Port A [SBGFWM-106] Added empty version of each NMEA message
2.1	Jun 17, 2020	RS	[SBGFWM-117] Fix missing output example for NMEA HDT message [SBGFWM-116] Fix ZDA fields numbering and added empty output example [SBGFWM-120] Add Cobham AT_ITINS output log support [SBGFWM-119] Added in SBG_ECOM_FEATURES gnss firmware version field [SBGFWM-118] Removed legacy IG-500 protocol support [SBGFWM-115] Simplified SbgEComGnssModelsStdIds enum for ELLIPSE-N and ELLIPSE-D
2.0	Apr 22, 2020	RS	[SBGFWM-112] Added status field for SBG_ECAN_MSG_AUTO_TRACK_SLIP_CURV (0x220) message [SBGFWM-111] Improve organization for High Performance INS [SBGFWM-110] Add slip angle / curvature support for high performance INS [SBGFWM-108] Add specific SBG_ECOM_THIRD_PARTY_ADA_01 output log [SBGFWM-107] Fix AHRS-500 euler rate to body rotation rates [SBGFWM-104] Review NMEA Quality indicator translation and DOP computation [SBGFWM-102] Added in SBG_ECOM_LOG_GPS#_POS status report for all GNSS constellations [SBGFWM-101] Updated GGA/RMC number of decimal digits for latitude / longitude to 8 [SBGFWM-99] Added SBG_ECOM_LOG_NMEA_1_GGK message output
1.9	Aug 19, 2019	RS	[SBGFWM-96] Add compatibility support Crossbow AHRS500 series [SBGFWM-95] SBG_ECOM_CMD_GNSS_1_LEVER_ARM_ALIGNMENT deprecated [SBGFWM-94] Added SBG_ECOM_CMD_GNSS_1_INSTALLATION command to set/get GNSS lever arm [SBGFWM-93] Add INDYN NMEA like message output for marine applications [SBGFWM-92] Add SBG_ECOM_OUTPUT_MODE_DIV_5 to output data at 40 Hz [SBGFWM-90]Now SBG_ECAN_MSG_ODO_VELOCITY doesn't use float anymore [SBGFWM-89] Now SBG_ECAN_MSG_UTC_1 uses microsecond instead of tenth of millisecond [SBGFWM-88] Updated CAN log default IDs for Ship Motion HP from 0x210 to 0x14A [SBGFWM-87] Add CAN output message with track, slip and curvature indications [SBGFWM-86] Add CAN output message with vehicle body velocity [SBGFWM-85] Fix PASHR missing IMU/INS status char field [SBGFWM-84] Fix typo issue on PSBGI message for gyroscope component Z [SBGFWM-83] Add configuration commands for CAN odometer support [SBGFWM-82] Add baseline length field in SBG_ECOM_LOG_GPS#_HDT log [SBGFWM-81] Added SBG_ECOM_SOL_DEPTH_USED in EKF solution status [SBGFWM-80] Renamed SBG_ECOM_SOL_PRESSURE_USED to SBG_ECOM_SOL_AIR_DATA_USED [SBGFWM-79] Renamed SBG_ECOM_AIDING_PRESSURE_RECV to SBG_ECOM_AIDING_AIR_DATA_RECV [SBGFWM-78] Converted SBG_ECOM_AIDING_EM_LOG_RECV to SBG_ECOM_AIDING_DEPTH_RECV [SBGFWM-77] Updated SBG_ECAN_MSG_PRESSURE to SBG_ECAN_MSG_AIR_DATA and added airspeed [SBGFWM-76] Updated SBG_ECOM_LOG_PRESSURE to SBG_ECOM_LOG_AIR_DATA and added airspeed [SBGFWM-75] Updated SBG_ECOM_CMD_AIDING_ASSIGNMENT command for AirData support [SBGFWM-74] Add SBG_ECOM_LOG_DEPTH and SBG_ECAN_MSG_DEPTH_INFO/ALTITUDE output logs [SBGFWM-73] Add listing for AsteRx4 and AsteRx-m2a GNSS receivers [SBGFWM-71] Add sbgECom Log Event Output useful for virtual odometer [SBGFWM-70] Renamed DVL standard deviation fields to quality indicators [SBGFWM-68] Updated SBG_ECOM_CMD_AIDING_ASSIGNMENT command for DVL support [SBGFWM-65] Implement AirData aiding configuration commands [SBGFWM-64] Implement DVL aiding configuration commands

Rev	Date	Author	Information
1.8	Nov 8, 2018	RS	[SBGFWM-63] Updated GNSS models for command SBG_ECOM_CMD_GNSS_MODEL_ID [SBGFWM-62] Added new Swell Mode Ship Motion status flags [SBGFWM-61] Added a new short IMU log message also used for post processing
1.7	Jun 1, 2018	AG	[SBGFWM-60] Fixed invalid Ethernet and CAN status flag definitions [SBGFWM-59] Added missing availability of log PASHR for EKINOX and APOGEE [SBGFWM-58] Removed magnetometer support for Ekinox [SBGFWM-56] Added DOLOG HRP message documentation [SBGFWM-55] Updated serial parity information as only EKINOX and APOGEE have this configuration
1.6	Feb 6, 2018	AG	[SBGFWM-54] Fixed NMEA GST message availability for ELLIPSE A [SBGFWM-53] Fixed invalid SBG_ECOM_LOG_EVENT_# availability matrix for ELLIPSE [SBGFWM-52] Added support for ELLIPSE Micro product line [SBGFWM-51] Added PORT E option for aiding module assignments [SBGFWM-50] Fixed TSS1 status flags for heading aided but unstable [SBGFWM-49] Fix PASHR heave sign definition [SBGFWM-48] Updated ship motion description for ELLIPSE as this product only output heave [SBGFWM-47] Fix reverted parameters in SBG_ECOM_CMD_COMPUTE_MAG_CALIB [SBGFWM-45] Add SBG_ECOM_CMD_VALIDITY_THRESHOLDS command definition [SBGFWM-44] Fixed GNSS HDT & RAW Binary Output Logs Availability description for Ellipse-N/D
1.5	Mar 6, 2017	AG	[SBGFWM-13] Added Kongsberg Binary 26 message definition [SBGFWM-40] Remove obsolete virtual odometer data [SBGFWM-42] Add an uptime indication in SBG_ECOM_LOG_STATUS [SBGFWM-38] Added SBG_ECOM_CMD_ETHERNET_INFO to retrieve current device IP address [SBGFWM-37] Added SBG_ECOM_CMD_ETHERNET_CONF to set / get the Ethernet configuration [SBGFWM-28] Add GST requirement in NMEA section [SBGFWM-33] Added SBG Proprietary NMEA message with acceleration and angular rate [SBGFWM-29] Add support for NMEA VTG output log on Ellipse series [SBGFWM-31] Fixed Pressure and Altimeter Description in Firmware Manual [SBGFWM-30] Added UAV motion profile definition (for low dynamic UAV applications) [SBGFWM-27] Added proprietary NMEA message PASHR for roll, pitch, heading, heave [SBGFWM-32] Added NMEA PSBGB prorpietary message [SBGFWM-34] Add new output log class for NMEA proprietary messages [SBGFWM-36] Removed deprecated "course" from the GNSS configurable aiding sources [SBGFWM-43] Added GPS number of SV used and diff corrections details in sbgECan protocol
1.4	Jun 7, 2016	RS	[SBGFWM-1] Merged Ellipse and Ekinox series Firmware Manuals [SBGFWM-2] Added IXBlue PHINF, PHTRO, PHLIN, PHOCT output log description [SBGFWM-3] Added SBG_ECOM_SOL_ALIGN_VALID flag for EKF logs [SBGFWM-4] Added SBG_ECOM_GENERAL_CPU_OK status flag [SBGFWM-5] Removed references to ship motion 1 to 3 due to new deported heave concepts [SBGFWM-6] Now, CAN outputs share all enum and constants definitions with sbgECom outputs [SBGFWM-8] Added ZUPT Used in Solution status [SBGFWM-9] Added KVH third party output format (SBG_ECOM_THIRD_PARTY_KVH) [SBGFWM -10] Added SBG_ECOM_LOG_FAST_IMU_DATA, 1KHz IMU message for low latency applications [SBGFWM-11] Added SBG_ECOM_CMD_OUTPUT_CLASS_ENABLE message documentation [SBGFWM-12] Added SBG Proprietary NMEA message with acceleration and angular rate [SBGFWM-15] Added SBF (Septentrio) protocol support on Ellipse-E [SBGFWM-18] Fixed invalid port ID definition for the SBG_ECOM_CMD_OUTPUT_CONF [SBGFWM-20] Fixed ID SBG_ECOM_LOG_PRESSURE from 31 to 36 in documentation [SBGFWM-22] Added direct PPS from internal GNSS Sync Output (SBG_ECOM_CMD_SYNC_OUT_CONF) [SBGFWM-23] Added Pressure Used in Solution status [SBGFWM-24] Fixed Accelerometer and Magnetometer message description [SBGFWM-25] Update support contact details

Rev	Date	Author	Information
1.3	May 4, 2015	NM	[EKIDOC-29] Fixed eCom logs name to be consistent with actual definitions [SBGFWM-19] Fixed CAN Pressure message missing [SBGECOM-19] Added SBG_ECOM_LOG_PRESSURE log for depth sensors and altimeters [EKIDOC-23] Fixed SBG_ECOM_ETH#_TX/RX_OK status definitions [EKIDOC-20] Add details about undulation for GPS data [SBGFWM-16] Added diff correction age, diff base id and num sv to the SBG_ECOM_LOG_GPS#_POS [EKIDOC-35] Merged protocol modifications from Ellipse to Ekinox
1.2	Mar 17, 2014	AG	Renamed GPS Omnistar solution type Removed SBG_ECOM_POS_OMNISTAR_G2 position solution type Added SBG_ECOM_POS_FIXED position type Updated heave output messages specifications Added new NMEA messages definition (GST, VBW) Added settings import/export commands
1.1	Nov 15, 2013	RS	Added pitch field in GPS true heading data Added DVL support output logs Updated status definitions for DVL Added Teledyne PD0 and PRDID output logs Added CAN messages specifications
1	May 6, 2013	NM	First version of this document

Index

SBG SYSTEMS

SBG SYSTEMS

1. Introduction

This firmware reference manual presents the input and output protocols supported by ELLIPSE, EKINOX, APOGEE, NAVSIGHT and QUANTA products families.

In this documentation, EKINOX, APOGEE, NAVSIGHT and QUANTA products are referenced as High Performance INS products. These products all share the same firmware and as such offer the same features, output logs and configurations. ELLIPSE series, on the other hand, have a specific firmware.

With this manual, you will learn how to read data from the device and how to send some configuration commands or aiding data to the device.

1.1. Types definitions

1.1.1. Scalar types

When required, the following types will be used to describe variables format.

Type	Description
Mask	This type defines an unsigned integer variable used to store a set of bit-masks; This type has no pre-defined size and user should refer to each occurrence for corresponding size.
Enum	This type defines a group of several bits defining a list of possible states. Each value corresponds to a state; This type has no pre-defined size and user should refer to each occurrence for corresponding size.
bool	8 bits boolean, 0x00 is FALSE, 0x01 is TRUE
uint8	8 bits unsigned integer
int8	8 bits signed integer
uint16	16 bits unsigned integer
int16	16 bits signed integer
uint32	32 bits unsigned integer
int32	32 bits signed integer
uint64	64 bits unsigned integer
int64	64 bits signed integer
float	32 bits single floating point, standard IEEE 754 format
double	64 bits double floating point, standard IEEE 754 format
void[]	Data buffer, with variable length
string	Standard, null terminated ASCII string. String max size is defined in the message.

1.2. Complex types

1.2.1. Vectors objects

Vectors are stored in a 1D array of float or double components.

$$V=\begin{pmatrix}V_0\\V_1\\V_2\end{pmatrix}$$

This vector is stored in memory this way:

V_0	V_1	V_2

1.2.2. Matrix objects

Matrix are stored in a 1D array of float or double items. They are expressed in vector column format.

$$M_{3x3}=\begin{pmatrix}U_0 & V_0 & W_0\\U_1 & V_1 & W_1\\U_2 & V_2 & W_2\end{pmatrix}$$

This matrix is stored this way:

U_0	U_1	U_2	V_0	V_1	V_2	W_0	W_1	W_2

1.2.3. Interpreting revision numbers

Some commands and messages contain revision numbers which are all coded as 32 bits unsigned integers. A general rule of thumb is that a higher value in the version number always implies a more recent revision.

Two types of revisions are used with a different interpretation:

1.2.3.1. Basic revision type

This type of version implements a very common scheme, Major.Minor.Rev.Build. It's used everywhere when a version is needed except for software and firmware. For example the basic revision type is used to identity the version of the calibration data or hardware revision.

The basic revision type is formatted as follows:

Field	Size (bits)	Definition
SCHEME	1 (MSB)	This field is common to basic and software revision type Set to 0 indicates a basic revision type
MAJOR	7	Major version number Range 0 to 127
MINOR	8	Minor version number Range 0 to 255
REV	8	Revision Range 0 to 255
BUILD	8	Build number Range 0 to 255

1.2.3.2. Software revision type

The software revision scheme is used to identify the version of all firmware and software items.

It is formatted as follows:

Field	Size (bits)	Definition
SCHEME	1 (MSB)	This field is common to basic and software revision type Set to 1 indicates a software revision type
STATUS	3	Status of the current software release
MAJOR	6	Major version number Range 0 to 63
MINOR	6	Minor version number Range 0 to 63
BUILD	16	Build number Range 0 to 65535

The status field provides useful information about the stability of this firmware version:

SbgVersionStatus enum definition

Status code	Value	Description
SBG_VERSION_STATUS_DEV	0x0	This version is under development – Internal use only
SBG_VERSION_STATUS_ALPHA	0x1	This version is under development – Internal use only
SBG_VERSION_STATUS_BETA	0x2	This version is a beta and should be used with caution
SBG_VERSION_STATUS_RC	0x3	This version is a release candidate
SBG_VERSION_STATUS_STABLE	0x4	This version is an official release
SBG_VERSION_STATUS_HOT_FIX	0x5	This version is a stable release and contains a bug fixes

1.2.3.3. Handling compatibility between basic and software revisions

It is easy to maintain compatibility between the basic revision and the software revision mode by just checking MSB value: if we read 0, we are facing a basic revision value, while if we read 1, we have a software revision type.

1.2.4. IP Address representation

The sbgECom protocol defines the type sbgIpAddress to handle an Ipv4 address. The underlying type is an uint32 that stores the four 8 bits IP address fields in big endian to comply with Ethernet endianness.

As a result, the ip address 192.168.1.2 (A.B.C.D) has the following memory organization:

	Little Endian Platform				Big Endian Platform			
Byte Index	(LSB) 0	1	2	(MSB) 3	(MSB) 0	1	2	(LSB) 3
Decimal value	192	168	1	2	192	168	1	2
Hex Value	0xC0	0xA8	0x01	0x02	0xC0	0xA8	0x01	0x02
Value as uint32	33 663 168				3 232 235 778			

1.3. Endianness

The sbgECom binary protocol uses little endian data format. Some binary third party messages can use big endian data format. In this case, it's clearly specified in this documentation.

The sbgECom communication library has been designed to be compatible with both little and big endian platforms.

For ASCII or NMEA messages, the platform endianness does not affect messages parsing or generation.

1.4. Conventions and units

SBG Systems AHRS and INS use the International System of Units (SI) when applicable. The device coordinate frame is defined as North East Down (NED).

Physical quantity	Unit description
Angle	Radians, roll, pitch, yaw.
Rotational speed	$rad.s^{-1}$
Acceleration	$m.s^{-2}$
Velocity	$m.s^{-1}$
Latitude	Degrees, positive North, negative South.
Longitude	Degrees, positive East, negative West.
Altitude	Meters, positive up, above Mean Sea Level, negative down.
Ship Motion	Surge positive forward, Sway positive right, Heave positive down expressed in meters.

1.5. Serial, Ethernet and Data-logger interfaces

Depending on the product you have, the device can feature different interfaces and configuration. All devices have at least serial RS-232 and/or RS-422 interfaces. More advanced products also provide Ethernet connectivity as well as an embedded datalogger.

You should refer to the Hardware Reference Manual of your product to get more details about available interfaces and they specificities.

1.5.1. Serial interfaces

All serial ports use the following transmission format for communications:

- Available baudrates: 4800, 9600, 19 200, 38 400, 57 600, 115 200, 230 400, 460 800, 921 600
- 8 bits data
- 1 stop bit
- Parity: None, Even, Odd, Mark, Space
- No control flow

Note: The parity configuration is only available for High Performance INS. For ELLIPSE, parity is always set to None.

1.5.2. Ethernet interfaces

Network enabled products feature up to five Ethernet virtual serial ports: ETH 0, ETH 1, ETH 2, ETH 3, ETH 4. Each port provides an input and output channel for full duplex operations.

Ethernet ports can be configured as virtual serial ports over TCP/IP (server or client) or UDP. The UDP option offers the best data throughput but doesn't guarantee that data are being sent or received correctly.

Note: All data that are transmitted over Ethernet are protected by a 32 bits CRC even for an UDP connection.

1.5.3. Data-logger interface

This interface is considered as a standard serial port and can be used to store any output data log. The data-logger acts exactly as if you have stored all the raw bytes coming from a serial port. The internal data-logger can thus log both binary logs and NMEA/ASCII messages.

1.6. CAN interface

The protocol described in this documentation is used to communicate with the AHRS / INS on a Controller-area network (CAN) bus.

The CAN bus is a message based protocol designed in a first time for automotive applications and used today in almost all industries.

The CAN implementation supports both CAN 2.0A and CAN 2.0B standards in a very versatile manner. It has been designed to maximize the compatibility with third party equipment and CAN software.

1.6.1. Specifications

The following bitrates are supported:

- 1 000 kBit/s
- 500 kBit/s
- 250 kBit/s
- 200 kBit/s
- 125 kBit/s
- 100 kBit/s
- 50 kBit/s
- 25 kBit/s
- 20 kBit/s

A maximum of 8 bytes per frame are transmitted, both standard (11 bits) and extended (29 bits) identifiers are supported.

2. sbgECom Binary Protocol

2.1. General description

The sbgECom protocol has been designed for compact and secured communications thanks to its binary form and 16 bits CRC. It's therefore very efficient for inertial navigation related communications that requires high throughput and high data integrity.

In addition, the sbgECom binary protocol is the best way to access the device full features and accuracy. It's strongly recommended to use this protocol to ensure the best integration into a host system.

2.1.1. Frame definition

All frames sent through the sbgECom protocol have a common format, which is described below:

Field	SYNC 1	SYNC 2	MSG	CLASS	LEN	DATA	CRC	ETX
Size (bytes)	1	1	1	1	2	0 to 4086	2	1
Description	Sync. word	Sync. word	Message ID	Message class	Length of DATA section	Payload data	16 bit CRC	End of frame
Value	0xFF	0x5A	-	-	-	-	-	0x33

Note 1: The LEN field contains the DATA section size in bytes. A 0 LEN field implies that no DATA section is present. Maximum length value is 4086.

Note 2: The whole protocol is defined in LITTLE endian, so LEN and CRC fields are written directly in little endian.

Note 3: CRC field is computed on [MSG, CLASS, LEN, DATA] fields. Check the following section CRC definition for more details about CRC computation.

Note 4: Some third party frames are available on output and will not comply with this protocol format. A specific format will then be defined for each frame. It belongs to the user to decode the different formats if several protocols are used at the same time.

2.1.1.1. CRC definition

The sbgECom protocol uses a 16 bit CRC to detect corrupted messages. This CRC uses the following polynomial value: 0x8408

You can find in the sbgECom library source code, the C code used to compute this CRC in the file misc/sbgCrc.c. The sbgECom CRC implementation uses a lookup table to speed up the CRC computation.

In the C code below, you have a non optimized method to compute the 16 CRC.

```
/*!
 * Compute a CRC for a specified buffer.
 *
 * \param[in]      pBuffer       Read only buffer to compute the CRC on.
 * \param[in]      bufferSize    Buffer size in bytes.
 * \return                        The computed 16 bit CRC.
 */
uint16_t calcCRC(const void *pBuffer, uint16_t bufferSize)
{
      const uint8_t *pBytesArray = (const uint8_t*)pBuffer;
      uint16_t       poly = 0x8408;
      uint16_t       crc = 0;
      uint8_t        carry;
      uint8_t        i;
      uint16_t       j;

      assert(pBuffer);

      for (j = 0; j < bufferSize; j++)
      {
             crc = crc ^ pBytesArray[j];

             for (i = 0; i < 8; i++)
             {
                    carry = crc & 1;
                    crc = crc / 2;
                    if (carry)
                    {
                           crc = crc^poly;
                    }
             }
      }

      return crc;
}
```

2.1.2. Messages classes

The following table lists the messages classes used in the sbgECom protocol.

CLASS	ID	Description
SBG_ECOM_CLASS_CMD_0	0x10	Class that contains sbgECom protocol commands
SBG_ECOM_CLASS_LOG_ECOM_0	0x00	Class that contains sbgECom output log messages.
SBG_ECOM_CLASS_LOG_ECOM_1	0x01	Class that contains sbgECom messages that handle high frequency output
SBG_ECOM_CLASS_LOG_NMEA_0	0x02	Class that contains standard NMEA output logs; typically ones that contain a talker ID. **Note:** This class is only used for identification purpose and does not contain any sbgECom message.
SBG_ECOM_CLASS_LOG_NMEA_1	0x03	Class that contains proprietary NMEA output logs; typically ones that start with a $P or unofficial logs with NMEA style formatting **Note:** This class is only used for identification purpose and does not contain any sbgECom message.
SBG_ECOM_CLASS_LOG_THIRD_PARTY_0	0x04	Class that contains other third party output logs. **Note:** This class is only used for identification purpose and does not contain any sbgECom message.

Table 1: SbgEComClass enum

Commands vs logs

All messages that are part of CMD classes are used for information or configuration purposes. These messages are not intended to change in real time and mainly reflect the device configuration or device name.

On the other hand, messages part of a LOG class contain changing data, such as sensor data, orientation, position and so on.

2.1.3. Synchronous / Asynchronous outputs

The unit handle and send two types of logs. Some logs are synchronous and are thus output on a regular time basis. You can then configure an output rate using a divider and the log will be sent by the device at the desired output. It's the case, for example, for inertial data, orientation output or even navigation logs.

Other logs such as raw GNSS, Odometer, DVL are considered as asynchronous. These logs are generated when a new information is received such as a new GNSS position or a new odometer pulse. These logs are sent on a 'new data' event and the log is time stamped to the age of the data it stores.

2.2. SBG_ECOM_CLASS_CMD_0

2.2.1. Introduction to sbgECom commands

This mode is used mainly for configuration operations. A setting command is sent to the device which provides, once the operation is performed, an appropriate answer. This type of communication is therefore a bi-directional communication.

The basic principle of operation behind the sbgECom commands is driven by the frame payload, as explained in the following section.

2.2.1.1. Reading configuration

When the user asks the sensor for a specific configuration, the configuration frame will be sent without any payload (or with a limited payload). For example, the frame SBG_ECOM_CMD_ODO_CONF, sent without any parameter, will request odometer's current configuration.

As an answer, the device will return the same frame SBG_ECOM_CMD_ODO_CONF, but with full payload:

Field	Description	Unit	Format	Size	Offset
GAIN	Odometer's gain	Pulses/m	float	4	0
GAIN_ERROR	User gain average error	%	uint8	1	4
DIRECTION	Odometer's direction: 0: positive 1: negative	-	bool	1	5
			Total size		6

2.2.1.2. Setting a new configuration

If the user want's to apply a new configuration, then he can send the configuration message with full payload.

Device's answer will be an acknowledge frame, providing information about potential errors during configuration.

Commands effectivity

There are three types of commands, and these types influence the way settings are applied:

- Commands that affect all settings at a time such as importing a complete configuration, or restoring default settings will take immediate effect and will force a device reboot as soon as applied.
- Commands that deeply affect sensor configuration will not be applied instantaneously and will remain cached in RAM memory until a SBG_ECOM_CMD_SETTINGS_ACTION (01) is issued, with SBG_ECOM_SAVE_SETTINGS parameter. The device will then save settings and reboot with new configuration.
- Commands that only slightly affect configuration will be applied immediately. These commands are just stored in RAM memory. If this configuration needs to be saved in non-volatile memory, then a SBG_ECOM_CMD_SETTINGS_ACTION (01) command with SBG_ECOM_SAVE_SETTINGS parameter must be used.

We specify in this document for all commands if new configuration is applied immediately or after reboot time.

2.2.1.3. Acknowledge

Name in sbgECom convention (msg ID): **`SBG_ECOM_CMD_ACK (00)`**

Most configuration commands will get an Acknowledge frame as answer. The ACK frame contains two fields, the first one is the command ID that is being acknowledged. The second one is the returned command status used to know if the command has been executed successfully or if an error has occurred.

Payload description

Field	Description	Unit	Format	Size	Offset
cmdId	Acknowledged CMD ID	-	uint8	1	0
classId	Acknowledged CLASS ID	-	uint8	1	1
errorCode	Returned error code if any - Table 2	enum	uint16	2	2
			Total size		4

Enum definitions

Error Code	Value	Description
SBG_NO_ERROR	0x00	The command has been properly executed
SBG_ERROR	0x01	Command could not be executed properly due to a generic error
SBG_NULL_POINTER	0x02	A pointer equaled NULL
SBG_INVALID_CRC	0x03	A frame with an invalid CRC has been received
SBG_INVALID_FRAME	0x04	The frame sent has an invalid format
SBG_TIME_OUT	0x05	A time out occurred before getting the answer
SBG_WRITE_ERROR	0x06	The device could not write some data
SBG_READ_ERROR	0x07	The device could not read some data
SBG_BUFFER_OVERFLOW	0x08	The buffer is too small to contain the whole frame
SBG_INVALID_PARAMETER	0x09	A parameter has a non valid value
SBG_NOT_READY	0x0A	The device is not ready for communication
SBG_MALLOC_FAILED	0x0B	Could not allocate memory
SBG_INCOMPATIBLE_HARDWARE	0x13	The command cannot be executed because of hardware incompatibility
SBG_INVALID_VERSION	0x14	The command cannot be executed because of version incompatibility

Table 2: SbgErrorCode enum

2.2.2. Large buffers transmission (Transfer sub-protocol)

Some commands from the sbgECom protocol involve data transmission. A transfer sub-protocol is in place to assure a reliable way to send large amount of data from and to the device.

This sub-protocol encapsulated into sbgECom protocol frames works in a master/slave communication scheme where the host acts as the master and the device the slave. The transfers are divided in 3 steps (initialization, data transmission, finalization) which is described below in case of emission and reception.

The following transfer sub-commands are available:

Transfer Command	Command ID	Description
SBG_ECOM_TRANSFER_START	0x0000	Initialization of the transfer
SBG_ECOM_TRANSFER_DATA	0x0001	Data transmission
SBG_ECOM_TRANSFER_END	0x0002	Finalization of the transfer

2.2.2.1. Sending data to the device

Initialization

To initiate a transfer, the host issue an SBG_ECOM_TRANSFER_START, followed by the total size of the transfer. The device will respond with an ACK if it was able to prepare for the reception, a NACK otherwise.

Field	Description	Unit	Format	Size	Offset
transferCmd	Transfer command: SBG_ECOM_TRANSFER_START (0x0000)	enum	uint16	2	0
transferSize	Total size of the data to be transferred	-	uint32	4	4
			Total size		6

Data transmission

Once the transfer is successfully initialized, the host sends buffers sequentially beginning with an SBG_ECOM_TRANSFER_DATA sub command, the offset from the start of the transfer and the byte stream. It must wait for the device ACK before sending the next one. If the device responds with a NACK or does not respond, the host must try sending the buffer again.

Field	Description	Unit	Format	Size	Offset
transferCmd	Transfer command: SBG_ECOM_TRANSFER_DATA (0x0001)	enum	uint16	2	0
offset	Offset from start where to copy new buffer	-	uint32	4	2
payload	Data buffer to transfer	-	void[ ]	n	6
			Total size		6 + n

Finalization

To end the transfer, after all the data has been sent, the host issues an SBG_ECOM_TRANSFER_END. The device will process the whole transfer and respond with an ACK or NACK whether it has validated the received data or not.

Field	Description	Unit	Format	Size	Offset
transferCmd	Transfer command: SBG_ECOM_TRANSFER_END (0x0002)	enum	unit16	2	0
			Total size		2

2.2.2.2. Receiving data from the device

Initialization

To initiate a transfer, the host issue an SBG_ECOM_TRANSFER_START. The device will prepare the data to send and respond with a transfer SBG_ECOM_TRANSFER_START followed by the total size of the transfer. If an error occurs, the device will issue an NACK.

Host request format:

Field	Description	Unit	Format	Size	Offset
transferCmd	Transfer command: SBG_ECOM_TRANSFER_START (0x0000)	enum	uint16	2	0
			Total size		2

Device response format:

Field	Description	Unit	Format	Size	Offset
transferCmd	Transfer command: SBG_ECOM_TRANSFER_START (0x0000)	enum	uint16	2	0
transferSize	Total size of the data to be transferred	-	uint32	4	4
			Total size		6

Data transmission

Once the transfer is successfully initialized, the host requests buffers sequentially beginning with an ECOM_TRANSFER_DATA, the offset from the start of the transfer and the amount of data it requests. It must validate every buffer it receives before requesting the next one. If the device responds with a NACK or does not respond, the host must try requesting the buffer again.

Host request format:

Field	Description	Unit	Format	Size	Offset
transferCmd	Transfer command: SBG_ECOM_TRANSFER_DATA (0x0001)	enum	uint16	2	0
offset	Offset from start where to copy new buffer	-	uint32	4	2
bufferSize	Data buffer size requested for next transfer	-	uint32	4	6
			Total size		10

Device response format:

Field	Description	Unit	Format	Size	Offset
transferCmd	Transfer command: SBG_ECOM_TRANSFER_DATA (0x0001)	enum	uint16	2	0
offset	Offset from start where to copy new buffer	-	uint32	4	2
payload	Data buffer to transfer	-	void[ ]	n	6
			Total size		6 + n

Finalization

To end the transfer, after all the data has been received, the host issues an ECOM_TRANSFER_END. The device will return in a non transfer state and respond with an ACK or an NACK if an error occurred.

Field	Description	Unit	Format	Size	Offset
transferCmd	Transfer command: SBG_ECOM_TRANSFER_END (0x0002)	enum	uint16	2	0
			Total size		2

2.2.3. Commands Overview

Inertial Navigation devices are a very advanced piece of technology with a lot of configuration options. All these options can be redefined using the powerful sbgECom low level protocol.

The following commands are available:

Name (command ID)	Description
SBG_ECOM_CMD_ACK (00)	Acknowledge
SBG_ECOM_CMD_SETTINGS_ACTION (01)	Performs various settings actions
SBG_ECOM_CMD_IMPORT_SETTINGS (02)	Imports a new settings structure to the sensor
SBG_ECOM_CMD_EXPORT_SETTINGS (03)	Export the whole configuration from the sensor
SBG_ECOM_CMD_INFO (04)	Get basic device information
SBG_ECOM_CMD_INIT_PARAMETERS (05)	Initial configuration
SBG_ECOM_CMD_MOTION_PROFILE_ID (07)	Get/Set motion profile ID
SBG_ECOM_CMD_IMU_ALIGNMENT_LEVER_ARM (08)	Sensor alignment on vehicle configuration
SBG_ECOM_CMD_AIDING_ASSIGNMENT (09)	Aiding assignments such as RTCM / GPS / Odometer configuration
SBG_ECOM_CMD_MAGNETOMETER_MODEL_ID (11)	Set/Get magnetometer error model ID
SBG_ECOM_CMD_MAGNETOMETER_REJECT_MODE (12)	Magnetometer aiding rejection mode
SBG_ECOM_CMD_SET_MAG_CALIB (13)	Set magnetic soft and hard Iron calibration data
SBG_ECOM_CMD_START_MAG_CALIB (14)	Start / reset internal magnetic field logging for calibration
SBG_ECOM_CMD_COMPUTE_MAG_CALIB (15)	Compute a magnetic calibration based on previously logged data
SBG_ECOM_CMD_GNSS_MODEL_ID (17)	Set/Get GNSS model ID
SBG_ECOM_CMD_GNSS_1_LEVER_ARM_ALIGNMENT (18)	**DEPRECATED:** GNSS installation configuration, lever arm, alignment
SBG_ECOM_CMD_GNSS_1_INSTALLATION (46)	GNSS mechanical installation parameters (lever arm)
SBG_ECOM_CMD_GNSS_1_REJECT_MODES (19)	GNSS aiding rejection modes configuration.
SBG_ECOM_CMD_ODO_CONF (20)	Odometer gain, direction configuration
SBG_ECOM_CMD_ODO_LEVER_ARM (21)	Odometer installation configuration (lever arm)
SBG_ECOM_CMD_ODO_REJECT_MODE (22)	Odometer aiding rejection mode configuration.
SBG_ECOM_CMD_ODO_CAN_CONF (45)	CAN odometer configuration to decode speed / reverse detection
SBG_ECOM_CMD_UART_CONF (23)	UART interfaces configuration
SBG_ECOM_CMD_CAN_BUS_CONF (24)	CAN bus interface configuration
SBG_ECOM_CMD_CAN_OUTPUT_CONF (25)	CAN identifiers configuration
SBG_ECOM_CMD_SYNC_IN_CONF (26)	Synchronization inputs configuration
SBG_ECOM_CMD_SYNC_OUT_CONF (27)	Synchronization outputs configuration
SBG_ECOM_CMD_NMEA_TALKER_ID (29)	NMEA talker ID configuration
SBG_ECOM_CMD_OUTPUT_CONF (30)	Output configuration
SBG_ECOM_CMD_ADVANCED_CONF (32)	Advanced settings configuration
SBG_ECOM_CMD_FEATURES (33)	Retrieve sensor and GNSS activated features
SBG_ECOM_CMD_LICENSE_APPLY (34)	Apply a new license using a license file / buffer.
SBG_ECOM_CMD_OUTPUT_CLASS_ENABLE (35)	Enable/disable the output of an entire class
SBG_ECOM_CMD_ETHERNET_CONF (36)	Set/get Ethernet configuration such as DHCP mode and IP address

Name (command ID)	Description
SBG_ECOM_CMD_ETHERNET_INFO (37)	Return the current IP used by the device.
SBG_ECOM_CMD_VALIDITY_THRESHOLDS (38)	Set/get the navigation validity thresholds
SBG_ECOM_CMD_DVL_MODEL_ID (39)	Set/get the DVL model id to use
SBG_ECOM_CMD_DVL_INSTALLATION (40)	Set/get the DVL installation configuration such as lever arm
SBG_ECOM_CMD_DVL_REJECT_MODES (41)	Define / retrieve the DVL rejection modes configuration.
SBG_ECOM_CMD_AIRDATA_MODEL_ID (42)	Set/get the AirData model id and input protocol to use
SBG_ECOM_CMD_AIRDATA_LEVER_ARM (43)	Set/get the AirData installation configuration such as lever arm
SBG_ECOM_CMD_AIRDATA_REJECT_MODES (44)	Define / retrieve the AirData rejection modes configuration.

2.2.4. Commands Availability

The table below details the sbgECom commands availability according to a specific product family and product type. For example, if you have an ELLIPSE-E-G4A2-B1, you should refer to the column 'E' in the main column 'ELLIPSE'. Each available command is indicated with the • symbol.

	ELLIPSE					High Performance INS					
Name (Message ID)	I	A	E	N	D	A	E	N	D	M	U
SBG_ECOM_CMD_ACK (00)	•	•	•	•	•	•	•	•	•	•	•
SBG_ECOM_CMD_SETTINGS_ACTION (01)	•	•	•	•	•	•	•	•	•	•	•
SBG_ECOM_CMD_IMPORT_SETTINGS (02)	•	•	•	•	•	•	•	•	•	•	•
SBG_ECOM_CMD_EXPORT_SETTINGS (03)	•	•	•	•	•	•	•	•	•	•	•
SBG_ECOM_CMD_INFO (04)	•	•	•	•	•	•	•	•	•	•	•
SBG_ECOM_CMD_INIT_PARAMETERS (05)	•	•	•	•	•						
SBG_ECOM_CMD_MOTION_PROFILE_ID (07)		•	•	•	•						
SBG_ECOM_CMD_IMU_ALIGNMENT_LEVER_ARM (08)	•	•	•	•	•						
SBG_ECOM_CMD_AIDING_ASSIGNMENT (09)			•	•	•						
SBG_ECOM_CMD_MAGNETOMETER_MODEL_ID (11)		•	•	•	•						
SBG_ECOM_CMD_MAGNETOMETER_REJECT_MODE (12)		•	•	•	•						
SBG_ECOM_CMD_SET_MAG_CALIB (13)	•	•	•	•	•						
SBG_ECOM_CMD_START_MAG_CALIB (14)	•	•	•	•	•						
SBG_ECOM_CMD_COMPUTE_MAG_CALIB (15)	•	•	•	•	•						
SBG_ECOM_CMD_GNSS_MODEL_ID (17)			•	•	•						
SBG_ECOM_CMD_GNSS_1_LEVER_ARM_ALIGNMENT (18)			•	•	•						
SBG_ECOM_CMD_GNSS_1_INSTALLATION (46)			•	•	•						
SBG_ECOM_CMD_GNSS_1_REJECT_MODES (19)			•	•	•						
SBG_ECOM_CMD_ODO_CONF (20)			•	•	•						
SBG_ECOM_CMD_ODO_LEVER_ARM (21)			•	•	•						
SBG_ECOM_CMD_ODO_REJECT_MODE (22)			•	•	•						
SBG_ECOM_CMD_ODO_CAN_CONF (45)			•	•	•						
SBG_ECOM_CMD_UART_CONF (23)	•	•	•	•	•						
SBG_ECOM_CMD_CAN_BUS_CONF (24)	•	•	•	•	•						
SBG_ECOM_CMD_CAN_OUTPUT_CONF (25)	•	•	•	•	•						
SBG_ECOM_CMD_SYNC_IN_CONF (26)	•	•	•	•	•						
SBG_ECOM_CMD_SYNC_OUT_CONF (27)	•	•	•	•	•						
SBG_ECOM_CMD_NMEA_TALKER_ID (29)	•	•	•	•	•						
SBG_ECOM_CMD_OUTPUT_CONF (30)	•	•	•	•	•						
SBG_ECOM_CMD_ADVANCED_CONF (32)	•	•	•	•	•						
SBG_ECOM_CMD_FEATURES (33)	•	•	•	•	•						
SBG_ECOM_CMD_LICENSE_APPLY (34)					•						
SBG_ECOM_CMD_OUTPUT_CLASS_ENABLE (35)	•	•	•	•	•						
SBG_ECOM_CMD_ETHERNET_CONF (36)						•	•	•	•	•	•
SBG_ECOM_CMD_ETHERNET_INFO (37)						•	•	•	•	•	•
SBG_ECOM_CMD_VALIDITY_THRESHOLDS (38)	•	•	•	•	•						
SBG_ECOM_CMD_DVL_MODEL_ID (39)			•	•	•						
SBG_ECOM_CMD_DVL_INSTALLATION (40)			•	•	•						
SBG_ECOM_CMD_DVL_REJECT_MODES (41)			•	•	•						
SBG_ECOM_CMD_AIRDATA_MODEL_ID (42)			•	•	•						

	ELLIPSE					High Performance INS					
Name (Message ID)	I	A	E	N	D	A	E	N	D	M	U
SBG_ECOM_CMD_AIRDATA_LEVER_ARM (43)			•	•	•						
SBG_ECOM_CMD_AIRDATA_REJECT_MODES (44)			•	•	•						

2.2.5. Special settings commands

All these commands will affect the whole configuration, and therefore will engage a device reboot right after the ACK is returned.

2.2.5.1. SBG_ECOM_CMD_SETTINGS_ACTION (01)

This command provides special settings actions such as saving settings or restoring default settings. User calls this command with the payload as described below.

Payload description

Field	Description	Unit	Format	Size	Offset
settingAction	Special setting action to perform - Table 3	enum	uint8	1	0
			Total size		1

Once this command is received, the device do the requested action, send an ACK and finally reboot the device.

Enum definitiond

Name	Value	Description
SBG_ECOM_REBOOT_ONLY	0	No special action. The device will just reboot.
SBG_ECOM_SAVE_SETTINGS	1	The device will save new settings to non-volatile memory, and then reboot.
SBG_ECOM_RESTORE_DEFAULT_SETTINGS	2	Restore factory settings and calibration data, and then reboot.

Table 3: SbgEComSettingAction

2.2.5.2. SBG_ECOM_CMD_IMPORT_SETTINGS (02)

This command is used to send a buffer that contains a complete new set of settings to the device.

Thanks to this command, you can automate the device configuration for production proposes.

This command uses the transfer sub-protocol since the set of settings is too large to be sent in a unique sbgECom protocol frame. See 2.2.2 Large buffers transmission (Transfer sub-protocol) and more specifically 2.2.2.1 Sending data to the device for further information.

Field	Description	Unit	Format	Size	Offset
cmd	Transfer protocol CMD	enum	uint16	2	0
payload	Transfer protocol payload	-	-	0-n	1
			Total size		2+n

Once the transfer protocol is finalized, the device will send an ACK and reboot with new imported settings.

2.2.5.3. SBG_ECOM_CMD_EXPORT_SETTINGS (03)

You can export all the device settings to a buffer using this command. Use it in pair with the SBG_ECOM_CMD_IMPORT_SETTINGS (02) command to automate a device configuration.

This command uses the transfer sub-protocol since the set of settings is too large to be received in a unique sbgECom protocol frame. See 2.2.2 Large buffers transmission (Transfer sub-protocol) and more specifically 2.2.2.2 Receiving data from the device for further information.

Field	Description	Unit	Format	Size	Offset
cmd	Transfer protocol CMD	enum	uint16	2	0
payload	Transfer protocol payload	-	-	0-n	1
			Total size		2+n

2.2.6. *Information commands*

2.2.6.1. SBG_ECOM_CMD_INFO (04)

Sent without any payload, this frame is considered as an information request. The following answer is returned:

Field	Description	Unit	Format	Size	Offset
productCode	Human readable Product Code	-	string	32	0
serialNumber	Device serial number	-	uint32	4	32
calibationRev	Calibration data revision	-	uint32	4	36
calibrationYear	Device Calibration Year	-	uint16	2	40
calibrationMonth	Device Calibration Month	-	uint8	1	42
calibrationDay	Device Calibration Day	-	uint8	1	43
hardwareRev	Device hardware revision	-	uint32	4	44
firmwareRev	Firmware revision	-	uint32	4	48
				Total size	52

Note: Please check section Interpreting revision numbers for more details about how to read revision values.

2.2.7. Sensor parameters

2.2.7.1. SBG_ECOM_CMD_INIT_PARAMETERS (05)

This frame configures the initial position and date. These parameters are used for gravity and magnetic declination computation.

When sent without any parameter, it is used to retrieve the current configuration. When sent with full payload, it is then used for new configuration.

Payload description

Field	Description	Unit	Format	Size	Offset
initLat	Initial latitude	°	double	8	0
initLong	Initial longitude	°	double	8	8
initAlt	Initial altitude (above WGS84 ellipsoid)	m	double	8	16
year	Year at startup	-	uint16	2	24
month	month in year at startup	-	uint8	1	26
day	day in month at startup	-	uint8	1	27
				Total size	28

If a configuration is performed, device answer is an ACK.

Note: The new configuration will only be applied after SBG_ECOM_CMD_SETTINGS_ACTION (01) command is issued, with SBG_ECOM_SAVE_SETTINGS parameter.

SBG SYSTEMS

2.2.7.2. SBG_ECOM_CMD_MOTION_PROFILE_ID (07)

This command is used to define the motion profile ID to be used at next boot time, or read the current motion profile configuration.

Sent with only the motion profile ID as parameter, the sensor will try to setup the new motion profile ID.

Field	Description	Unit	Format	Size	Offset
motionProfileId	Requested Motion profile identifier	-	uint32	4	0
				Total size	4

Once this frame is received, the system will check if the motion profile ID requested is available and answer with an ACK.

Note: The new configuration will only be applied after SBG_ECOM_CMD_SETTINGS_ACTION (01) command is issued, with SBG_ECOM_SAVE_SETTINGS parameter.

Sending this frame without any payload will ask for the current motion profile configuration. The system will answer with the full payload version:

Field	Description	Unit	Format	Size	Offset
motionProfileId	Motion profile identifier	-	uint32	4	0
motionProfileRevision	Motion profile version number	-	uint32	4	4
				Total size	8

Note: Please check section Interpreting revision numbers for more details about how to read revision values.

Standard motion profile IDs

The following IDs may be used in standard. Other specific models may be developed upon special request.

Name	ID	Description
SBG_ECOM_MOTION_PROFILE_GENERAL_PURPOSE	1	Should be used as a default when other profiles do not apply
SBG_ECOM_MOTION_PROFILE_AUTOMOTIVE	2	Dedicated to car applications
SBG_ECOM_MOTION_PROFILE_MARINE	3	Used in marine and underwater applications
SBG_ECOM_MOTION_PROFILE_AIRPLANE	4	For fixed wings aircraft
SBG_ECOM_MOTION_PROFILE_HELICOPTER	5	For rotary wing aircraft
SBG_ECOM_MOTION_PROFILE_UAV	7	For UAV platforms with low dynamics

2.2.7.3. SBG_ECOM_CMD_IMU_ALIGNMENT_LEVER_ARM (08)

Sent without payload, this frame retrieves the current sensor alignment in vehicle frame as well as main lever arm. Sent with full payload, this frame configures the sensor alignment in vehicle.

Payload description

Field	Description	Unit	Format	Size	Offset
axisDirectionX	IMU X axis direction in vehicle frame - Table 4	enum	uint8	1	0
axisDirectionY	IMU Y axis direction in vehicle frame - Table 4	enum	uint8	1	1
misRoll	Residual roll error after axis alignment	rad	float	4	2
misPitch	Residual pitch error after axis alignment	rad	float	4	6
misYaw	Residual yaw error after axis alignment	rad	float	4	10
leverArm X	Primary lever arm in IMU X axis (once IMU alignment is applied)	m	float	4	14
leverArm Y	Primary lever arm in IMU Y axis (once IMU alignment is applied)	m	float	4	18
leverArm Z	Primary lever arm in IMU Z axis (once IMU alignment is applied)	m	float	4	22
				Total size	26

If a configuration is performed, device answer is an ACK.

Note: The new configuration will only be applied after SBG_ECOM_CMD_SETTINGS_ACTION (01) command is issued, with SBG_ECOM_SAVE_SETTINGS parameter.

Enum definitions

Name	Value	Description
SBG_ECOM_ALIGNMENT_FORWARD	0	IMU Axis is turned in vehicle's forward direction
SBG_ECOM_ALIGNMENT_BACKWARD	1	IMU Axis is turned in vehicle's backward direction
SBG_ECOM_ALIGNMENT_LEFT	2	IMU Axis is turned in vehicle's left direction
SBG_ECOM_ALIGNMENT_RIGHT	3	IMU Axis is turned in vehicle's right direction
SBG_ECOM_ALIGNMENT_UP	4	IMU Axis is turned in vehicle's up direction
SBG_ECOM_ALIGNMENT_DOWN	5	IMU Axis is turned in vehicle's down direction

Table 4: SbgEComAxisDirection enum

2.2.7.4. SBG_ECOM_CMD_AIDING_ASSIGNMENT (09)

Sent without payload, this frame retrieves the current aiding modules assignment (which interfaces connects to which module). Sent with full payload, this frame configures the aiding modules assignment.

Payload description

Field	Description	Unit	Format	Size	Offset
gnss1Port	GNSS 1 port assignment - Table 5	enum	uint8	1	0
gnss1Sync	GNSS 1 sync assignment for PPS - Table 6	enum	uint8	1	1
reserved	Reserved field for future use	-	uint32	4	2
dvlPort	DVL port assignment - Table 5	enum	uint8	1	6
dvlSync	DVL sync assignment used for time-stamping - Table 6	enum	uint8	1	7
rtcmPort	RTCM input port assignment for DGPS and RTK - Table 5	enum	uint8	1	8
airDataPort	True airspeed and altimeter port assignment - Table 5	enum	uint8	1	9
odometerPin	Odometer module pin assignment – Table 7	enum	uint8	1	10
				Total size	11

If a configuration is performed, device answer is an ACK.

Note 1: The new configuration will only be applied after SBG_ECOM_CMD_SETTINGS_ACTION (01) command is issued, with SBG_ECOM_SAVE_SETTINGS parameter.

Enum definitions

You can find below all enumerations and constant definitions used in this command.

Name	Value	Description
SBG_ECOM_MODULE_DISABLED	255	Module is disabled
SBG_ECOM_MODULE_PORT_A	0	Module connected on PORT_A
SBG_ECOM_MODULE_PORT_B	1	Module connected on PORT_B
SBG_ECOM_MODULE_PORT_C	2	Module connected on PORT_C
SBG_ECOM_MODULE_PORT_D	3	Module connected on PORT_D
SBG_ECOM_MODULE_PORT_E	4	Module connected on PORT_E
SBG_ECOM_MODULE_INTERNAL	5	Module is connected internally such as for internal GNSS receivers

Table 5: SbgEComModulePortAssignment enum

Name	Value	Description
SBG_ECOM_MODULE_SYNC_DISABLED	0	Module is disabled
SBG_ECOM_MODULE_SYNC_IN_A	1	Synchronization is done using SYNC_IN_A pin
SBG_ECOM_MODULE_SYNC_IN_B	2	Synchronization is done using SYNC_IN_B pin
SBG_ECOM_MODULE_SYNC_IN_C	3	Synchronization is done using SYNC_IN_C pin
SBG_ECOM_MODULE_SYNC_IN_D	4	Synchronization is done using SYNC_IN_D pin
SBG_ECOM_MODULE_SYNC_INTERNAL	5	Synchronization is internal
SBG_ECOM_MODULE_SYNC_OUT_A	6	Synchronization is done using SYNC_OUT_A pin
SBG_ECOM_MODULE_SYNC_OUT_B	7	Synchronization is done using SYNC_OUT_B pin

Table 6: SbgEComModuleSyncAssignment enum

Name	Value	Description
SBG_ECOM_MODULE_ODO_DISABLED	0	Odometer is disabled
SBG_ECOM_MODULE_ODO_A	1	Odometer connected only to ODO_A (unidirectional).
SBG_ECOM_MODULE_ODO_A_B	2	Odometer connected to both ODO_A (signal A) and ODO_B (Signal B or direction) for bidirectional odometer.
SBG_ECOM_MODULE_ODO_CAN	3	Odometer is read from the vehicle CAN bus

Table 7: SbgEComOdometerPinAssignment enum

2.2.8. Magnetometer configuration

2.2.8.1. SBG_ECOM_CMD_MAGNETOMETER_MODEL_ID (11)

This command is used to define the magnetometer model ID to be used at next boot time, or read the current motion profile configuration.

Sent with only the magnetometer model ID as parameter, the sensor will try to setup the new motion profile ID.

Field	Description	Unit	Format	Size	Offset
magModelId	Requested Magnetometer model identifier	-	uint32	4	0
				Total size	4

Once this frame is received, the system will check if the magnetometer model ID requested is available and answer with an ACK.

Note: The new configuration will only be applied after SBG_ECOM_CMD_SETTINGS_ACTION (01) command is issued, with SBG_ECOM_SAVE_SETTINGS parameter.

Sending this frame without any payload will ask for the current motion profile configuration. The system will answer with the full payload version:

Field	Description	Unit	Format	Size	Offset
magModelId	Magnetometer model identifier	-	uint32	4	0
magModelRevision	Magnetometer model version number	-	uint32	4	4
				Total size	8

Note: Please check section Interpreting revision numbers for more details about how to read revision values.

Standard magnetometer model IDs

The following IDs may be used in standard. Other specific models may be developed upon special request.

Name	ID	Description
SBG_ECOM_MAG_MODEL_NORMAL	201	Should be used in most applications
SBG_ECOM_MAG_MODEL_NOISY_MAG_TOLERANT	202	Should be used in disturbed magnetic environment

2.2.8.2. SBG_ECOM_CMD_MAGNETOMETER_REJECT_MODE (12)

Sent without parameter, this command asks for the magnetometer rejection mode. Sent with full payload, this command is used to configure the unit with provided rejection mode.

Payload description

Field	Description	Unit	Format	Size	Offset
magRejectMode	Rejection mode for magnetic field - Table 8	enum	uint8	1	0
				Total size	1

If a configuration is performed, device answer is an ACK.

Note: The new configuration will be applied immediately. Saving settings to flash after this command will save new parameters for next use.

Enum definitions

Name	Value	Description
SBG_ECOM_NEVER_ACCEPT_MODE	0	Measurement is not taken into account
SBG_ECOM_AUTOMATIC_MODE	1	Measurement is rejected if inconsistent with current estimate (depending on error model)
SBG_ECOM_ALWAYS_ACCEPT_MODE	2	Measurement is always accepted

Table 8: SbgEComRejectionMode enum

2.2.8.3. SBG_ECOM_CMD_SET_MAG_CALIB (13)

This command sends a new magnetic calibration to the ELLIPSE. A magnetic calibration is composed of a magnetic offset vector used to compensate for hard iron effects and a 3x3 correction matrix that accounts for soft iron distortions.

Field	Description	Unit	Format	Size	Offset
offset	Hard Iron correction vector Offset	-	Vector (float)	12	0
matrix	Hard & Soft Iron correction matrix	-	Matrix (float)	36	12
				Total size	48

If a configuration is performed, device answer is an ACK.

Note: The new configuration will be applied immediately. Saving settings to flash after this command will save new parameters for next use.

2.2.9. Magnetometer calibration

The ELLIPSE can perform an on-board magnetic calibration used to compensate both soft and hard iron effects.

The are several calibration modes that can be used to better accommodate application specificities.

2.2.9.1. SBG_ECOM_CMD_START_MAG_CALIB (14)

Start the magnetic calibration process. As soon as this command is sent, the device will start logging magnetic field data internally. This set of data will be used later by the magnetic calibration algorithms to map the surrounding magnetic field.

Field	Description	Unit	Format	Size	Offset
mode	Define which magnetic calibration type to perform - Table 9	enum	uint8	1	0
bandwidth	Specify the dynamics that will be experienced during the calibration - Table 10	enum	uint8	1	1
				Total size	2

If the command is correctly taken into account by the device, and ACK is answered.

Note: You can call multiple times this command. Each time this command is issued, the internal set of magnetic field data will be cleared acting as a reset feature.

Enum definitions

Name	Value	Description
SBG_ECOM_MAG_CALIB_2D	1	Tell the device that the magnetic calibration will be performed with limited motions. This calibration mode is only designed to be used when roll and pitch motions are less than ± 5°. To work correctly, the device should be rotated through at least a full circle.
SBG_ECOM_MAG_CALIB_3D	2	Tell the device to start a full 3D magnetic calibration procedure. The 3D magnetic calibration offers the best accuracy but needs at least motion of ± 30° on the roll and pitch angles.

Table 9: SbgEComMagCalibMode enum

Name	Value	Description
SBG_ECOM_MAG_CALIB_LOW_BW	0	Use this parameter in case of strong magnetic noise during calibration. Motion during calibration is then limited to slow rotations.
SBG_ECOM_MAG_CALIB_MEDIUM_BW	1	Tell the device that medium dynamics will be observed during the magnetic calibration process. It can be used in case of medium magnetic noise during calibration process. Medium dynamics are used during calibration.
SBG_ECOM_MAG_CALIB_HIGH_BW	2	This parameter is suitable to most applications. It can be used when the dynamics during calibration are relatively high.

Table 10: SbgEComMagCalibBandwidth enum

2.2.9.2. SBG_ECOM_CMD_COMPUTE_MAG_CALIB (15)

This command computes a magnetic calibration solution based on the magnetic field logged since the last call to the command SBG_ECOM_CMD_START_MAG_CALIB (14). As soon as the computations are done, the device will answer with quality indicators, status flags and if possible a valid magnetic calibration matrix and offset.

If a valid magnetic calibration has been computed, the returned offset and matrix fields should be sent to the device using the command SBG_ECOM_CMD_SET_MAG_CALIB (13) to apply the new calibration.

Sending the command SBG_ECOM_CMD_COMPUTE_MAG_CALIB (15) without payload will compute a new magnetic calibration based on previously logged magnetic field. As soon as the computations are done, the device will answer the following frame:

Field	Description	Unit	Format	Size	Offset
quality	General magnetic calibration quality indicator - Table 11 This indicator is computed based on the magnetic field norm deviation after calibration.	enum	uint8	1	0
confidence	Confidence indicator of the above quality indicator - Table 12 The better the magnetic field is mapped, the better the confidence indicator is. If you only cover very few portions of the 3D sphere or 2D circle, the confidence indicator will be low.	enum	uint8	1	1
advancedStatus	Set of bit masks used to report advanced information on the magnetic calibration status - Table 13 Read this status to get more details on a magnetic calibration that has failed.	Mask	uint16	2	2
beforeMeanError	Mean magnetic field norm error observed before calibration.	-	float	4	4
beforeStdError	Standard deviation of the magnetic field norm error observed before calibration.	-	float	4	8
beforeMaxError	Maximum magnetic field norm error observed before calibration.	-	float	4	12
afterMeanError	Mean magnetic field norm error observed after calibration.	-	float	4	16
afterStdError	Standard deviation of the magnetic field norm error observed after calibration.	-	float	4	20
afterMaxError	Maximum magnetic field norm error observed after calibration.	-	float	4	24
meanAccuracy	Mean expected heading accuracy in radians.	-	float	4	28
stdAccuracy	Standard deviation of the expected heading accuracy in radians.	-	float	4	32
maxAccuracy	Maximum expected heading accuracy in radians.	-	float	4	36
numPoints	Number of magnetic field points stored internally and used to compute the magnetic calibration.	-	uint16	2	40
maxNumPoints	Maximum number of magnetic field points that can be stored internally.	-	uint16	2	42
offset	Computed Hard Iron correction vector offset.	-	Vector (float)	12	44
matrix	Computed Hard & Soft Iron correction matrix.	-	Matrix (float)	36	56
				Total size	92

Enum definition

Name	Value	Description
SBG_ECOM_MAG_CALIB_QUAL_OPTIMAL	0	All acquired points fit very well on a unit sphere after the calibration.
SBG_ECOM_MAG_CALIB_QUAL_GOOD	1	Small deviations of the magnetic field norm have been detected. The magnetic calibration should although provide accurate heading.
SBG_ECOM_MAG_CALIB_QUAL_POOR	2	Large deviations of the magnetic field norm have been detected. It may come from external magnetic distortions during the calibration.
SBG_ECOM_MAG_CALIB_QUAL_INVALID	3	No valid magnetic calibration has been computed. It could comes from too much magnetic disturbances, insufficient or invalid motions.

Table 11: SbgEComMagCalibQuality enum

Name	Value	Description
SBG_ECOM_MAG_CALIB_TRUST_HIGH	0	Reported quality indicator can be trusted as enough remarkable magnetic field points have been acquired.
SBG_ECOM_MAG_CALIB_TRUST_MEDIUM	1	Few remarkable magnetic field points have been used to compute the magnetic calibration leading to a medium confidence in reported quality indicators.
SBG_ECOM_MAG_CALIB_TRUST_LOW	2	Even if the quality indicator could report an excellent calibration, The data set used to compute the magnetic calibration was not meaningful enough to compute meaningful quality indicators. This calibration should be used carefully.

Table 12: SbgEComMagCalibConfidence enum

Name	Value	Description
SBG_ECOM_MAG_CALIB_NOT_ENOUGH_POINTS	0x0001	Not enough valid magnetic points have been acquired.
SBG_ECOM_MAG_CALIB_TOO_MUCH_DISTORTIONS	0x0002	Unable to compute a magnetic calibration due to magnetic interference or incorrect data set distribution.
SBG_ECOM_MAG_CALIB_X_MOTION_ISSUE	0x0004	For a 3D calibration: not enough motion on X axis For a 2D calibration; too much motion on X axis
SBG_ECOM_MAG_CALIB_Y_MOTION_ISSUE	0x0008	For a 3D calibration: not enough motion on Y axis For a 2D calibration; too much motion on Y axis
SBG_ECOM_MAG_CALIB_Z_MOTION_ISSUE	0x0010	For a 3D or 2D calibration: not enough motion on Z axis
SBG_ECOM_MAG_CALIB_ALIGNMENT_ISSUE	0x0020	For a 3D calibration: the alignment between the magnetometers and the inertial frame seems to be invalid.

Table 13: SbgEComMagCalibAdvStatus mask

Note 1: The call to this command doesn't reset the logged magnetic field so you can safely call this method multiple time until you get the desired magnetic calibration results.

Note 2: Whereas most commands provide an answer immediately, this calibration command requires some time to compute the output. User should consider a few seconds delay before reading out the answer.

In addition, several output logs may be sent before this command answer is issued.

Note 3: While the calibration computation is running, the system stops logging new magnetic field data. Magnetic field data logging starts again as soon as the computations are finished.

2.2.10. GNSS configuration

2.2.10.1. SBG_ECOM_CMD_GNSS_MODEL_ID (17)

This command is used to define the GNSS model ID to be used at next boot time, or read the current GNSS model ID configuration.

Sent with only the GNSS model ID as parameter, the sensor will try to setup the new model ID.

Field	Description	Unit	Format	Size	Offset
gnssModelId	Requested GNSS model identifier	-	uint32	4	0
				Total size	4

Once this frame is received, the system will check if the GNSS model ID requested is available and answer with an ACK.

Note: The new configuration will only be applied after SBG_ECOM_CMD_SETTINGS_ACTION (01) command is issued, with SBG_ECOM_SAVE_SETTINGS parameter.

Sending this frame without any payload will ask for the current GNSS model configuration. The system will answer with the full payload version:

Field	Description	Unit	Format	Size	Offset
gnssModelId	GNSS model identifier	-	uint32	4	0
reserved	Reserved, ignore returned value	-	uint32	4	4
				Total size	8

Standard GNSS model IDs

The following IDs can be used in standard. Other specific models may be developed upon special request.

Name	ID	Description
SBG_ECOM_GNSS_MODEL_INTERNAL	101	Default internal GNSS for ELLIPSE-N & ELLIPSE-D
SBG_ECOM_GNSS_MODEL_NMEA	102	Used on ELLIPSE-E to accept an external GNSS using NMEA protocol
SBG_ECOM_GNSS_MODEL_UBLOX_GPS_BEIDOU	103	Used on ELLIPSE-N hw 1&2 to setup the GNSS in GPS+BEIDOU
SBG_ECOM_GNSS_MODEL_UBLOX_EXTERNAL	104	Used on ELLIPSE-E to accept an external Ublox (read only)
SBG_ECOM_GNSS_MODEL_NOVATEL_EXTERNAL	106	Used on ELLIPSE-E to accept an external Novatel (read only)
SBG_ECOM_GNSS_MODEL_SEPTENTRIO_EXTERNAL	109	Used on ELLIPSE-E to accept an external Septentrio (read only)

Note 1: For ELLIPSE-N hardware revision 1 & 2, the internal GNSS receiver default model uses GPS+GLONASS constellations and SBG_ECOM_GNSS_MODEL_UBLOX_GPS_BEIDOU can be used to select GPS+BeiDou instead.

2.2.10.2. SBG_ECOM_CMD_GNSS_1_LEVER_ARM_ALIGNMENT (18)

Sent without parameter, this command asks for the current GNSS lever arm and dual antenna alignment. Sent with full payload, this command configures the GNSS lever arm and dual antenna parameters.

Field	Description	Unit	Format	Size	Offset
leverArmX	GNSS antenna lever arm in IMU X axis	m	float	4	0
leverArmY	GNSS antenna lever arm in IMU Y axis	m	float	4	4
leverArmZ	GNSS antenna lever arm in IMU Z axis	m	float	4	8
pitchOffset	Pitch offset for dual antenna GNSS	rad	float	4	12
yawOffset	Yaw offset for dual antenna GNSS	rad	float	4	16
antennaDistance	Distance between two GNSS antennas	m	float	4	20
				Total size	24

If a configuration is performed, device answer is an ACK.

Note 1: The new configuration will only be applied after SBG_ECOM_CMD_SETTINGS_ACTION (01) command is issued, with SBG_ECOM_SAVE_SETTINGS parameter.

Note 2: Pitch and Yaw offsets as well as antenna distance parameters should only be considered in case of dual antenna GNSS receiver. It can be left to 0 otherwise.

DEPRECATED: This command has been deprecated and replaced by the command SBG_ECOM_CMD_GNSS_1_INSTALLATION (46)

2.2.10.3. SBG_ECOM_CMD_GNSS_1_INSTALLATION (46)

Sent without parameter, this command asks for the GNSS main and secondary antenna lever arms. For single antenna operation or if the secondary antenna is not known/filled, a zero lever arm will be returned.

Sent with full payload, this command configures the main and secondary GNSS lever arm as well as which mechanical installation parameter mode should be used.

Field	Description	Unit	Format	Size	Offset
leverArmPrimaryX	GNSS primary antenna lever arm in IMU X axis	m	float	4	0
leverArmPrimaryY	GNSS primary antenna lever arm in IMU Y axis	m	float	4	4
leverArmPrimaryZ	GNSS primary antenna lever arm in IMU Z axis	m	float	4	8
leverArmPrimaryPrecise	True if the primary lever arm has been entered accurately	-	bool	1	12
leverArmSecondaryX	GNSS secondary antenna lever arm in IMU X axis	rad	float	4	13
leverArmSecondaryY	GNSS secondary antenna lever arm in IMU Y axis	rad	float	4	17
leverArmSecondaryZ	GNSS secondary antenna lever arm in IMU Z axis	m	float	4	21
leverArmSecondaryMode	Secondary antenna installation mode - Table 14	enum	uint8	1	25
				Total size	26

If a configuration is performed, device answer is an ACK.

Note 1: The new configuration will only be applied after SBG_ECOM_CMD_SETTINGS_ACTION (01) command is issued, with SBG_ECOM_SAVE_SETTINGS parameter.

Note 2: Secondary lever arm should only be filled for dual antenna operations. Please leave it to zero if not use and fill leverArmSecondaryMode accordingly.

Enum definition

Name	Value	Description
SBG_ECOM_GNSS_INSTALLATION_MODE_SINGLE	1	Single antenna mode, secondary lever arm is ignored
SBG_ECOM_GNSS_INSTALLATION_MODE_DUAL_AUTO	2	Reserved, do not use
SBG_ECOM_GNSS_INSTALLATION_MODE_DUAL_ROUGH	3	Dual antenna mode, a rough secondary lever arm guess is provided
SBG_ECOM_GNSS_INSTALLATION_MODE_DUAL_PRECISE	4	Dual antenna mode, a precise secondary lever arm is provided

Table 14: GNSS secondary installation mode enum

2.2.10.4. SBG_ECOM_CMD_GNSS_1_REJECT_MODES (19)

Sent without parameter, this command asks for the GNSS receiver its rejection modes. Sent with full payload, this command is used to configure the unit with provided rejection modes.

Payload description

Field	Description	Unit	Format	Size	Offset
posRejectMode	Rejection mode for position - Table 8	enum	uint8	1	0
velRejectMode	Rejection mode for velocity - Table 8	enum	uint8	1	1
reserved	Reserved field – leave to 0	enum	uint8	1	2
hdtRejectMode	Rejection mode for true heading - Table 8	enum	uint8	1	3
				Total size	4

If a configuration is performed, device answer is an ACK.

Note: The new configuration will be applied immediately. Saving settings to flash after this command will save new parameters for next use.

SBG SYSTEMS

2.2.11. DVL configuration

2.2.11.1. SBG_ECOM_CMD_DVL_MODEL_ID (39)

This command is used to define the DVL model ID to be used at next boot time, or read the current DVL model ID configuration. The DVL model is used to both define the protocol to use as well as the associated error model.

Sent with only the DVL model ID as parameter, the sensor will try to setup the new model ID.

Field	Description	Unit	Format	Size	Offset
dvlModelId	DVL model identifier	-	uint32	4	0
				Total size	4

Once this frame is received, the system will check if the DVL model ID requested is available and answer with an ACK.

Note: The new configuration will only be applied after SBG_ECOM_CMD_SETTINGS_ACTION (01) command is issued, with SBG_ECOM_SAVE_SETTINGS parameter.

Sending this frame without any payload will ask for the current DVL model ID configuration. The system will answer with the following payload:

Field	Description	Unit	Format	Size	Offset
dvlModelId	DVL model identifier	-	uint32	4	0
				Total size	4

Standard DVL model IDs

The following IDs can be used in standard. Other specific models may be developed upon special request.

Name	ID	Description
SBG_ECOM_DVL_MODEL_GENERIC_PD6	202	Generic DVL model using PD6 protocol format

2.2.11.2. SBG_ECOM_CMD_DVL_INSTALLATION (40)

Sent without parameter, this command asks for the current DVL lever arm and alignment. Sent with full payload, this command configures the DVL lever arm and alignment parameters.

Field	Description	Unit	Format	Size	Offset
leverArmX	DVL lever arm in IMU X axis	m	float	4	0
leverArmY	DVL lever arm in IMU Y axis	m	float	4	4
leverArmZ	DVL lever arm in IMU Z axis	m	float	4	8
rollOffset	DVL roll alignment offset	rad	float	4	12
pitchOffset	DVL pitch alignment offset	rad	float	4	16
yawOffset	DVL yaw alignment offset	rad	float	4	20
preciseInstallation	True if lever arm & alignment have been entered accurately	-	bool	1	21
				Total size	25

If a configuration is performed, device answer is an ACK.

> **Note 1:** The new configuration will only be applied after SBG_ECOM_CMD_SETTINGS_ACTION (01) command is issued, with SBG_ECOM_SAVE_SETTINGS parameter.

2.2.11.3. SBG_ECOM_CMD_DVL_REJECT_MODES (41)

Sent without parameter, this command asks for the DVL rejection modes. Sent with full payload, this command is used to configure the rejection modes for the DVL.

Payload description

Field	Description	Unit	Format	Size	Offset
bottomLayer	Rejection mode for the bottom tracking velocity - Table 8	enum	uint8	1	0
waterLayer	Rejection mode for the water layer velocity - Table 8	enum	uint8	1	1
				Total size	2

If a configuration is performed, device answer is an ACK.

> **Note:** The new configuration will be applied immediately. Saving settings to flash after this command will save new parameters for next use.

2.2.12. Odometer configuration

2.2.12.1. SBG_ECOM_CMD_ODO_CONF (20)

Sent without parameter, this frame retrieves the configuration for an odometer using quadrature/pulses. The device will answer by the same frame with full payload.

When sent with full payload, this frame configures new odometer's parameters.

Payload description

Field	Description	Unit	Format	Size	Offset
GAIN	Odometer's gain	Pulses/m	float	4	0
GAIN_ERROR	User gain average error	%	uint8	1	4
DIRECTION	Odometer's direction: 0: positive 1: negative	-	bool	1	5
			Total size		6

If a configuration is performed, device answer is an ACK.

Note: The new configuration will only be applied after SBG_ECOM_CMD_SETTINGS_ACTION (01) command is issued, with SBG_ECOM_SAVE_SETTINGS parameter.

2.2.12.2. SBG_ECOM_CMD_ODO_LEVER_ARM (21)

Sent without parameter, this frame retrieves the odometer lever arm. The device will answer by the same frame with full payload.

When sent with full payload, this frame configures new odometer's lever arm.

Payload description

Field	Description	Unit	Format	Size	Offset
leverArmX	Odometer lever arm in IMU X axis	m	float	4	0
leverArmY	Odometer lever arm in IMU Y axis	m	float	4	4
leverArmZ	Odometer lever arm in IMU Z axis	m	float	4	8
			Total size		12

If a configuration is performed, device answer is an ACK.

Note: The new configuration will only be applied after SBG_ECOM_CMD_SETTINGS_ACTION (01) command is issued, with SBG_ECOM_SAVE_SETTINGS parameter.

2.2.12.3. SBG_ECOM_CMD_ODO_REJECT_MODE (22)

Sent without parameter, this command asks for the odometer rejection mode. Sent with full payload, this command is used to configure the unit with provided rejection mode.

Payload description

Field	Description	Unit	Format	Size	Offset
rejectMode	Rejection mode for odometer velocity - Table 8	enum	uint8	1	0
			Total size		1

If a configuration is performed, device answer is an ACK.

Note: The new configuration will be applied immediately. Saving settings to flash after this command will save new parameters for next use.

2.2.12.4. SBG_ECOM_CMD_ODO_CAN_CONF (45)

This command defines or retrieves the configuration for odometer over a CAN bus. Each automotive odometer manufacturer is using specific CAN messages. With this command you can define which CAN message contains the odometer velocity information and provide all necessary information to decode it.

The same way, if the vehicle supports it, you can setup a CAN message to detect the vehicle direction of motion (velocity sign / reverse detection).

To handle all type of vehicles CAN bus, you can setup which CAN message to parse and how to decode its payload based on a simple mechanism similar to CAN DBC messages definition. You thus define where, in the CAN payload, the field of interest starts, its length in bits and the scaling / offset to apply to get a physical value such as a velocity in m.s^1.

Setting request payload

To request the CAN odometer configuration for a specific CAN channel information, the following payload has to be sent.

Field	Description	Unit	Format	Size	Offset
canChannel	The CAN channel to define associated DBC configuration - Table 15	enum	uint8	1	0
			Total size		1

The device will answer with the full payload version as defined below to retrieve the current configuration for the requested channel.

Setting configuration payload

To define the CAN odometer configuration for a specific channel, please send the full payload defined below to the device. This payload is also sent by the device as an answer to a setting request payload.

Field	Description	Unit	Format	Size	Offset
canChannel	The CAN channel to define associated DBC configuration - Table 15	enum	uint8	1	0
options	Set of options bit masks such as CAN extended – Table 16	mask	uint16	2	1
canId	CAN msg ID from which the odometer velocity will be parsed	-	uint32	4	3
dataOffset	Index of first data bit within the payload as defined by DBC (from 0 to 63).	-	uint8	1	7
dataSize	Length in bits of the odometer velocity field (1 to 64)	-	uint8	1	8
scale	Value to multiply the parsed field with to get physical unit	LSB/m.s^{-1}	float	4	9
offset	Offset to add on the scaled velocity information	m.s^{-1}	float	4	13
minValue	The minimum velocity to consider the message valid	m.s^{-1}	float	4	17
maxValue	The maximum velocity to consider the message valid	m.s^{-1}	float	4	21
			Total size		25

If a configuration is performed, device answer is an ACK.

Note: The new configuration will only be applied after SBG_ECOM_CMD_SETTINGS_ACTION (01) command is issued, with SBG_ECOM_SAVE_SETTINGS parameter.

Warning: DBC standard use a counter-intuitive bit numbering that depends from endianness! Please refer to DBC specifications to correctly fill the dataOffset field.

Enum definition

Name	Value	Description
SBG_ECOM_CMD_ODO_CAN_CH_VELOCITY	1	Channel used to decode the vehicle velocity information
SBG_ECOM_CMD_ODO_CAN_CH_REVERSE	2	Channel used to decode the vehicle velocity reverse info (if available).

Table 15: Odometer CAN channels enum

Name	Value	Description
SBG_ECOM_CMD_ODO_CAN_ENABLE	0x0001	Set to enable CAN odometer information decoding
SBG_ECOM_CMD_ODO_CAN_ID_EXTENDED	0x0002	Set for a 29 bit extended CAN message, otherwise standard 11 bit
SBG_ECOM_CMD_ODO_CAN_BIG_ENDIAN	0x0003	Set if the velocity is encoded in big endian, otherwise little endian
SBG_ECOM_CMD_ODO_CAN_SIGNED	0x0004	Set to interpret the parsed value as signed, otherwise unsigned

Table 16: Odometer CAN decoding options bitmask

2.2.13. Airdata configuration

2.2.13.1. SBG_ECOM_CMD_AIRDATA_MODEL_ID (42)

This command is used to define the AirData model ID to be used at next boot time, or read the current AirData model ID configuration. The AirData model is used to both define the protocol to use as well as the associated error model.

Sent with only the model ID as parameter, the sensor will try to setup the new model ID.

Field	Description	Unit	Format	Size	Offset
airDataModelId	AirData model identifier to set	-	uint32	4	0
			Total size		4

Once this frame is received, the system will check if the model ID requested is available and answer with an ACK.

Note: The new configuration will only be applied after SBG_ECOM_CMD_SETTINGS_ACTION (01) command is issued, with SBG_ECOM_SAVE_SETTINGS parameter.

Sending this frame without any payload will ask for the current AirData model ID configuration. The system will answer with the following payload:

Field	Description	Unit	Format	Size	Offset
airDataModelId	AirData model identifier	-	uint32	4	0
			Total size		4

Standard AirData model IDs

The following IDs can be used in standard. Other specific models may be developed upon special request.

Name	ID	Description
SBG_ECOM_AIRDATA_MODEL_INTERNAL	1	Will use the internal barometer sensor if available
SBG_ECOM_AIRDATA_MODEL_GENERIC_ECOM	2	Generic AirData model using sbgECom protocol format
SBG_ECOM_AIR_DATA_MODEL_AHRS_500	3	Crossbow AHRS-500 compatible input for altitude and airspeed

2.2.13.2. SBG_ECOM_CMD_AIRDATA_LEVER_ARM (43)

Sent without parameter, this command asks for the current AirData lever arm setting. Sent with full payload, this command configures the AirData lever arm parameters.

Field	Description	Unit	Format	Size	Offset
leverArmX	Airspeed sensor lever arm in IMU X axis	m	float	4	0
leverArmY	Airspeed sensor lever arm in IMU Y axis	m	float	4	4
leverArmZ	Airspeed sensor lever arm in IMU Z axis	m	float	4	8
			Total size		12

If a configuration is performed, device answer is an ACK.

Note 1: The new configuration will only be applied after SBG_ECOM_CMD_SETTINGS_ACTION (01) command is issued, with SBG_ECOM_SAVE_SETTINGS parameter.

2.2.13.3. SBG_ECOM_CMD_AIRDATA_REJECT_MODES (44)

Sent without parameter, this command asks for the AirData rejection modes. Sent with full payload, this command is used to configure the rejection modes for the DVL.

Payload description

Field	Description	Unit	Format	Size	Offset
airSpeed	Rejection mode for the true air speed aiding - Table 8	-	enum	1	0
altitude	Rejection mode for the barometer altitude aiding - Table 8	-	enum	1	1
			Total size		2

If a configuration is performed, device answer is an ACK.

Note: The new configuration will be applied immediately. Saving settings to flash after this command will save new parameters for next use.

2.2.14. Interfaces configuration

2.2.14.1. SBG_ECOM_CMD_UART_CONF (23)

When this frame is sent with the limited payload, this frame is used to retrieve a UART port configuration.

Setting request payload description

Field	Description	Unit	Format	Size	Offset
portID	Port ID to returns its configuration	-	enum	1	0
			Total size		1

Device will answer with the full payload version. If sent using full payload, the device will configure the requested interface according to new settings.

Setting configuration / return payload description

Field	Description	Unit	Format	Size	Offset
portID	Port ID to configure or retrieved configuration - Table 17	-	enum	1	0
baudRate	Baudrate to use for communications. Possible values are: • 4 800 • 9 600 • 19 200 • 38 400 • 115 200 • 230 400 • 460 800 • 921 600	bps	uint32	4	1
mode	UART mode - Table 18	-	enum	1	5
			Total size		6

If a configuration is performed, device answer is an ACK.

Note: The new configuration will only be applied after SBG_ECOM_CMD_SETTINGS_ACTION (01) command is issued, with SBG_ECOM_SAVE_SETTINGS parameter.

Enum definitions

Name	Value	Description
SBG_ECOM_PORT_A	0	Main communication interface. Full duplex.
SBG_ECOM_PORT_B	1	Auxiliary input interface for RTCM
SBG_ECOM_PORT_C	2	Auxiliary communication interface. Full duplex.
SBG_ECOM_PORT_D	3	Auxiliary input interface
SBG_ECOM_PORT_E	4	Auxiliary input/output interface

Table 17: SbgEComPortId enum

Name	Value	Description
SBG_ECOM_UART_MODE_OFF	0	This interface is turned OFF.
SBG_ECOM_UART_MODE_232	1	This interface is using RS-232 communications
SBG_ECOM_UART_MODE_422	2	This interface is using RS-422 communications

Table 18: SbgEComPortMode enum

Note: There may be limitations regarding the ports configuration. Please refer to your product Hardware Reference Manual for valid configuration and option availability.

2.2.14.2. SBG_ECOM_CMD_CAN_BUS_CONF (24)

Sent without any payload, this frame retrieves the CAN bus configuration. Sent with full payload, the frame configures the CAN bus interface.

Full payload description

Field	Description	Unit	Format	Size	Offset
bitRate	Bit rate used on the CAN bus - Table 19	-	enum	2	0
mode	CAN bus mode - Table 20	-	enum	1	2
			Total size		3

If a configuration is performed, device answer is an ACK.

Note: The new configuration will only be applied after SBG_ECOM_CMD_SETTINGS_ACTION (01) command is issued, with SBG_ECOM_SAVE_SETTINGS parameter.

Enum definitions

Name	Value	Description
SBG_ECOM_CAN_BITRATE_DISABLED	0	The CAN interface is disabled
SBG_ECOM_CAN_BITRATE_10	10	CAN interface speed set to 10 Kb/s
SBG_ECOM_CAN_BITRATE_20	20	CAN interface speed set to 20 Kb/s
SBG_ECOM_CAN_BITRATE_25	25	CAN interface speed set to 25 Kb/s
SBG_ECOM_CAN_BITRATE_50	50	CAN interface speed set to 50 Kb/s
SBG_ECOM_CAN_BITRATE_100	100	CAN interface speed set to 100 Kb/s
SBG_ECOM_CAN_BITRATE_125	125	CAN interface speed set to 125 Kb/s
SBG_ECOM_CAN_BITRATE_250	250	CAN interface speed set to 250 Kb/s
SBG_ECOM_CAN_BITRATE_500	500	CAN interface speed set to 500 Kb/s
SBG_ECOM_CAN_BITRATE_750	750	CAN interface speed set to 750 Kb/s
SBG_ECOM_CAN_BITRATE_1000	1000	CAN interface speed set to 1 Mb/s

Table 19: SbgEComCanBitRate enum

Name	Value	Description
SBG_ECOM_CAN_MODE_SPY	1	CAN is only listening and
SBG_ECOM_CAN_MODE_NORMAL	2	The device is allowed to both send and receive over the CAN bus

Table 20: SbgEComCanMode enum

2.2.14.3. SBG_ECOM_CMD_CAN_OUTPUT_CONF (25)

Sent with the limited payload, this frame retrieves a specific CAN message ID and output mode given its internal ID.

Limited payload description

Field	Description	Unit	Format	Size	Offset
canInternalID	Internal CAN message ID - CAN Messages Overview	-	enum	2	0
			Total size		2

Device will answer with the full payload version. If the frame is sent using full payload, the device will configure the requested interface according to new settings.

Full payload description

Field	Description	Unit	Format	Size	Offset
canInternalID	Internal CAN message ID - CAN Messages Overview	-	enum	2	0
outputMode	Output mode - Table 28	-	enum	2	2
userID	User defined CAN message identifier	-	uint32	4	4
canExtID	CAN Standard / Extended ID type. Possible values are: • TRUE: Extended ID • FALSE: Standard ID	-	bool	1	8
			Total size		9

If a configuration is performed, device answer is an ACK.

Note: The new configuration will only be applied after SBG_ECOM_CMD_SETTINGS_ACTION (01) command is issued, with SBG_ECOM_SAVE_SETTINGS parameter.

2.2.15. Ethernet Configuration

2.2.15.1. SBG_ECOM_CMD_ETHERNET_CONF (36)

Sent without payload, this frame retrieves the current device Ethernet configuration.

> **Note:** If you would like to read the IP address currently used by the device, please use the command SBG_ECOM_CMD_ETHERNET_INFO (37)

Setting configuration / return payload description

Field	Description	Unit	Format	Size	Offset
mode	Define DHCP or Static IP mode - Table 21	-	enum	1	0
ipAddress	For static mode, defines the IP address	-	uint32	4	1
netmask	For static mode, defines the subnet mask address	-	uint32	4	5
gateway	For static mode, defines the default gateway address	-	uint32	4	9
dns1	For static mode, defines the primary DNS address	-	uint32	4	13
dns2	For static mode, defines the secondary DNS address	-	uint32	4	17
				Total size	21

If a configuration is performed, device answer is an ACK.

> **Note:** The new configuration will only be applied after SBG_ECOM_CMD_SETTINGS_ACTION (01) command is issued, with SBG_ECOM_SAVE_SETTINGS parameter.

Enum definition

Name	Value	Description
SBG_ECOM_ETHERNET_DHCP	0	The TCP/IP configuration should be acquired from a DHCP server
SBG_ECOM_ETHERNET_STATIC	1	The TCP/IP configuration is manually defined

Table 21: SbgEComEthernetMode enum

2.2.15.2. SBG_ECOM_CMD_ETHERNET_INFO (37)

This command is used to read the TCP/IP parameters currently used by the device such as the IP address.

Return payload description

Field	Description	Unit	Format	Size	Offset
mode	Define DHCP or Static IP mode - Table 21	-	enum	1	0
ipAddress	For static mode, defines the IP address	-	uint32	4	1
netmask	For static mode, defines the subnet mask address	-	uint32	4	5
gateway	For static mode, defines the default gateway address	-	uint32	4	9
dns1	For static mode, defines the primary DNS address	-	uint32	4	13
dns2	For static mode, defines the secondary DNS address	-	uint32	4	17
				Total size	21

2.2.16. Events configuration

2.2.16.1. SBG_ECOM_CMD_SYNC_IN_CONF (26)

Sent with a limited payload, this frame retrieves the given Synchronization Input configuration.

Limited payload description

Field	Description	Unit	Format	Size	Offset
syncInId	Logic input pin ID - Table 22	-	enum	1	0
				Total size	1

Device will answer with the full payload version. If the frame is sent using full payload, the device will configure the requested logic input according to new settings.

Full payload description

Field	Description	Unit	Format	Size	Offset
syncInId	Logic input pin ID - Table 22	-	enum	1	0
sensitivity	Logic input corresponding sensitivity - Table 23	-	enum	1	1
delayNs	Delay added or subtracted to this event input for time-stamping	ns	int32	4	5
				Total size	6

If a configuration is performed, device answer is an ACK.

Note: The new configuration will only be applied after SBG_ECOM_CMD_SETTINGS_ACTION (01) command is issued, with SBG_ECOM_SAVE_SETTINGS parameter.

Enum definitions

Name	Value	Description
SBG_ECOM_SYNC_IN_A	0	Sync IN A on Main connector
SBG_ECOM_SYNC_IN_B	1	Sync IN B on Main connector
SBG_ECOM_SYNC_IN_C	2	Sync IN C on Auxiliary connector
SBG_ECOM_SYNC_IN_D	3	Sync IN D on Auxiliary connector

Table 22: SbgEComSyncInID enum

Name	Value	Description
SBG_ECOM_SYNC_IN_DISABLED	0	This trigger is turned OFF.
SBG_ECOM_SYNC_IN_FALLING_EDGE	1	The trigger will be activated by a falling edge
SBG_ECOM_SYNC_IN_RISING_EDGE	2	The trigger will be activated by a rising edge
SBG_ECOM_SYNC_IN_BOTH_EDGES	3	The trigger is activated by a level change (rising or falling edge)

Table 23: SbgEComSyncInSensitivity enum

2.2.16.2. SBG_ECOM_CMD_SYNC_OUT_CONF (27)

Sent with a limited payload, this frame retrieves the given Synchronization Output configuration.

Limited payload description

Field	Description	Unit	Format	Size	Offset
syncOutId	Logic output pin ID - Table 24	-	enum	1	0
				Total size	1

Device will answer with the full payload version. If the frame is sent using full payload, the device will configure the requested logic output pin according to new settings.

Full payload description

Field	Description	Unit	Format	Size	Offset
syncOutId	Logic output pin ID - Table 24	-	enum	1	0
reserved	-	-	uint8	1	1
outputFunction	Main function of the sync Out pin - Table 25	-	enum	2	2
polarity	Polarity of corresponding logic output - Table 26	-	enum	1	4
durationNs	Pulse duration	ns	uint32	4	5
				Total size	9

If a configuration is performed, device answer is an ACK.

Note: The new configuration will only be applied after SBG_ECOM_CMD_SETTINGS_ACTION (01) command is issued, with SBG_ECOM_SAVE_SETTINGS parameter.

Enum definitions

Name	Value	Description
SBG_ECOM_SYNC_OUT_A	0	Synchronization output signal A
SBG_ECOM_SYNC_OUT_B	1	Synchronization output signal B

Table 24: SbgEComSyncOutID enum

Name	Value	Description
SBG_ECOM_SYNC_OUT_MODE_DISABLED	0	Output is disabled
SBG_ECOM_SYNC_OUT_MODE_MAIN_LOOP	1	Output is generated at 200Hz
SBG_ECOM_SYNC_OUT_MODE_DIV_2	2	Output is generated at 100Hz
SBG_ECOM_SYNC_OUT_MODE_DIV_4	4	Output is generated at 50Hz
SBG_ECOM_SYNC_OUT_MODE_DIV_8	8	Output is generated at 25Hz
SBG_ECOM_SYNC_OUT_MODE_DIV_10	10	Output is generated at 20Hz
SBG_ECOM_SYNC_OUT_MODE_DIV_20	20	Output is generated at 10Hz
SBG_ECOM_SYNC_OUT_MODE_DIV_40	40	Output is generated at 5Hz
SBG_ECOM_SYNC_OUT_MODE_DIV_200	200	Output is generated at 1Hz
SBG_ECOM_SYNC_OUT_MODE_PPS	10000	Pulse Per Second. Same mode as above.
SBG_ECOM_SYNC_OUT_MODE_EVENT_IN_A	10003	Output is generated on a Sync In A event
SBG_ECOM_SYNC_OUT_MODE_EVENT_IN_B	10004	Output is generated on a Sync In B event
SBG_ECOM_SYNC_OUT_MODE_EVENT_IN_C	10005	Output is generated on a Sync In C event
SBG_ECOM_SYNC_OUT_MODE_EVENT_IN_D	10006	Output is generated on a Sync In D event
SBG_ECOM_SYNC_OUT_MODE_DIRECT_PPS	10100	The internal GNSS PPS signal is directly routed to the Sync Out. This mode is only valid for ELLIPSE-N with hardware revisions above 1.2.1.0 Polarity and duration parameters are ignored with this specific mode.

Table 25: SbgEComSyncOutFunction enum

Name	Value	Description
SBG_ECOM_LOGIC_OUT_FALLING_EDGE	0	The trigger will generate a falling edge
SBG_ECOM_LOGIC_OUT_RISING_EDGE	1	The trigger will generate a rising edge
SBG_ECOM_LOGIC_OUT_TOGGLE	2	The trigger is a level change

Table 26: SbgEComSyncOutPolarity enum

2.2.17. Output configuration

2.2.17.1. SBG_ECOM_CMD_NMEA_TALKER_ID (29)

This command can be used to read or set the NMEA Talker ID used on a specific output port.

Reading payload definitions

Field	Description	Unit	Format	Size	Offset
outputPortId	Port Name - Table 27	-	enum	1	0
				Total size	1

The device will answer with the full payload frame. Sent with this full payload, the frame will configure a new talker ID.

Return or setting configuration payload definition

Field	Description	Unit	Format	Size	Offset
outputPortId	Port Name - Table 27	-	enum	1	0
talker0	First character of the talker ID	-	char	1	1
talker1	Second character of the talker ID	-	char	1	1
				Total size	3

If a configuration is performed, device answer is an ACK.

Note: The new configuration will only be applied after SBG_ECOM_CMD_SETTINGS_ACTION (01) command is issued, with SBG_ECOM_SAVE_SETTINGS parameter.

Enum definitions

Name	Value	Description
SBG_ECOM_OUTPUT_PORT_A	0	Main output port
SBG_ECOM_OUTPUT_PORT_C	2	Auxiliary output interface for ELLIPSE-E
SBG_ECOM_OUTPUT_PORT_E	4	Miscellaneous output interface for B1 models

Table 27: SbgEComOutputPortID enum

2.2.17.2. SBG_ECOM_CMD_OUTPUT_CONF (30)

This command can be used either to configure or retrieve an output log.

Sent with the following payload, this frame asks the device for the output configuration

Limited payload definition

Field	Description	Unit	Format	Size	Offset
outputPortId	Port Name - Table 27	-	enum	1	0
msgId	Output message identifier		enum	1	1
classId	Output class identifier - Table 1		enum	1	1
				Total size	3

The device answer is the full output configuration, as defined in the following frame.

Sent with full payload, this frame configures an output.

Full Payload definitions

Field	Description	Unit	Format	Size	Offset
outputPortId	Port Name - Table 27	-	enum	1	0
msgId	Output message identifier		enum	1	1
classId	Output class identifier - Table 1		enum	1	1
outputMode	Output mode - Table 28	-	enum	2	3
				Total size	5

If a configuration is performed, device answer is an ACK.

Note: The new configuration will be applied immediately. Saving settings to flash after this command will save new parameters for next use.

Please check sections SBG_ECOM_CLASS_LOG_ECOM_0, NMEA Logs Overview and Third Party Logs Overview for more details about available messages IDs.

Enum definitions

Name	Value	Description
SBG_ECOM_OUTPUT_MODE_DISABLED	0	Output is disabled
SBG_ECOM_OUTPUT_MODE_MAIN_LOOP	1	Output is generated at 200Hz
SBG_ECOM_OUTPUT_MODE_DIV_2	2	Output is generated at 100Hz
SBG_ECOM_OUTPUT_MODE_DIV_4	4	Output is generated at 50Hz
SBG_ECOM_OUTPUT_MODE_DIV_5	5	Output is generated at 40Hz
SBG_ECOM_OUTPUT_MODE_DIV_8	8	Output is generated at 25Hz
SBG_ECOM_OUTPUT_MODE_DIV_10	10	Output is generated at 20Hz
SBG_ECOM_OUTPUT_MODE_DIV_20	20	Output is generated at 10Hz
SBG_ECOM_OUTPUT_MODE_DIV_40	40	Output is generated at 5Hz
SBG_ECOM_OUTPUT_MODE_DIV_200	200	Output is generated at 1Hz
SBG_ECOM_OUTPUT_MODE_PPS	10000	Pulse Per Second. Same mode as above.
SBG_ECOM_OUTPUT_MODE_NEW_DATA	10001	Output sent when a new data is available.
SBG_ECOM_OUTPUT_MODE_EVENT_IN_A	10003	Output is generated on a Sync In A event
SBG_ECOM_OUTPUT_MODE_EVENT_IN_B	10004	Output is generated on a Sync In B event
SBG_ECOM_OUTPUT_MODE_EVENT_IN_C	10005	Output is generated on a Sync In C event
SBG_ECOM_OUTPUT_MODE_EVENT_IN_D	10006	Output is generated on a Sync In D event

Table 28: SbgEComOutputMode enum

2.2.17.3. SBG_ECOM_CMD_OUTPUT_CLASS_ENABLE (35)

This command will act as a master switch for the different output classes on a specific PORT. It's possible to disable all sbgECom or NMEA messages from one port using a single command.

Sent with limited payload, the command will ask for the current configuration for a specific port and message class.

Limited payload definition

Field	Description	Unit	Format	Size	Offset
outputPortId	Port Name - Table 27	-	enum	1	0
classId	Output class identifier - Table 1	-	enum	1	1
				Total size	2

Sent with a full payload, this frames sets a new output configuration.

Full Payload definitions

Field	Description	Unit	Format	Size	Offset
outputPortId	Port Name - Table 27	-	enum	1	0
classId	Output class identifier - Table 1	-	enum	1	1
enable	Enable or Disable the message class on defined port	-	bool	1	2
				Total size	3

If a configuration is performed, device answer is an ACK.

2.2.18. *Advanced configuration*

2.2.18.1. SBG_ECOM_CMD_ADVANCED_CONF (32)

Sent without parameter, this command asks for the advanced configurations such as time reference source. Sent with full payload, this command is used to configure the advanced configuration.

Payload description

Field	Description	Unit	Format	Size	Offset
timeReference	Defines if an external clock reference must be used for synchronization - Table 29	-	enum	1	0
				Total size	1

If a configuration is performed, device answer is an ACK.

Enum definitions

Name	Value	Description
SBG_ECOM_TIME_REF_DISABLED	0	No external time reference is used. Internal clock is used instead.
SBG_ECOM_TIME_REF_SYNC_IN_A	1	The system will be synchronized on the clock input observed at SYNC_IN_A pin.
SBG_ECOM_TIME_REF_UTC_GPS_1	2	The system will be synchronized to a GNSS PPS signal, (see GPS module assignment).

Table 29: SbgEComTimeReferenceSrc enum

Note: The new configuration will only be applied after SBG_ECOM_CMD_SETTINGS_ACTION (01) command is issued, with SBG_ECOM_SAVE_SETTINGS parameter.

Note: Note the clock input must also be configured in the Event Input configuration to enable detection on correct edges (rising, falling or both edges). See SBG_ECOM_CMD_SYNC_IN_CONF (26) for more information.

2.2.18.2. SBG_ECOM_CMD_VALIDITY_THRESHOLDS (38)

Sent without any payload, this command will ask the device for the different validity thresholds currently used. These thresholds are used in the SOLUTION_STATUS definition. They are useful to indicate whether the position, velocity, attitude and heading precision is within user acceptable limits or not. Sent with full payload, the command will configure user values for these thresholds.

Passing 0.0 threshold as parameter will leave corresponding parameter to default settings. This allows setting only part of these thresholds.

The full payload message is defined below:

Field	Description	Unit	Format	Size	Offset
positionThreshold	Position estimated accuracy threshold (3D standard deviation)	m	float	4	0
velocityThreshold	Velocity estimated accuracy threshold (3D standard deviation)	m/s	float	4	4
attitudeThreshold	Roll / Pitch estimated accuracy threshold (max of roll or pitch standard deviations)	rad	float	4	8
headingThreshold	Heading standard deviation threshold	rad	float	4	12
				Total size	16

2.2.19. Licenses and features

2.2.19.1. SBG_ECOM_CMD_FEATURES (33)

Sent without any payload, this command will ask the device for the different features available such as navigation capability and internal GNSS receiver options.

The sensor will answer with the full payload version as defined below:

Field	Description	Unit	Format	Size	Offset
sensorFeaturesMask	The different measurement capabilities of this unit - Table 30	-	bitmask	4	0
gnssType	The type of GNSS receiver used (brand and model) - Table 31	-	enum	1	4
gnssUpdateRate	The actual GNSS update rate	Hz	uint8	1	5
gnssSignalsMask	GNSS receiver signals tracking - Table 32	-	uint32	4	6
gnssFeaturesMask	GNSS receiver computation and output features - Table 33	-	uint32	4	10
gnssProductCode	String containing the GNSS receiver product code ("\0" if unknown)	-	string	32	14
gnssSerialNumber	String containing the GNSS receiver serial number ("\0" if unknown)	-	string	32	46
gnssFirmwareVersion	String containing the GNSS firmware version ("\0" if unknown)	-	string	32	78
			Total size		110

Enum definitions

Name	Value	Description
SBG_ECOM_SENSOR_FEATURE_IMU	0x00000001	This unit can provide IMU data
SBG_ECOM_SENSOR_FEATURE_AHRS	0x00000002	This unit can provide orientation data
SBG_ECOM_SENSOR_FEATURE_NAVIGATION	0x00000004	This unit can provide position and velocity data
SBG_ECOM_SENSOR_FEATURE_SHIP_MOTION	0x00000008	This unit can provide ship motion data output (heave)

Table 30: SbgEComSensorFeaturesMask mask

Name	Value	Description
SBG_ECOM_GNSS_TYPE_DISABLED	0	GNSS module disabled
SBG_ECOM_GNSS_TYPE_EXTERNAL	1	External GNSS module (all features are unknown)
SBG_ECOM_GNSS_TYPE_UBX_MAX_M8	2	Ublox MAX-M8 module
SBG_ECOM_GNSS_TYPE_NOV_OEM615	3	Novatel OEM615 device
SBG_ECOM_GNSS_TYPE_NOV_OEM615_DUAL	4	Two Novatel OEM615 devices for dual antenna
SBG_ECOM_GNSS_TYPE_NOV_OEM617D	5	Novatel OEM617D device
SBG_ECOM_GNSS_TYPE_SEP_AX4	6	Septentrio AsteRx4
SBG_ECOM_GNSS_TYPE_SEP_AXM2A	7	Septentrio AsteRx-m2a
SBG_ECOM_GNSS_TYPE_UBX_F9P	8	Ublox ZED-F9P high precision GNSS

Table 31: SbgEComGnssType enum

Name	Value	Description
SBG_ECOM_GNSS_SIGNAL_GPS_L1	0x00000001	This GNSS receiver tracks GPS L1 band.
SBG_ECOM_GNSS_SIGNAL_GPS_L2	0x00000002	This GNSS receiver tracks GPS L2 band.
SBG_ECOM_GNSS_SIGNAL_GPS_L5	0x00000004	This GNSS receiver tracks GPS L5 band.

Name	Value	Description
SBG_ECOM_GNSS_SIGNAL_GLONASS_L1	0x00000008	This GNSS receiver tracks GLONASS L1 band.
SBG_ECOM_GNSS_SIGNAL_GLONASS_L2	0x00000010	This GNSS receiver tracks GLONASS L2 band.
SBG_ECOM_GNSS_SIGNAL_BEIDOU_B1	0x00000020	This GNSS receiver tracks BEIDOU B1 band.
SBG_ECOM_GNSS_SIGNAL_BEIDOU_B2	0x00000040	This GNSS receiver tracks BEIDOU B2 band.
SBG_ECOM_GNSS_SIGNAL_BEIDOU_B3	0x00000080	This GNSS receiver tracks BEIDOU B3 band.
SBG_ECOM_GNSS_SIGNAL_GALILEO_E1	0x00000100	This GNSS receiver tracks GALILEO E1 band.
SBG_ECOM_GNSS_SIGNAL_GALILEO_E5	0x00000200	This GNSS receiver tracks GALILEO E5 band.
SBG_ECOM_GNSS_SIGNAL_GALILEO_E6	0x00000400	This GNSS receiver tracks GALILEO E6 band.
SBG_ECOM_GNSS_SIGNAL_QZSS	0x00000800	This GNSS receiver tracks QZSS signals
SBG_ECOM_GNSS_SIGNAL_SBAS	0x00001000	This GNSS receiver tracks SBAS signals
SBG_ECOM_GNSS_SIGNAL_L_BAND	0x00002000	This GNSS receiver tracks L-Band (for PPP services)

Table 32: SbgEComGnssSignalsMask bitmask

Name	Value	Description
SBG_ECOM_GNSS_FEATURE_DUAL_ANT	0x00000001	This GNSS receiver provides a dual antenna heading
SBG_ECOM_GNSS_FEATURE_RTK_LIMITED	0x00000002	This GNSS receiver has limited RTK accuracy (Trimble RTK 30/30)
SBG_ECOM_GNSS_FEATURE_RTK	0x00000004	This GNSS receiver can provide full RTK accuracy
SBG_ECOM_GNSS_FEATURE_PPP	0x00000008	This GNSS receiver can provide PPP computations
SBG_ECOM_GNSS_FEATURE_RAW_DATA	0x00000010	This GNSS receiver provides RAW data output
SBG_ECOM_GNSS_FEATURE_RAIM	0x00000020	This GNSS provides Receiver Autonomous Integrity Monitoring
SBG_ECOM_GNSS_FEATURE_HIGH_SPEED	0x00000040	This GNSS receiver has no high speed limitation (> 515m/s)

Table 33: SbgEComGnssFeaturesMask bitmask

2.2.19.2. SBG_ECOM_CMD_LICENSE_APPLY (34)

This command is used to upload and apply a software license provided as a file by SBG Systems.

This command uses the transfer sub-protocol since the set of settings could be too large to be sent in a unique sbgECom protocol frame. See 2.2.2 Large buffers transmission (Transfer sub-protocol) and more specifically 2.2.2.1 Sending data to the device for further information.

Field	Description	Unit	Format	Size	Offset
transferCmd	Transfer protocol CMD	-	enum	2	0
transferPayload	Transfer protocol payload	-	-	0:n	1
			Total size		2+n

Note: This command will apply the configuration, save it to the non-volatile memory, and initiate a device reboot immediately after sending the ACK.

2.3. SBG_ECOM_CLASS_LOG_ECOM_0

This message class contains only input/output logs. These messages contain a fixed set of data. Each log stores coherent data that can be fully interpreted without any additional information.

In addition, a time stamp expressed in microseconds is included in each log. It can be used to synchronize the data precisely.

Finally, the sbgECom binary logs have been designed to ease post processing operations by including a status field to know how to interpret some specific logs.

Note: SBG Systems reserves the right to add at the end of logs new fields in future revision of the sbgECom protocol for upward compatibility. Therefore, user must consider the DATA sizes defined in this document as a minimum size.

2.3.1. Binary Output Logs Overview

The following list, provides a quick overview of all available logs for this message class. It briefly describle which parameters are contained in each output log.

Name (Msg ID)	Description
SBG_ECOM_LOG_STATUS (01)	Status general, clock, com aiding, solution, heave
SBG_ECOM_LOG_UTC_TIME (02)	Provides UTC time reference
SBG_ECOM_LOG_IMU_DATA (03)	Includes IMU status, acc., gyro, temp delta speeds and delta angles values
SBG_ECOM_LOG_IMU_SHORT (44)	Asynchronous delta angles and delta velocities values from IMU directly
SBG_ECOM_LOG_EKF_EULER (06)	Includes roll, pitch, yaw and their accuracies on each axis
SBG_ECOM_LOG_EKF_QUAT (07)	Includes the 4 quaternions values
SBG_ECOM_LOG_EKF_NAV (08)	Position and velocities in NED coordinates with the accuracies on each axis
SBG_ECOM_LOG_SHIP_MOTION (09)	Real time heave, surge, sway, accelerations and velocity
SBG_ECOM_LOG_SHIP_MOTION_HP (32)	Delayed heave, surge, sway, accelerations and velocity
SBG_ECOM_LOG_MAG (04)	Magnetic data with associated accelerometer on each axis
SBG_ECOM_LOG_MAG_CALIB (05)	Magnetometer calibration data (raw buffer)
SBG_ECOM_LOG_GPS1_VEL (13)	GNSS velocity from primary receiver
SBG_ECOM_LOG_GPS1_POS (14)	GNSS position from primary receiver
SBG_ECOM_LOG_GPS1_HDT (15)	GNSS true heading from primary receiver
SBG_ECOM_LOG_GPS1_RAW (31)	GNSS raw data from primary receiver
SBG_ECOM_LOG_GPS2_VEL (16)	GNSS velocity from secondary receiver
SBG_ECOM_LOG_GPS2_POS (17)	GNSS position from secondary receiver
SBG_ECOM_LOG_GPS2_HDT (18)	GNSS true heading from secondary receiver
SBG_ECOM_LOG_GPS2_RAW (38)	GNSS raw data from secondary receiver
SBG_ECOM_LOG_ODO_VEL (19)	Provides odometer velocity measured by the device
SBG_ECOM_LOG_AIR_DATA (36)	Barometric altimeter input/output for airdata support.
SBG_ECOM_LOG_DEPTH (47)	Depth sensor measurement log used for subsea navigation.
SBG_ECOM_LOG_DVL_BOTTOM_TRACK (29)	Doppler Velocity Log for bottom tracking data

Name (Msg ID)	Description
SBG_ECOM_LOG_DVL_WATER_TRACK (30)	Doppler Velocity log for water layer data
SBG_ECOM_LOG_USBL (37)	Raw USBL position data for subsea navigation
SBG_ECOM_LOG_EVENT_A (24)	Event marker sent when a signal is detected on Sync In A pin
SBG_ECOM_LOG_EVENT_B (25)	Event marker sent when a signal is detected on Sync In B pin
SBG_ECOM_LOG_EVENT_C (26)	Event marker sent when a signal is detected on Sync In C pin
SBG_ECOM_LOG_EVENT_D (27)	Event marker sent when a signal is detected on Sync In D pin
SBG_ECOM_LOG_EVENT_E (28)	Event marker sent when a signal is detected on Sync In E pin
SBG_ECOM_LOG_EVENT_OUT_A (45)	Event marker used to time stamp each generated Sync Out A signal.
SBG_ECOM_LOG_EVENT_OUT_B (46)	Event marker used to time stamp each generated Sync Out B signal.

2.3.2. Binary Output Logs Availability

The table below details the sbgECom binary logs availability according to a specific product family and product type. For example, if you have an ELLIPSE-E-G4A2-B1, you should refer to the column 'E' in the main column 'ELLIPSE'. Each available message is indicated with the ● symbol.

	ELLIPSE					High Performance INS					
Name (Message ID)	I	A	E	N	D	A	E	N	D	M	U
SBG_ECOM_LOG_STATUS (01)	●	●	●	●	●	●	●	●	●	●	●
SBG_ECOM_LOG_UTC_TIME (02)	●	●	●	●	●	●	●	●	●	●	●
SBG_ECOM_LOG_IMU_DATA (03)	●	●	●	●	●	●	●	●	●	●	●
SBG_ECOM_LOG_IMU_SHORT (44)	●	●	●	●	●						
SBG_ECOM_LOG_EKF_EULER (06)		●	●	●	●	●	●	●	●	●	●
SBG_ECOM_LOG_EKF_QUAT (07)		●	●	●	●	●	●	●	●	●	●
SBG_ECOM_LOG_EKF_NAV (08)			●	●	●		●	●	●		●
SBG_ECOM_LOG_SHIP_MOTION (09)		●	●	●	●	●	●	●	●	●	●
SBG_ECOM_LOG_SHIP_MOTION_HP (32)						●	●	●	●	●	●
SBG_ECOM_LOG_MAG (04)	●	●	●	●	●						
SBG_ECOM_LOG_MAG_CALIB (05)	●	●	●	●	●						
SBG_ECOM_LOG_GPS1_VEL (13)			●	●	●	●	●	●	●	●	●
SBG_ECOM_LOG_GPS1_POS (14)			●	●	●	●	●	●	●	●	●
SBG_ECOM_LOG_GPS1_HDT (15)			●		●	●	●	●	●	●	●
SBG_ECOM_LOG_GPS1_RAW (31)			●		●	●	●	●	●	●	●
SBG_ECOM_LOG_GPS2_VEL (16)							●	●	●		●
SBG_ECOM_LOG_GPS2_POS (17)							●	●	●		●
SBG_ECOM_LOG_GPS2_HDT (18)							●	●	●		●
SBG_ECOM_LOG_GPS2_RAW (38)							●	●	●		●
SBG_ECOM_LOG_ODO_VEL (19)			●	●	●		●	●	●		
SBG_ECOM_LOG_AIR_DATA (36)			●	●	●						
SBG_ECOM_LOG_DEPTH (47)							●	●	●		●
SBG_ECOM_LOG_DVL_BOTTOM_TRACK (29)			●	●	●		●	●	●		●

Name (Message ID)	ELLIPSE					High Performance INS					
	I	A	E	N	D	A	E	N	D	M	U
SBG_ECOM_LOG_DVL_WATER_TRACK (30)			•	•	•		•	•	•		•
SBG_ECOM_LOG_USBL (37)							•	•	•		•
SBG_ECOM_LOG_EVENT_A (24)	•	•	•	•	•	•	•	•	•	•	•
SBG_ECOM_LOG_EVENT_B (25)		•	•	•	•	•	•	•	•	•	•
SBG_ECOM_LOG_EVENT_C (26)				•		•	•	•	•		
SBG_ECOM_LOG_EVENT_D (27)				•	•	•	•	•	•	•	•
SBG_ECOM_LOG_EVENT_E (28)						•	•	•	•	•	•
SBG_ECOM_LOG_EVENT_OUT_A (45)											
SBG_ECOM_LOG_EVENT_OUT_B (46)											

Note: Please also check the Hardware Reference Manual of the product to confirm log availability. For example, ELLIPSE Micro don't support SBG_ECOM_LOG_EVENT_B message.

2.3.3. General information and time

2.3.3.1. Device status

This output combines all system status data, divided into several categories: General, Communications, Aiding..

This log is useful for advanced status information.

Message name (ID)	SBG_ECOM_LOG_STATUS (01)				
Field	Description	Unit	Format	Size	Offset
TIME STAMP	Time since sensor is powered up	µs	uint32	4	0
GENERAL STATUS	General status bitmask and enums	-	uint16	2	4
RESERVED 1	Reserved status field for future use	-	uint16	2	6
COM STATUS	Communication status bitmask and enums.	-	uint32	4	8
AIDING STATUS	Aiding equipment status bitmask and enums.	-	uint32	4	12
RESERVED 2	Reserved status field for future use	-	uint32	4	16
RESERVED 3	Reserved field for future use	-	uint16	2	20
UP TIME	System up time since the power on.	s	uint32	4	22
				Total size	26

GENERAL_STATUS definition

Provides general device status and information such as the power supplies (main, IMU, GNSS), settings, temperature and data-logger.

Bit	Name	Type	Description
0	SBG_ECOM_GENERAL_MAIN_POWER_OK	Mask	Set to 1 when main power supply is OK.
1	SBG_ECOM_GENERAL_IMU_POWER_OK	Mask	Set to 1 when IMU power supply is OK.
2	SBG_ECOM_GENERAL_GPS_POWER_OK	Mask	Set to 1 when GPS power supply is OK.
3	SBG_ECOM_GENERAL_SETTINGS_OK	Mask	Set to 1 if settings were correctly loaded
4	SBG_ECOM_GENERAL_TEMPERATURE_OK	Mask	Set to 1 when temperature is within specified limits.
5	SBG_ECOM_GENERAL_DATALOGGER_OK	Mask	Set to 1 when the data-logger is working correctly.
6	SBG_ECOM_GENERAL_CPU_OK	Mask	Set to 1 if the CPU headroom is correct.

COM_STATUS definition

Provide information on ports, tells is they are valid or saturated.

Bit	Name	Type	Description
0	SBG_ECOM_PORTA_VALID	Mask	Set to 0 in case of low level communication error.
1	SBG_ECOM_PORTB_VALID	Mask	Set to 0 in case of low level communication error.
2	SBG_ECOM_PORTC_VALID	Mask	Set to 0 in case of low level communication error.
3	SBG_ECOM_PORTD_VALID	Mask	Set to 0 in case of low level communication error.
4	SBG_ECOM_PORTE_VALID	Mask	Set to 0 in case of low level communication error.
5	SBG_ECOM_PORTA_RX_OK	Mask	Set to 0 in case of saturation on PORT A input

Bit	Name	Type	Description
6	SBG_ECOM_PORTA_TX_OK	Mask	Set to 0 in case of saturation on PORT A output
7	SBG_ECOM_PORTB_RX_OK	Mask	Set to 0 in case of saturation on PORT B input
8	SBG_ECOM_PORTB_TX_OK	Mask	Set to 0 in case of saturation on PORT B output
9	SBG_ECOM_PORTC_RX_OK	Mask	Set to 0 in case of saturation on PORT C input
10	SBG_ECOM_PORTC_TX_OK	Mask	Set to 0 in case of saturation on PORT C output
11	SBG_ECOM_PORTD_RX_OK	Mask	Set to 0 in case of saturation on PORT D input
12	SBG_ECOM_PORTD_TX_OK	Mask	Set to 0 in case of saturation on PORT D output
13	SBG_ECOM_PORTE_RX_OK	Mask	Set to 0 in case of saturation on PORT E input
14	SBG_ECOM_PORTE_TX_OK	Mask	Set to 0 in case of saturation on PORT E output
15	SBG_ECOM_ETH0_VALID	Mask	Set to 0 in case of saturation on PORT ETH0
16	SBG_ECOM_ETH1_VALID	Mask	Set to 0 in case of saturation on PORT ETH1
17	SBG_ECOM_ETH2_VALID	Mask	Set to 0 in case of saturation on PORT ETH2
18	SBG_ECOM_ETH3_VALID	Mask	Set to 0 in case of saturation on PORT ETH3
19	SBG_ECOM_ETH4_VALID	Mask	Set to 0 in case of saturation on PORT ETH4
25	SBG_ECOM_CAN_VALID	Mask	Set to 0 in case of low level communication error.
26	SBG_ECOM_CAN_RX_OK	Mask	Set to 0 in case of saturation on CAN Bus input buffer
27	SBG_ECOM_CAN_TX_OK	Mask	Set to 0 in case of saturation on CAN Bus output buffer
28-30	SBG_ECOM_CAN_BUS	Enum	Define the CAN Bus status (see the table Table 5).

You can find below the values that each clock enumeration can have:

Value	Name	Description
0	SBG_ECOM_CAN_BUS_OFF	Bus OFF operation due to too much errors.
1	SBG_ECOM_CAN_BUS_TX_RX_ERR	Transmit or received error.
2	SBG_ECOM_CAN_BUS_OK	The CAN bus is working correctly.
3	SBG_ECOM_CAN_BUS_ERROR	A general error has occurred on the CAN bus.

Table 34: CAN Bus status enumeration

AIDING_STATUS definition

Tells which aiding data is received.

Bit	Name	Type	Description
0	SBG_ECOM_AIDING_GPS1_POS_RECV	Mask	Set to 1 when valid GPS 1 position data is received
1	SBG_ECOM_AIDING_GPS1_VEL_RECV	Mask	Set to 1 when valid GPS 1 velocity data is received
2	SBG_ECOM_AIDING_GPS1_HDT_RECV	Mask	Set to 1 when valid GPS 1 true heading data is received
3	SBG_ECOM_AIDING_GPS1_UTC_RECV	Mask	Set to 1 when valid GPS 1 UTC time data is received
4	SBG_ECOM_AIDING_GPS2_POS_RECV	Mask	Set to 1 when valid GPS 2 position data is received
5	SBG_ECOM_AIDING_GPS2_VEL_RECV	Mask	Set to 1 when valid GPS 2 velocity data is received
6	SBG_ECOM_AIDING_GPS2_HDT_RECV	Mask	Set to 1 when valid GPS 2 true heading data is received
7	SBG_ECOM_AIDING_GPS2_UTC_RECV	Mask	Set to 1 when valid GPS 2 UTC time data is received
8	SBG_ECOM_AIDING_MAG_RECV	Mask	Set to 1 when valid Magnetometer data is received
9	SBG_ECOM_AIDING_ODO_RECV	Mask	Set to 1 when Odometer pulse is received
10	SBG_ECOM_AIDING_DVL_RECV	Mask	Set to 1 when valid DVL data is received
11	SBG_ECOM_AIDING_USBL_RECV	Mask	Set to 1 when valid USBL data is received
12	SBG_ECOM_AIDING_DEPTH_RECV	Mask	Set to 1 when valid Depth sensor data is received
13	SBG_ECOM_AIDING_AIR_DATA_RECV	Mask	Set to 1 when valid altitude and/or airspeed is received

2.3.3.2. UTC and GPS Time

Provides UTC time reference. This frame also provides a time correspondence between the device TIME_STAMP value and the actual UTC Time. You thus have to use this frame if you would like to time stamp all data to an absolute UTC or GPS time reference.

Message name (ID)	`SBG_ECOM_LOG_UTC_TIME (02)`				
Field	Description	Unit	Format	Size	Offset
TIME_STAMP	Time since sensor is powered up	µs	uint32	4	0
CLOCK_STATUS	General UTC time and clock sync status	-	uint16	2	4
YEAR	Year	year	uint16	2	6
MONTH	Month in Year [1 ... 12]	month	uint8	1	8
DAY	Day in Month [1 ... 31]	d	uint8	1	9
HOUR	Hour in day [0 ... 23]	h	uint8	1	10
MIN	Minute in hour [0 ... 59]	min	uint8	1	11
SEC	Second in minute [0 ... 60] Note 60 is when a leap second is added.	s	uint8	1	12
NANOSEC	Nanosecond of second.	ns	uint32	4	13
GPS_TOW	GPS Time of week	ms	uint32	4	17
				Total size	21

CLOCK_STATUS definition

Provide status on the clock stability, error and synchronization.

Bit	Name	Type	Description
0	SBG_ECOM_CLOCK_STABLE_INPUT	Mask	Set to 1 when a clock input can be used to synchronize the internal clock.
[1-4]	SBG_ECOM_CLOCK_STATUS	Enum	Define the internal clock estimation status (see the Table 6: SbgEComModuleSyncAssignment enum above).
5	SBG_ECOM_CLOCK_UTC_SYNC	Mask	Set to 1 if UTC time is synchronized with a PPS
[6-9]	SBG_ECOM_CLOCK_UTC_STATUS	Enum	Define the UTC validity status (see the Table 7: SbgEComOdometerPinAssignment enum above).

You can find below the values that each clock enumeration can have:

Value	Name	Description
0	SBG_ECOM_CLOCK_ERROR	An error has occurred on the clock estimation.
1	SBG_ECOM_CLOCK_FREE_RUNNING	The clock is only based on the internal crystal.
2	SBG_ECOM_CLOCK_STEERING	A PPS has been detected and the clock is converging to it.
3	SBG_ECOM_CLOCK_VALID	The clock has converged to the PPS and is within 500ns.

Table 35: Clock Status enumeration

Value	Name	Description
0	SBG_ECOM_UTC_INVALID	The UTC time is not known, we are just propagating the UTC time internally.
1	SBG_ECOM_UTC_NO_LEAP_SEC	We have received valid UTC time information but we don't have the leap seconds information.
2	SBG_ECOM_UTC_VALID	We have received valid UTC time data with valid leap seconds.

Table 36: UTC time status enumeration

2.3.4. Inertial Sensor Data

2.3.4.1. Time filtered IMU data

Provides accelerometers, gyros, delta angles and delta velocities data directly from the IMU.

Message name (ID)	SBG_ECOM_LOG_IMU_DATA (03)				
Field	Description	Unit	Format	Size	Offset
TIME_STAMP	Time since sensor is powered up	µs	uint32	4	0
IMU_STATUS	IMU Status bitmask	-	uint16	2	4
ACCEL_X	Filtered Accelerometer – X axis	m/s^2	float	4	6
ACCEL_Y	Filtered Accelerometer – Y axis	m/s^2	float	4	10
ACCEL_Z	Filtered Accelerometer – Z axis	m/s^2	float	4	14
GYRO_X	Filtered Gyroscope – X axis	rad/s	float	4	18
GYRO_Y	Filtered Gyroscope – Y axis	rad/s	float	4	22
GYRO_Z	Filtered Gyroscope – Z axis	rad/s	float	4	26
TEMP	Internal Temperature	°C	float	4	30
DELTA_VEL_X	Sculling output – X axis	m/s^2	float	4	34
DELTA_VEL_Y	Sculling output – Y axis	m/s^2	float	4	38
DELTA_VEL_Z	Sculling output – Z axis	m/s^2	float	4	42
DELTA_ANGLE_X	Coning output – X axis	rad/s	float	4	46
DELTA_ANGLE_Y	Coning output – Y axis	rad/s	float	4	50
DELTA_ANGLE_Z	Coning output – Z axis	rad/s	float	4	54
				Total size	58

IMU_STATUS definition

Status used to know if sensors are working correctly and are in their measurement range.

Bit	Name	Description
0 (LSB)	SBG_ECOM_IMU_COM_OK	Set to 1 if the communication with the IMU is ok.
1	SBG_ECOM_IMU_STATUS_BIT	Set to 1 if internal IMU passes Built In Test (Calibration, CPU)
2	SBG_ECOM_IMU_ACCEL_X_BIT	Set to 1 if accelerometer X passes Built In Test
3	SBG_ECOM_IMU_ACCEL_Y_BIT	Set to 1 if accelerometer Y passes Built In Test
4	SBG_ECOM_IMU_ACCEL_Z_BIT	Set to 1 if accelerometer Z passes Built In Test
5	SBG_ECOM_IMU_GYRO_X_BIT	Set to 1 if gyroscope X passes Built In Test
6	SBG_ECOM_IMU_GYRO_Y_BIT	Set to 1 if gyroscope Y passes Built In Test
7	SBG_ECOM_IMU_GYRO_Z_BIT	Set to 1 if gyroscope Z passes Built In Test
8	SBG_ECOM_IMU_ACCELS_IN_RANGE	Set to 1 if accelerometers are within operating range
9	SBG_ECOM_IMU_GYROS_IN_RANGE	Set to 1 if gyroscopes are within operating range

2.3.4.2. IMU Short data

Compact and asynchronous log that returns delta angle and delta velocities from the IMU. This log is recommended for post processing as it ensure best accuracy and smallest size.

Message name (ID)	SBG_ECOM_LOG_IMU_SHORT (44)				
Field	Description	Unit	Format	Size	Offset
TIME_STAMP	Time since sensor is powered up	µs	uint32	4	0
IMU_STATUS	IMU Status bitmask	-	uint16	2	4
DELTA_VEL_X	X axis delta velocity – Scaling 1048576 LSB for 1 $m.s^{-2}$	$m.s^{-2}$	int32	4	6
DELTA_VEL_Y	Y axis delta velocity – Scaling 1048576 LSB for 1 $m.s^{-2}$	$m.s^{-2}$	int32	4	10
DELTA_VEL_Z	Z axis delta velocity – Scaling 1048576 LSB for 1 $m.s^{-2}$	$m.s^{-2}$	int32	4	14
DELTA_ANGLE_X	X axis delta angle – Scaling 67108864 LSB for 1 $rad.s^{-1}$	$rad.s^{-1}$	int32	4	18
DELTA_ANGLE_Y	Y axis delta angle – Scaling 67108864 LSB for 1 $rad.s^{-1}$	$rad.s^{-1}$	int32	4	22
DELTA_ANGLE_Z	Z axis delta angle – Scaling 67108864 LSB for 1 $rad.s^{-1}$	$rad.s^{-1}$	int32	4	26
TEMP	IMU Temperature – Scaling 256 LSB for 1°C	°C	int16	2	28
				Total size	32

IMU_STATUS definition

Status used to know if sensors are working correctly and are in their measurement range.

Bit	Name	Description
0 (LSB)	SBG_ECOM_IMU_COM_OK	Set to 1 if the communication with the IMU is ok.
1	SBG_ECOM_IMU_STATUS_BIT	Set to 1 if internal IMU passes Built In Test (Calibration, CPU)
2	SBG_ECOM_IMU_ACCEL_X_BIT	Set to 1 if accelerometer X passes Built In Test
3	SBG_ECOM_IMU_ACCEL_Y_BIT	Set to 1 if accelerometer Y passes Built In Test
4	SBG_ECOM_IMU_ACCEL_Z_BIT	Set to 1 if accelerometer Z passes Built In Test
5	SBG_ECOM_IMU_GYRO_X_BIT	Set to 1 if gyroscope X passes Built In Test
6	SBG_ECOM_IMU_GYRO_Y_BIT	Set to 1 if gyroscope Y passes Built In Test
7	SBG_ECOM_IMU_GYRO_Z_BIT	Set to 1 if gyroscope Z passes Built In Test
8	SBG_ECOM_IMU_ACCELS_IN_RANGE	Set to 1 if accelerometers are within operating range
9	SBG_ECOM_IMU_GYROS_IN_RANGE	Set to 1 if gyroscopes are within operating range

2.3.5. EKF output logs

The following logs provide the navigation unit output, in terms of Euler angles, quaternion, velocity, position and heave. They share a common status field described below:

SOLUTION_STATUS definition

Provide information on the internal Kalman filter status such as which aiding data is used to compute the solution and the provided solution mode.

Bit	Name	Description
[0-3]	SBG_ECOM_SOLUTION_MODE	Defines the Kalman filter computation mode (see the table 37 below)
4	SBG_ECOM_SOL_ATTITUDE_VALID	Set to 1 if Attitude data is reliable (Roll/Pitch error < 0,5°)
5	SBG_ECOM_SOL_HEADING_VALID	Set to 1 if Heading data is reliable (Heading error < 1°)
6	SBG_ECOM_SOL_VELOCITY_VALID	Set to 1 if Velocity data is reliable (velocity error < 1.5 m/s)
7	SBG_ECOM_SOL_POSITION_VALID	Set to 1 if Position data is reliable (Position error < 10m)
8	SBG_ECOM_SOL_VERT_REF_USED	Set to 1 if vertical reference is used in solution
9	SBG_ECOM_SOL_MAG_REF_USED	Set to 1 if magnetometer is used in solution
10	SBG_ECOM_SOL_GPS1_VEL_USED	Set to 1 if GPS velocity is used in solution
11	SBG_ECOM_SOL_GPS1_POS_USED	Set to 1 if GPS Position is used in solution
13	SBG_ECOM_SOL_GPS1_HDT_USED	Set to 1 if GPS True Heading is used in solution
14	SBG_ECOM_SOL_GPS2_VEL_USED	Set to 1 if GPS2 velocity is used in solution
15	SBG_ECOM_SOL_GPS2_POS_USED	Set to 1 if GPS2 Position is used in solution
17	SBG_ECOM_SOL_GPS2_HDT_USED	Set to 1 if GPS2 True Heading is used in solution
18	SBG_ECOM_SOL_ODO_USED	Set to 1 if Odometer is used in solution
19	SBG_ECOM_SOL_DVL_BT_USED	Set to 1 if DVL Bottom Tracking is used in solution
20	SBG_ECOM_SOL_DVL_WT_USED	Set to 1 if DVL Water Layer is used in solution
24	SBG_ECOM_SOL_USBL_USED	Set to 1 if USBL / LBL is used in solution.
25	SBG_ECOM_SOL_AIR_DATA_USED	Set to 1 if an altitude or true airspeed is used in solution
26	SBG_ECOM_SOL_ZUPT_USED	Set to 1 if a ZUPT is used in solution
27	SBG_ECOM_SOL_ALIGN_VALID	Set to 1 if sensor alignment and calibration parameters are valid
28	SBG_ECOM_SOL_DEPTH_USED	Set to 1 if Depth sensor (for subsea navigation) is used in solution

Note: All "used in solution" flags have a timeout mechanism to ease their interpretation. For example, the flag SBG_ECOM_SOL_ODO_USED will remain valid during 3s following the latest time the odometer has been acctualy used in the solution.

You can find below the different available solution modes:

Value	Name	Description
0	SBG_ECOM_SOL_MODE_UNINITIALIZED	The Kalman filter is not initialized and the returned data are all invalid.
1	SBG_ECOM_SOL_MODE_VERTICAL_GYRO	The Kalman filter only rely on a vertical reference to compute roll and pitch angles. Heading and navigation data drift freely.
2	SBG_ECOM_SOL_MODE_AHRS	A heading reference is available, the Kalman filter provides full orientation but navigation data drift freely.
3	SBG_ECOM_SOL_MODE_NAV_VELOCITY	The Kalman filter computes orientation and velocity. Position is freely integrated from velocity estimation.
4	SBG_ECOM_SOL_MODE_NAV_POSITION	Nominal mode, the Kalman filter computes all parameters (attitude, velocity, position). Absolute position is provided.

Table 37: Solution modes enumeration

2.3.5.1. Euler angles

Provides computed orientation in Euler angles format.

Message name (ID)	**SBG_ECOM_LOG_EKF_EULER (06)**				
Field	**Description**	**Unit**	**Format**	**Size**	**Offset**
TIME_STAMP	Time since sensor is powered up	µs	uint32	4	0
ROLL	Roll angle	rad	float	4	4
PITCH	Pitch angle	rad	float	4	8
YAW	Yaw angle (heading)	rad	float	4	12
ROLL_ACC	1σ Roll angle accuracy	rad	float	4	16
PITCH_ACC	1σ Pitch angle accuracy	rad	float	4	20
YAW_ACC	1σ Yaw angle accuracy	rad	float	4	24
SOLUTION_STATUS	Global solution status. See SOLUTION_STATUS definition for more details.	-	uint32	4	28
				Total size	32

2.3.5.2. Quaternion attitude

Provides orientation in quaternion format.

Message name (ID)	SBG_ECOM_LOG_EKF_QUAT (07)				
Field	**Description**	**Unit**	**Format**	**Size**	**Offset**
TIME_STAMP	Time since sensor is powered up	µs	uint32	4	0
Q0	First quaternion parameter (W)	-	float	4	4
Q1	Second quaternion parameter (X)	-	float	4	8
Q2	Third quaternion parameter (Y)	-	float	4	12
Q3	Forth quaternion parameter (Z)	-	float	4	16
ROLL_ACC	1σ Roll angle accuracy	rad	float	4	20
PITCH_ACC	1σ Pitch angle accuracy	rad	float	4	24
YAW_ACC	1σ Yaw angle accuracy	rad	float	4	28
SOLUTION_STATUS	Global solution status. See SOLUTION_STATUS definition for more details.	-	uint32	4	32
				Total size	36

2.3.5.3. Navigation, position, velocity

Provides velocity in NED coordinate system and position (Latitude, Longitude, Altitude), and associated accuracy parameters.

Message name (ID)	SBG_ECOM_LOG_EKF_NAV (08)				
Field	**Description**	**Unit**	**Format**	**Size**	**Offset**
TIME_STAMP	Time since sensor is powered up	µs	uint32	4	0
VELOCITY_N	Velocity in North direction	m/s	float	4	4
VELOCITY_E	Velocity in East direction	m/s	float	4	8
VELOCITY_D	Velocity in Down direction	m/s	float	4	12
VELOCITY_N_ACC	1σ Velocity in North direction accuracy	m/s	float	4	16
VELOCITY_E_ACC	1σ Velocity in East direction accuracy	m/s	float	4	20
VELOCITY_D_ACC	1σ Velocity Down direction accuracy	m/s	float	4	24
LATITUDE	Latitude	°	double	8	28
LONGITUDE	Longitude	°	double	8	36
ALTITUDE	Altitude above Mean Sea Level	m	double	8	44
UNDULATION	Altitude difference between the geoid and the Ellipsoid. (WGS-84 Altitude = MSL Altitude + undulation)	m	float	4	52
LATITUDE_ACC	1σ Latitude accuracy	m	float	4	56
LONGITUDE_ACC	1σ Longitude accuracy	m	float	4	60
ALTITUDE_ACC	1σ Vertical Position accuracy	m	float	4	64
SOLUTION_STATUS	Global solution status. See SOLUTION_STATUS definition for more details.	-	uint32	4	68
				Total size	72

2.3.5.4. Heave, Surge, Sway

Provides ship motion data such as surge, sway, heave, velocity and accelerations.

This output may come from a real time computation mode, or from a delayed shipMotionHP computation. In this case, the time stamp will be the actual data time of validity instead of current time.

Ship motions HP only provide delayed heave measurements so surge and sway components are left to zero.

Message name (ID)	SBG_ECOM_LOG_SHIP_MOTION (09), SBG_ECOM_LOG_SHIP_MOTION_HP (32)				
Field	Description	Unit	Format	Size	Offset
TIME_STAMP	Time since sensor is powered up	µs	uint32	4	0
HEAVE_PERIOD	Main heave period in seconds.	s	float	4	4
SURGE	Surge at main location (positive forward)	m	float	4	8
SWAY	Sway at main location (positive right)	m	float	4	12
HEAVE	Heave at main location (positive down)	m	float	4	16
ACCEL_X	Longitudinal acceleration (positive forward)	$m.s^{-2}$	float	4	20
ACCEL_Y	Lateral acceleration (positive right)	$m.s^{-2}$	float	4	24
ACCEL_Z	Vertical acceleration (positive down)	$m.s^{-2}$	float	4	28
VEL_X	Longitudinal velocity (positive forward)	$m.s^{-1}$	float	4	32
VEL_Y	Lateral velocity (positive right)	$m.s^{-1}$	float	4	36
VEL_Z	Vertical velocity (positive down)	$m.s^{-1}$	float	4	40
HEAVE_STATUS	Ship motion output status	-	uint16	2	44
				Total size	46

HEAVE_STATUS definition

This field must be checked in order to know which fields are active in the output and to know if data is valid or not.

Bit	Name	Description
0	SBG_ECOM_HEAVE_VALID	Set to 1 after heave convergence time. Set to 0 in following conditions: • Turn occurred and no velocity aiding is available • Heave reached higher/lower limits • If a step is detected and filter has to re-converge • If internal failure
1	SBG_ECOM_HEAVE_VEL_AIDED	Set to 1 if heave output is compensated for transient accelerations
2	SBG_ECOM_HEAVE_SURGE_SWAY_INCLUDED	Set to 1 if surge and sway channels are provided in this output
3	SBG_ECOM_HEAVE_PERIOD_INCLUDED	Set to 1 if the swell period is provided in this output
4	SBG_ECOM_HEAVE_PERIOD_VALID	Set to 1 if the period returned is assumed to be valid or not.
5	SBG_ECOM_HEAVE_SWELL_MODE	Set to 1 if real time heave is using the swell mode computations

> **Note:** ELLIPSE products only support heave output so surge and sway related fields will be set to zero.

2.3.6. Aiding sensors outputs

2.3.6.1. Magnetometer

Provides magnetometer data and associated accelerometer. In case of internal magnetometer used, the internal accelerometer is also provided.

Message name (ID)	SBG_ECOM_LOG_MAG (04)				
Field	Description	Unit	Format	Size	Offset
TIME_STAMP	Time since sensor is powered up	µs	uint32	4	0
MAG_STATUS	Magnetometer status bitmask	-	uint16	2	4
MAG_X	Magnetometer output – X axis	a.u	float	4	6
MAG_Y	Magnetometer output – Y axis	a.u	float	4	10
MAG_Z	Magnetometer output – Z axis	a.u	float	4	14
ACCEL_X	Accelerometer output – X axis	m/s²	float	4	18
ACCEL_Y	Accelerometer output – Y axis	m/s²	float	4	22
ACCEL_Z	Accelerometer output – Z axis	m/s²	float	4	26
				Total size	30

MAG_STATUS definition

Bit	Name	Description
0(LSB)	SBG_ECOM_MAG_MAG_X_BIT	Set to 1 if the magnetometer X has passed the self test.
1	SBG_ECOM_MAG_MAG_Y_BIT	Set to 1 if the magnetometer Y has passed the self test.
2	SBG_ECOM_MAG_MAG_Z_BIT	Set to 1 if the magnetometer Z has passed the self test.
3	SBG_ECOM_MAG_ACCEL_X_BIT	Set to 1 if the accelerometer X has passed the self test.
4	SBG_ECOM_MAG_ACCEL_Y_BIT	Set to 1 if the accelerometer Y has passed the self test.
5	SBG_ECOM_MAG_ACCEL_Z_BIT	Set to 1 if the accelerometer Z has passed the self test.
6	SBG_ECOM_MAG_MAGS_IN_RANGE	Set to 1 if magnetometer is not saturated
7	SBG_ECOM_MAG_ACCELS_IN_RANGE	Set to 1 if accelerometer is not saturated
8	SBG_ECOM_MAG_CALIBRATION_OK	Set to 1 if magnetometer seems to be calibrated

2.3.6.2. Magnetometer calibration data

This log provides a RAW buffer for magnetic calibration procedure.

Message name (ID)	SBG_ECOM_LOG_MAG_CALIB (05)				
Field	Description	Unit	Format	Size	Offset
TIME_STAMP	Time since sensor is powered up	µs	uint32	4	0
RESERVED	Reserved field for future uses	-	uint16	2	4
BUFFER	Raw magnetic calibration buffer	-	16 bytes	16	6
				Total size	22

2.3.6.3. GNSS velocity

Provides velocity and course information from primary or secondary GNSS receiver. The time stamp is not aligned on main loop but instead of that, it dates the actual GNSS velocity data.

Message name (ID)	SBG_ECOM_LOG_GPS1_VEL (13), SBG_ECOM_LOG_GPS2_VEL (16)				
Field	Description	Unit	Format	Size	Offset
TIME_STAMP	Time since sensor is powered up	µs	uint32	4	0
GPS_VEL_STATUS	GPS velocity fix and status bitmask	-	uint32	4	4
GPS_TOW	GPS Time of Week	ms	uint32	4	8
VEL_N	Velocity in North direction	m/s	float	4	12
VEL_E	Velocity in East direction	m/s	float	4	16
VEL_D	Velocity in Down direction	m/s	float	4	20
VEL_ACC_N	1σ Accuracy in North direction	m/s	float	4	24
VEL_ACC_E	1σ Accuracy in East direction	m/s	float	4	28
VEL_ACC_D	1σ Accuracy in Down direction	m/s	float	4	32
COURSE	True direction of motion over ground (0 to 360°)	°	float	4	36
COURSE_ACC	1σ course accuracy (0 to 360°).	°	float	4	40
				Total size	44

GPS_VEL_STATUS definition

Bit	Type	Name	Description
[0-5]	Enum	SBG_ECOM_GPS_VEL_STATUS	The raw GPS velocity status (see the Table 38 below).
[6-11]	Enum	SBG_ECOM_GPS_VEL_TYPE	The raw GPS velocity type (see the Table 39 below).

You can find below the GPS velocity status and type enumerations:

Value	Name	Description
0	SBG_ECOM_VEL_SOL_COMPUTED	A valid solution has been computed.
1	SBG_ECOM_VEL_INSUFFICIENT_OBS	Not enough valid SV to compute a solution.
2	SBG_ECOM_VEL_INTERNAL_ERROR	An internal error has occurred.
3	SBG_ECOM_VEL_LIMIT	Velocity limit exceeded.

Table 38: Raw GPS velocity status enumeration

Value	Name	Description
0	SBG_ECOM_VEL_NO_SOLUTION	No valid velocity solution available.
1	SBG_ECOM_VEL_UNKNOWN_TYPE	An unknown solution type has been computed.
2	SBG_ECOM_VEL_DOPPLER	A Doppler velocity has been computed.
3	SBG_ECOM_VEL_DIFFERENTIAL	A velocity has been computed between two positions.

Table 39: Raw GPS velocity type enumeration

Note: Both the GPS velocity status and type should be tested to make sure that the output velocity is valid.

2.3.6.4. GNSS position

Provides position information from primary or secondary GNSS receiver.

The time stamp is not aligned on main loop but instead of that, it dates the actual GPS position data.

Message name (ID)	SBG_ECOM_LOG_GPS1_POS (14) / SBG_ECOM_LOG_GPS2_POS (17)				
Field	Description	Unit	Format	Size	Offset
TIME_STAMP	Time since sensor is powered up	µs	uint32	4	0
GPS_POS_STATUS	GPS position fix and status bitmask	-	uint32	4	4
GPS_TOW	GPS Time of Week	ms	uint32	4	8
LAT	Latitude, positive North	°	double	8	12
LONG	Longitude, positive East	°	double	8	20
ALT	Altitude Above Mean Sea Level	m	double	8	28
UNDULATION	Altitude difference between the geoid and the Ellipsoid (WGS-84 Altitude - MSL Altitude)	m	float	4	36
POS_ACC_LAT	1σ Latitude Accuracy	m	float	4	40
POS_ACC_LONG	1σ Longitude Accuracy	m	float	4	44
POS_ACC_ALT	1σ Altitude Accuracy	m	float	4	48
NUM_SV_USED	Number of space vehicles used in GNSS solution	-	uint8	1	52
BASE_STATION_ID	ID of the DGPS/RTK base station in use	-	uint16	2	54
DIFF_AGE	Differential data age	0.01 s	uint16	2	56
				Total size	57

GPS_POS_STATUS definition

Bit	Type	Name	Description
[0-5]	Enum	SBG_ECOM_GPS_POS_STATUS	The raw GPS position status (see the Table 40 below).
[6-11]	Enum	SBG_ECOM_GPS_POS_TYPE	The raw GPS position type (see the Table 41 below).
12	Mask	SBG_ECOM_GPS_POS_GPS_L1_USED	Set to 1 if GPS L1CA/L1P is used in the solution
13	Mask	SBG_ECOM_GPS_POS_GPS_L2_USED	Set to 1 if GPS L2P/L2C is used in the solution
14	Mask	SBG_ECOM_GPS_POS_GPS_L5_USED	Set to 1 if GPS L5 is used in the solution
15	Mask	SBG_ECOM_GPS_POS_GLO_L1_USED	Set to 1 if GLONASS L1CA is used in the solution
16	Mask	SBG_ECOM_GPS_POS_GLO_L2_USED	Set to 1 if GLONASS L2C/L2P is used in the solution
17	Mask	SBG_ECOM_GPS_POS_GLO_L3_USED	Set to 1 if GLONASS L3 is used in the solution
18	Mask	SBG_ECOM_GPS_POS_GAL_E1_USED	Set to 1 if Galileo E1 is used in solution
19	Mask	SBG_ECOM_GPS_POS_GAL_E5A_USED	Set to 1 if Galileo E5a is used in solution
20	Mask	SBG_ECOM_GPS_POS_GAL_E5B_USED	Set to 1 if Galileo E5b is used in solution
21	Mask	SBG_ECOM_GPS_POS_GAL_E5ALT_USED	Set to 1 if Galileo E5 AltBoc is used in solution
22	Mask	SBG_ECOM_GPS_POS_GAL_E6_USED	Set to 1 if Galileo E6 is used in solution
23	Mask	SBG_ECOM_GPS_POS_BDS_B1_USED	Set to 1 if BeiDou B1 is used in solution
24	Mask	SBG_ECOM_GPS_POS_BDS_B2_USED	Set to 1 if BeiDou B2 is used in solution
25	Mask	SBG_ECOM_GPS_POS_BDS_B3_USED	Set to 1 if BeiDou B3 is used in solution

Bit	Type	Name	Description
26	Mask	SBG_ECOM_GPS_POS_QZSS_L1_USED	Set to 1 if QZSS L1CA is used in solution
27	Mask	SBG_ECOM_GPS_POS_QZSS_L2_USED	Set to 1 if QZSS L2C is used in solution
28	Mask	SBG_ECOM_GPS_POS_QZSS_L3_USED	Set to 1 if QZSS L5 is used in solution

You can find below the GPS position status and type enumerations:

Value	Name	Description
0	SBG_ECOM_POS_SOL_COMPUTED	A valid solution has been computed.
1	SBG_ECOM_POS_INSUFFICIENT_OBS	Not enough valid SV to compute a solution.
2	SBG_ECOM_POS_INTERNAL_ERROR	An internal error has occurred.
3	SBG_ECOM_POS_HEIGHT_LIMIT	The height limit has been exceeded.

Table 40: Raw GPS position status enumeration

Value	Name	Description
0	SBG_ECOM_POS_NO_SOLUTION	No valid solution available.
1	SBG_ECOM_POS_UNKNOWN_TYPE	An unknown solution type has been computed.
2	SBG_ECOM_POS_SINGLE	Single point solution position.
3	SBG_ECOM_POS_PSRDIFF	Standard Pseudorange Differential Solution (DGPS).
4	SBG_ECOM_POS_SBAS	SBAS satellite used for differential corrections.
5	SBG_ECOM_POS_OMNISTAR	Omnistar VBS Position (L1 sub-meter).
6	SBG_ECOM_POS_RTK_FLOAT	Floating RTK ambiguity solution (20 cms RTK).
7	SBG_ECOM_POS_RTK_INT	Integer RTK ambiguity solution (2 cms RTK).
8	SBG_ECOM_POS_PPP_FLOAT	Precise Point Positioning with float ambiguities
9	SBG_ECOM_POS_PPP_INT	Precise Point Positioning with fixed ambiguities
10	SBG_ECOM_POS_FIXED	Fixed location solution position

Table 41: Raw GPS position type enumeration

Note: Both the GPS position status and type should be tested to make sure that the output position is valid.

2.3.6.5. GNSS True Heading

Provides true heading data from primary or secondary dual antenna GNSS receiver.

The time stamp is not aligned on main loop but instead of that, it dates the actual GPS true heading data.

Message name (ID)	SBG_ECOM_LOG_GPS1_HDT (15) / SBG_ECOM_LOG_GPS2_HDT (18)				
Field	Description	Unit	Format	Size	Offset
TIME_STAMP	Time since sensor is powered up	µs	uint32	4	0
GPS_HDT_STATUS	GPS True Heading status.	-	uint16	2	4
GPS_TOW	GPS Time of Week	ms	uint32	4	6
GPS_TRUE_HEADING	True heading angle (0 to 360°).	°	float	4	10
GPS_TRUE_HEADING_ACC	1σ True heading estimated accuracy (0 to 360°).	°	float	4	14
GPS_PITCH	Pitch angle from the master to the rover	°	float	4	18
GPS_PITCH_ACC	1σ pitch estimated accuracy	°	float	4	22
GPS_BASELINE	Distance between main and aux antenna.	m	float	4	26
				Total size	30

GPS_HDT_STATUS definition

Bit	Type	Name	Description
[0-5]	Enum	SBG_ECOM_GPS_HDT_STATUS	The raw GPS true heading status (see the Table 42 below).
6	Mask	SBG_ECOM_GPS_HDT_BASELINE_VALID	Set if the baseline length field is filled and valid.

You can find below the GPS true heading status enumeration:

Value	Name	Description
0	SBG_ECOM_HDT_SOL_COMPUTED	A valid solution has been computed.
1	SBG_ECOM_HDT_INSUFFICIENT_OBS	Not enough valid SV to compute a solution.
2	SBG_ECOM_HDT_INTERNAL_ERROR	An internal error has occurred.
3	SBG_ECOM_HDT_HEIGHT_LIMIT	The height limit has been exceeded.

Table 42: Raw GPS true heading status enumeration

2.3.6.6. GNSS raw data

This special log is used to store raw GPS data for post processing purposes. It stores directly untouched binary messages that are relevant for post processing from the GNSS receiver. Each message can store up to 4096 bytes of raw GNSS data.

Message name (ID)	SBG_ECOM_LOG_GPS1_RAW (31) / SBG_ECOM_LOG_GPS2_RAW (38)				
Field	Description	Unit	Format	Size	Offset
RAW_BUFFER	Buffer that stores GNSS raw data as returned by the receiver.	-	void	[0-4096]	0
				Total size	[0-4096]

2.3.6.7. Odometer velocity

Returns the external velocity aiding expressed in meters per second. This external velocity can come from any type of speed sensor that measures a velocity along the IMU X axis (direction of travel).

Typical external velocity aiding equipment could be a quadrature/pulse odometer or an odometer integrated in cars and retrieved through a CAN bus.

This asynchronous log is sent by the unit when a new velocity aiding information is available (for example if an odometer pulse is detected).

Timestamp considerations

When used as an output log, it is asynchronous and as such, time since reset is not aligned on main loop but instead of that, it dates the actual velocity data.

Message name (ID)	`SBG_ECOM_LOG_ODO_VEL (19)`				
Field	Description	Unit	Format	Size	Offset
TIME_STAMP	Time since sensor is powered up	µs	uint32	4	0
ODO_STATUS	Odometer velocity status bit-mask	-	uint16	2	4
ODO_VEL	Velocity in odometer direction	m/s	float	4	6
				Total size	10

ODO_VEL_STATUS definition

Bit	Name	Description
0 (LSB)	SBG_ECOM_ODO_REAL_MEAS	Set to 1 if this log comes from a real pulse measurement or 0 if it comes from a timeout.
1	SBG_ECOM_ODO_TIME_SYNC	Set to 1 if the velocity information is correctly time synchronized.

2.3.6.8. Airdata, altitude and true airspeed

The Airdata log provides both a barometric altitude above reference level as well as a true airspeed indication. The altitude is generally measured using a barometer whereas the true airspeed is measured using a pitot tube. The altitude is by default referenced to a standard 1013.25 hPa zero level pressure.

Unlike other sbgECom logs, this log is both used as an input and output log. It is send by the unit when a new internal airdata information is available but it can also be used to inject an external altitude and true airspeed aiding information..

Timestamp considerations

When used as an output log, it is asynchronous and as such, time since reset is not aligned on main loop but instead of that, it dates the actual altitude / airspeed measurement.

When used as an external altitude aiding input information, the TIME_STAMP field can either represent an absolute time information or just a measurement delay according to the status flags.

The measurement delay information will be used by the INS to compute an absolute timestamp based on the reception time, serial transmission delay (if applicable) and the specified delay information.

Message name (ID)	`SBG_ECOM_LOG_AIR_DATA (36)`				
Field	Description	Unit	Format	Size	Offset
TIME_STAMP / DELAY	Time since sensor is powered up or measurement delay	µs	uint32	4	0
AIRDATA_STATUS	Airdata information status	-	uint16	2	4
PRESSURE_ABS	Raw absolute pressure measured by the barometer sensor	Pa	float	4	6
ALTITUDE	Altitude computed from barometric altimeter	m	float	4	10
PRESSURE_DIFF	Raw differential pressure measured by the pitot tube	Pa	float	4	14
TRUE_AIRSPEED	True airspeed measured by the pitot tube	m.s^-1	float	4	18
AIR_TEMPERATURE	Outside air temperature used for airspeed computations	°C	float	4	22
				Total size	26

AIRDATA_STATUS definition

Bit	Name	Description
0 (LSB)	SBG_ECOM_AIR_DATA_TIME_IS_DELAY	Set to 1 if the TIME_STAMP field is a measurement delay instead of an absolute time stamping information.
1	SBG_ECOM_AIR_DATA_PRESSURE_ABS_VALID	Set to 1 if the absolute pressure field is filled and valid.
2	SBG_ECOM_AIR_DATA_ALTITUDE_VALID	Set to 1 if the barometric altitude field is filled and valid.
3	SBG_ECOM_AIR_DATA_PRESSURE_DIFF_VALID	Set to 1 if the differential pressure field is filled and valid.
4	SBG_ECOM_AIR_DATA_AIRSPEED_VALID	Set to 1 if the true airspeed field is filled and valid.
5	SBG_ECOM_AIR_DATA_TEMPERATURE_VALID	Set to 1 if the output air temperature field is filled and valid.

2.3.6.9. Doppler Velocity Log

This log stores either bottom or water track velocity data from a Doppler Velocity Log.

Time since reset is not aligned on main loop but instead of that, it dates the actual DVL velocity data.

Message name (ID)	`SBG_ECOM_LOG_DVL_BOTTOM_TRACK (29), SBG_ECOM_LOG_DVL_WATER_TRACK (30)`				
Field	Description	Unit	Format	Size	Offset
TIME_STAMP	Time since sensor is powered up	µs	uint32	4	0
DVL_STATUS	DVL velocity status bit-mask	-	uint16	2	4
VELOCITY_X	Velocity X expressed in the DVL instrument frame	m/s	float	4	6
VELOCITY_Y	Velocity Y expressed in the DVL instrument frame	m/s	float	4	10
VELOCITY_Z	Velocity Z expressed in the DVL instrument frame	m/s	float	4	14
VELOCITY_QUALITY_X	X velocity quality expressed in the DVL instrument frame	m/s	float	4	18
VELOCITY_QUALITY_Y	Y velocity quality expressed in the DVL instrument frame	m/s	float	4	22
VELOCITY_QUALITY_Z	Z velocity quality expressed in the DVL instrument frame	m/s	float	4	26
				Total size	30

DVL_STATUS definition

Bit	Name	Description
0 (LSB)	SBG_ECOM_DVL_VELOCITY_VALID	Set to 1 if the DVL equipment was able to measure a valid velocity.
1	SBG_ECOM_DVL_TIME_SYNC	Set to 1 if the data is accurately time stamped using a Sync In or Sync Out.

2.3.6.10. Depth Sensor

This log stores the pressure and depth information that comes from a subsea depth sensor.

Time since reset is not aligned on main loop but instead of that, it dates the actual depth measurement.

Message name (ID)	`SBG_ECOM_LOG_DEPTH (47)`				
Field	Description	Unit	Format	Size	Offset
TIME_STAMP	Time since sensor is powered up or delay	µs	uint32	4	0
DEPTH_STATUS	Depth sensor status bit-mask	-	uint16	2	4
PRESSURE_ABS	Absolute water pressure expressed in Pascals	Pa	float	4	6
DEPTH	Underwater depth measurement, positive upward	m/s	float	4	10
				Total size	14

DEPTH_STATUS definition

Bit	Name	Description
0 (LSB)	SBG_ECOM_DEPTH_TIME_IS_DELAY	Set to 1 if the time stamp field represents a delay instead of an absolute time stamp.
1	SBG_ECOM_DEPTH_PRESSURE_ABS_VALID	Set to 1 if the pressure field is filled and valid.
2	SBG_ECOM_DEPTH_ALTITUDE_VALID	Set to 1 if the depth altitude field is filled and valid.

2.3.6.11. USBL position

This log is used to retrieve the position as returned by an USBL beacon.

Time since reset is not aligned on main loop but instead of that, it dates the actual positioning data.

Message name (ID)	SBG_ECOM_LOG_USBL (37)				
Field	Description	Unit	Format	Size	Offset
TIME_STAMP	Time since sensor is powered up	µs	uint32	4	0
USBL_STATUS	USBL system status bitmask	-	uint16	2	4
LATITUDE	Latitude in degrees, positive north	°	double	8	6
LONGITUDE	Longitude in degrees, positive east	°	double	8	14
DEPTH	Depth in meters below mean sea level (positive down)	m	float	4	22
LATITUDE_STD	1 σ latitude accuracy in meters	m	float	4	26
LONGITUDE_STD	1 σ longitude accuracy in meters	m	float	4	30
DEPTH_STD	1 σ depth accuracy in meters	m	float	4	34
				Total size	38

USBL_STATUS definition

Bit	Name	Description
0 (LSB)	SBG_ECOM_USBL_TIME_SYNC	Set to 1 if the altimeter equipment was correctly initialized,
1	SBG_ECOM_USBL_POSITION_VALID	Set to 1 if the USBL data represents a valid 2D position
2	SBG_ECOM_USBL_DEPTH_VALID	Set to 1 if the USBL data has a valid depth information

SBG SYSTEMS

2.3.7. Miscellaneous logs

2.3.7.1. Event Markers

All SBG Systems Inertial Systems can detect events markers at up to 1 kHz on synchronization input signals such as Sync A, Sync B, Sync C, Sync D and Sync E. For each input synchronization signal, the device can output a binary log that returns the time of each received event during the last past 5 milliseconds (the maximum output rate is 200 Hz).

The TIME_STAMP field dates the first event that has been received during the last 5 ms. Other events received during the same time slot (5ms) are dated using a time offset to reduce the log size.

Sometimes, Sync Out signals are used to trigger for instance a camera either based on a time period or even based on a traveled distance (virtual odometer). The user is interested in having the exact time, attitude, velocity and position at which the picture has been taken.

To easily address this situation, SBG Systems products are also able to generate output event markers. The product will generate an electrical sync out signal and at the same time an output log that dates this event.

Example

If three events are received during the last 5 ms, each event will be dated using the following rules:

- First received event time is directly stored in TIME_STAMP
- Second received event time is TIME_STAMP + TIME_OFFSET_0
- Thrid received event time is TIME_STAMP + TIME_OFFSET_1

The other time offset fields will be set to 0 and the EVENT_STATUS flag will reflect which time offset fields are valid.

Message name (ID)	SBG_ECOM_LOG_EVENT_A (24), SBG_ECOM_LOG_EVENT_B (25), SBG_ECOM_LOG_EVENT_C (26), SBG_ECOM_LOG_EVENT_D (27), SBG_ECOM_LOG_EVENT_E (28), SBG_ECOM_LOG_EVENT_OUT_A (45), SBG_ECOM_LOG_EVENT_OUT_B (46),				
Field	Description	Unit	Format	Size	Offset
TIME_STAMP	Measurement time since the sensor power up.	µs	uint32	4	0
EVENT_STATUS	Status bit mask	-	uint16	2	4
TIME_OFFSET_0	Time offset for the second received event.	µs	uint16	2	6
TIME_OFFSET_1	Time offset for the third received event.	µs	uint16	2	8
TIME_OFFSET_2	Time offset for the fourth received event.	µs	uint16	2	10
TIME_OFFSET_3	Time offset for the fifth received event.	µs	uint16	2	12
				Total size	14

EVENT_STATUS definition

Bit	Name	Description
0 (LSB)	SBG_ECOM_EVENT_OVERFLOW	Set to 1 if we have received events at a higher rate than 1 kHz.
1	SBG_ECOM_EVENT_OFFSET_0_VALID	Set to 1 if at least two events have been received.
2	SBG_ECOM_EVENT_OFFSET_1_VALID	Set to 1 if at least three events have been received.
3	SBG_ECOM_EVENT_OFFSET_2_VALID	Set to 1 if at least four events have been received.
4	SBG_ECOM_EVENT_OFFSET_3_VALID	Set to 1 if five events have been received.

Note: The device can support events markers at up to 1 kHz. If too much events are sent, it may overload the internal CPU leading to decreased performance and reliability.

Warning: Never leave an activated Sync In signal unconnected as noise on the line may trigger spurious events at very high rates.

2.4. SBG_ECOM_CLASS_LOG_ECOM_1

This message class contains only special output logs that can be generated at high update rate (ie 1000Hz). Each log stores coherent data that can be fully interpreted without any additional information.

In addition, a time stamp expressed in microseconds is included in each log. It can be used to synchronize the data precisely.

Note: SBG Systems reserves the right to add at the end of logs new fields in future revision of the sbgECom protocol for upward compatibility. Therefore, user must consider the DATA sizes defined in this document as a minimum size.

Warning: Unlike other message classes, enabling this class on an output interface will disable any other output on the interface to maintain priority for high output rate messages.

2.4.1. Output Logs Overview

The following list, provides a quick overview of all available logs for this message class. It briefly describe which parameters are contained in each output log.

Name (MSG ID)	Description
SBG_ECOM_LOG_FAST_IMU_DATA (00)	1KHz Inertial Measurement Unit message – Calibrated Accelerometers and Gyroscopes sample

2.4.2. Output Logs Availability

The table below details the sbgECom binary logs availability according to a specific product family and product type. For example, if you have an ELLIPSE-E-G4A2-B1, you should refer to the column 'E' in the main column 'ELLIPSE'. Each available message is indicated with the • symbol.

	ELLIPSE					High Performance INS					
Name (Message ID)	I	A	E	N	D	A	E	N	D	M	U
SBG_ECOM_LOG_FAST_IMU_DATA (00)	•	•	•	•	•						

2.4.3. Fast Internal IMU data

Provides accelerometers, gyroscopes, time and status at 1KHz rate.

Message name (ID)	SBG_ECOM_LOG_FAST_IMU_DATA (00)					
Field	Description	Scale	Unit	Format	Size	Offset
TIME_STAMP	Time since sensor is powered up	1	µs	uint32	4	0
IMU_STATUS	IMU Status bitmask		-	uint16	2	4
ACCEL_X	Accelerometer sample – X axis	10^{-2}	m/s²	int16	2	6
ACCEL_Y	Accelerometer sample – Y axis	10^{-2}	m/s²	int16	2	8
ACCEL_Z	Accelerometer sampler – Z axis	10^{-2}	m/s²	int16	2	10
GYRO_X	Gyroscope sample – X axis	10^{-3}	rad/s	int16	2	12
GYRO_Y	Gyroscope sample – Y axis	10^{-3}	rad/s	int16	2	24
GYRO_Z	Gyroscope sample – Z axis	10^{-3}	rad/s	int16	2	16
					Total size	18

IMU_STATUS definition:

Status used to know if sensors are working correctly and are in their measurement range.

Bit	Name	Description
0 (LSB)	SBG_ECOM_IMU_COM_OK	Set to 1 if the communication with the IMU is ok.
1	SBG_ECOM_IMU_STATUS_BIT	Set to 1 if internal IMU passes Built In Test (Calibration, CPU)
2	SBG_ECOM_IMU_ACCEL_X_BIT	Set to 1 if accelerometer X passes Built In Test
3	SBG_ECOM_IMU_ACCEL_Y_BIT	Set to 1 if accelerometer Y passes Built In Test
4	SBG_ECOM_IMU_ACCEL_Z_BIT	Set to 1 if accelerometer Z passes Built In Test
5	SBG_ECOM_IMU_GYRO_X_BIT	Set to 1 if gyroscope X passes Built In Test
6	SBG_ECOM_IMU_GYRO_Y_BIT	Set to 1 if gyroscope Y passes Built In Test
7	SBG_ECOM_IMU_GYRO_Z_BIT	Set to 1 if gyroscope Z passes Built In Test
8	SBG_ECOM_IMU_ACCELS_IN_RANGE	Set to 1 if accelerometers are within operating range
9	SBG_ECOM_IMU_GYROS_IN_RANGE	Set to 1 if gyroscopes are within operating range

3. NMEA Protocol description

In order to allow straightforward integration with third party equipment and software, the Inertial System supports NMEA messages input and outputs.

3.1. NMEA sentences format

The implemented NMEA sentences are based on NMEA 0183 Version 4.1.

The following example is described in the table below:

```
$GPZDA,201530.00,04,07,2002,00,00*60<CR><LF>
```

Field	Value	Description	Example
Start of frame	$	All frames start with $	$
Talker ID	<XX>	GP for GPS GL for GLONASS...	GP
Sentence Formater	<XXX>	Type of message content	ZDA
[,value]		Data field are separated by a ',' Data field can vary even for a certain field	,201530.00,04,07,2002,00,00
Check-sum	*<Checksum>	Start with a '*' and consist of a 2 characters representing a 8 bits hex value. The checksum is the XOR of all previous values except '$' and '*'	*60
End of Frame	<CR><LF>	All frames end with a carriage return and line feed.	<CR><LF>

Note 1: For each output interface, the NMEA talker ID may be configured accordingly. When input NMEA data are sent to the AHRS/INS, the talker ID field is ignored.

Note 2: Each data field is comma separated. Sometimes, a field cannot be defined and can be left empty. In this case the frame may contain several blank fields such as in the following example:

$GPZDA,,,,,,*XX<CR><LF>

3.1.1. NMEA types conventions

To ease NMEA messages definitions, we define two conventions for both integers and decimal number format.

3.1.1.1. Integer numbers

Integer numbers are represented using the char 'i'. The number of 'i' chars define the maximum number of digits that can be used to represent this integer.

The char '-' is prepended to represent a negative integer number.

Example

The integer format iii could be used to represent the following integers: -234, 13, -3

3.1.1.2. Decimal numbers

Decimal numbers are represented by the char 'f'. The char '.' is used to separate the integer part from the decimal one. The number of 'f' chars define the maximum number of digits that can be used to represent both the integer and decimal part.

The char '-' or '+' can be prepended to represent negative or positive decimal number.

Example

The decimal format ff.fff could be used to represent the following decimal numbers: -34.2, 1.205, 24.126

3.1.1.3. Time convention

The time expressed in NMEA messages is always expressed in terms of UTC time. It has the following form:

hhmmss.ss. As an example 125430.0 corresponds to the following UTC Time: 12:54 and 30.00 seconds.

3.2. SBG_ECOM_CLASS_LOG_NMEA_0

The AHRS/INS can output standard NMEA 0183 version 4.1 logs for GPS drop in replacement and to ease integration with third party systems.

All NMEA outputs generated by the INS are based on the Inertial Navigation Solution and are not related directly to aiding data such as from a GNSS receiver.

For instance, even if a GNSS receiver is providing an RTK solution, if the INS is not aligned the product will output GGA messages with a quality indicator of 0 (invalid). This behavior is expected as an INS can still deliver RTK like accuracy solutions even during GNSS outages.

3.2.1. Quality indicators

NMEA messages provide different quality indicators. The table below details how these quality indicators are translated from the Inertial Navigation Sol oution position standard deviation to NMEA standard.

Horizontal Accuracy	> 100 m	< 100 m	< 10m	< 1.2 m	< 30 cm	< 10 cm
NMEA name Navigational Status	Invalid Fix Not valid	Dead Reckoning Caution	Standalone Fix Safe	DGPS Fix Safe	Floating RTK Safe	Fixed RTK Safe
GGA quality indicator	0	6	1	2	5	4
GGK quality indicator	0		1	4	2	3
RMC/VTG mode	N	E	A	D	F	R
RMC Nav. Status	V	C	S	S	S	S

3.2.2. DOP computation

The DOP (Dillution of Precision) is a specific GNSS related quality indicator only based on Space Vehicles geometry it has thus no meaning for an INS. This DOP parameter is today mostly deprecated as INS/GNSS algorithms now provide much better and reliable quality indicators.

However, some software still use the DOP information to discard outliers and inaccurate position solutions. The INS outputs a DOP based directly on the estimated position standard deviation:

$$DOP_{horizontal} = \sqrt{North_{Std}^2 + East_{Std}^2}$$

3.2.3. NMEA Logs Overview

Name (MSG ID)	Description
SBG_ECOM_LOG_NMEA_GGA (00)	Latitude, Longitude, Altitude, Quality indicator
SBG_ECOM_LOG_NMEA_RMC (01)	Latitude, Longitude, velocity, course over ground
SBG_ECOM_LOG_NMEA_VTG (08)	Track and speed over the ground
SBG_ECOM_LOG_NMEA_ZDA (02)	UTC date and time
SBG_ECOM_LOG_NMEA_HDT (03)	Heading from true north
SBG_ECOM_LOG_NMEA_GST (04)	Position error statistics
SBG_ECOM_LOG_NMEA_VBW (05)	Dual ground / water Speed
SBG_ECOM_LOG_NMEA_DPT (07)	Depth of water

3.2.4. NMEA Logs Availability

The table below details the NMEA logs availability according to a specific product family and product type. For example, if you have an ELLIPSE-E-G4A2-B1, you should refer to the column 'E' in the main column 'ELLIPSE'. Each available message is indicated with the ● symbol.

	ELLIPSE					High Performance INS					
Name (Message ID)	I	A	E	N	D	A	E	N	D	M	U
SBG_ECOM_LOG_NMEA_GGA (00)			●	●	●		●	●	●		●
SBG_ECOM_LOG_NMEA_RMC (01)			●	●	●		●	●	●		●
SBG_ECOM_LOG_NMEA_VTG (08)			●	●	●		●	●	●		●
SBG_ECOM_LOG_NMEA_ZDA (02)	●	●	●	●	●	●	●	●	●	●	●
SBG_ECOM_LOG_NMEA_HDT (03)		●	●	●	●	●	●	●	●	●	●
SBG_ECOM_LOG_NMEA_GST (04)			●	●	●		●	●	●		●
SBG_ECOM_LOG_NMEA_VBW (05)			●	●	●		●	●	●		●
SBG_ECOM_LOG_NMEA_DPT (07)							●	●	●		●

3.2.5. GGA message

Name in sbgECom convention (msg ID): **SBG_ECOM_LOG_NMEA_GGA (00)**

The GGA log provides detailed Kalman filtered position, altitude and accuracy data.

Message format

```
$GPGGA,,,,,,0,00,20.0,,,,,,*7A<CR><LF>
$GPGGA,000010.00,4852.10719,N,00209.42313,E,0,00,0.0,-44.7,M,0.0,M,,,*63<CR><LF>
```

Field	Name	Format	Description
0	$##GGA	string	Message ID – GGA frame
1	Time	hhmmss.ss	UTC Time, current time
2	Latitude	ddmm.mmmmmmmm	Latitude: degree + minutes with 8 decimal digits
3	N/S	char	North / South indicator
4	Longitude	dddmm.mmmmmmmm	Longitude: degree + minutes with 8 decimal digits
5	E/W	char	East / West indicator
6	Quality	i	Solution quality (see definition in Quality indicators)
7	SV used	ii	Number of satellites used in solution
8	Horizontal DOP	ff.f	Horizontal dilution of precision, (see definition in Quality indicators)
9	Altitude MSL	ffff.fff	Altitude above Mean Sea Level in meters
10	M	M	Altitude unit (Meters) fixed field.
11	Undulation	fff.fff	Geoidal separation between WGS-84 and MSL in meters).
12	M	M	Units for geoidal separation (Meters) fixed field.
13	Diff. Age	-	Age of differential corrections. Not filled by the device, always empty.
14	Diff. station ID	-	Differential station id. Not filled by the device, always empty.
15	Check sum	*cs	Xor of all previous bytes except $
16	End of frame	<CR><LF>	Carriage return and line feed

Note: The position quality and DOP are computed using the INS position estimated standard deviation. It is not related to the GNSS solution directly.

3.2.6. RMC message

Name in sbgECom convention (msg ID): **SBG_ECOM_LOG_NMEA_RMC (01)**

This is the “minimum recommended GNSS data” frame that contains Kalman enhanced 2D position, velocity and course over ground as well as quality indicators.

Message format

```
$GPRMC,,V,,,,,,,,,,N,V*29<CR><LF>
$GPRMC,010802.26,A,4852.13326,N,00209.49001,E,0.2,195.49,290512,,,A*67<CR><LF>
```

Field	Name	Format	Description
0	$##RMC	string	Message ID – RMC frame
1	time	hhmmss.ss	UTC Time, current time
2	status	char	Status field: A = Valid data. V = Invalid data.
3	latitude	ddmm.mmmmmmmm	Latitude: degree + minutes with 8 decimal digits
4	N/S	char	North / South indicator
5	longitude	dddmm.mmmmmmmm	Longitude: degree + minutes with 8 decimal digits
6	E/W	char	East / West indicator
7	speed	fff.f	Speed over ground in Knots
8	course	fff.f	Course over Ground in degrees [0; 360]
9	date	ddmmyy	UTC day, month, year
10	variation	fff.ff	Magnetic variation value in degrees [0; 180]
11	E/W	char	Direction of magnetic variation (East / West)
12	mode	char	Position mode indicator (see definition in Quality indicators section)
13	navStatus	char	Navigational status indicator (see definition in Quality indicators section)
14	Check sum	*cs	Xor of all previous bytes except $
15	End of frame	<CR><LF>	Carriage return and line feed

Note: The position mode and navigational status are computed using the INS position estimated standard deviation. It is not related to the GNSS solution directly.

3.2.7. VTG message

Name in sbgECom convention (msg ID): **SBG_ECOM_LOG_NMEA_VTG (08)**

This NMEA frame contains the track and velocity over ground as computed by the Kalman filter. It also includes a quality indicator as for the RMC message.

Message format

```
$GPVTG,,,,,,,,,N*30<CR><LF>
$GPVTG,256.31,T,256.44,M,45.401,N,84.084,K,N*2A<CR><LF>
```

Field	Name	Format	Description
0	$##VTG	string	Message ID – RMC frame
1	Course True	fff.ff	True course over ground in degrees [0; 360]
2	T	char	Course over ground is relative to true north
3	Course Magnetic	fff.ff	Magnetic course over ground in degrees [0; 360]
4	M	char	Course over ground is relative to magnetic north
5	Speed Knots	fff.fff	Ground speed in knots
6	N	char	The speed is expressed in knots
7	Speed Km/h	fff.fff	Ground speed in kilometers per hour
8	K	char	The speed is expressed in kilometers per hour
9	mode	char	Position mode indicator (see definition in Quality indicators section)
10	Check sum	*cs	Xor of all previous bytes except $
11	End of frame	<CR><LF>	Carriage return and line feed

Note: The position mode indicator is computed using the INS position estimated standard deviation. It is not related to the GNSS solution directly.

3.2.8. ZDA message

Name in sbgECom convention (msg ID): **SBG_ECOM_LOG_NMEA_ZDA (02)**

This message contains UTC time and date information.

Message format

```
$GPZDA,,,,,,*48<CR><LF>
$GPZDA,201530.00,04,07,2002,00,00*60<CR><LF>
```

Field	Name	Format	Description
0	$##ZDA	string	Message ID – ZDA frame
1	Time	hhmmss.ss	UTC Time, current time
2	Day	dd	Day of month [01 - 31]
3	Month	mm	Month of year [01 - 12]
4	Year	yyyy	Year (4 digits)
5	Ltzh	0	Local zone hours (not supported, fixed 00)
6	Ltzh	0	Local zone minutes (not supported, fixed 00)
7	Check sum	*cs	Xor of all previous bytes except $
8	End of frame	<CR><LF>	Carriage return and line feed

3.2.9. HDT Message

Name in sbgECom convention (msg ID): **SBG_ECOM_LOG_NMEA_HDT (03)**

The message output the INS Kalman filtered true heading value. The true heading is the direction that the vehicle is pointing and is not necessarily the direction of travel (course over ground).

Message format

```
$GPHDT,,T*1B<CR><LF>
$GPHDT,191.94,T*01<CR><LF>
```

Field	Name	Format	Description
0	$##HDT	string	Message ID – HDT frame
1	Heading	fff.ff	True heading in degrees [0 - 360]
2	T	char	T means true heading
3	Check sum	*cs	Xor of all previous bytes except $
4	End of frame	<CR><LF>	Carriage return and line feed

3.2.10. GST Message

Name in sbgECom convention (msg ID): **SBG_ECOM_LOG_NMEA_GST (04)**

The message output detailed position error statistics of the Kalman filtered position solution. Please keep in mind that the returned data reflects the estimated inertial position and not the GNSS quality fix directly.

Message format

```
$GPGST,,,,,,,,*67<CR><LF>
$GPGST,172814.00,,0.023,0.020,273.62,0.023,0.015,0.031*46<CR><LF>
```

Field	Name	Format	Description
0	$##GST	string	Message ID – GST frame
1	Time	hhmmss.ss	UTC Time, current time
2	psrResidual	NULL	RMS value of pseudorange residuals. Always NULL, not supported.
3	sMajorAxisError	fff.fff	Error ellipse semi-major axis 1 sigma error, in meters
4	sMinorAxisError	fff.fff	Error ellipse semi-minor axis 1 sigma error, in meters
5	errorEllipseAng	fff.ff	Error ellipse orientation, degrees from true north [0 - 360]
6	latError	fff.fff	Latitude 1 sigma error, in meters
7	longError	fff.fff	Longitude 1 sigma error, in meters
8	altError	fff.fff	Height 1 sigma error, in meters
9	Check sum	*cs	Xor of all previous bytes except $
10	End of frame	<CR><LF>	Carriage return and line feed

3.2.11. VBW Message

Name in sbgECom convention (msg ID): **SBG_ECOM_LOG_NMEA_VBW (05)**

The message outputs Ground and Water speed from the INS data fusion algorithm. The data are expressed in the INS (body) frame.

This log can be filled correctly only if the INS receives valid bottom tracking and water tracking DVL data. Bottom and water velocities are used by the Inertial Navigation System Kalman filter to estimate the water current and thus compute the water speed in the vessel coordinate frame.

Message format

```
$GPVBW,,,,,,,,*67<CR><LF>
$GPVBW,0.312,0.910,A,0.410,0.950,A*55<CR><LF>
```

Field	Name	Format	Description
0	$##VBW	string	Message ID – VBW frame
1	longWaterSpeed	fff.fff	Longitudinal water speed, knots (positive forward)
2	transvWaterSpeed	fff.fff	Transverse water speed, knots (positive right)
3	waterSpeedValid	char	Status: Water speed, A = Data valid, V = Invalid
4	longGroundSpeed	fff.fff	Longitudinal ground speed, knots (positive forward)
5	transvGroundSpeed	fff.fff	Transverse ground speed, knots (positive right)
6	groundSpeedValid	char	Status: Ground speed, A = Data valid, V = Invalid
7	Check sum	*cs	Xor of all previous bytes except $
10	End of frame	<CR><LF>	Carriage return and line feed

3.2.12. DPT Message

Name in sbgECom convention (msg ID): **SBG_ECOM_LOG_NMEA_DPT (07)**

The message outputs the depth in meters as received by the device by an external pressure sensor. This value is untouched by the device and is basically a copy of the receive data.

This log can be filled correctly only if valid pressure sensor data are feed to the INS.

Message format

```
$GPDPT,0.000,,*55<CR><LF>
```

Field	Name	Format	Description
0	$##DPT	string	Message ID – DPT frame
1	Depth	fff.fff	Depth below surface in meters (positive down)
2	Offset	-	Offset from transducer. Not supported and returned empty
3	Check sum	*cs	Xor of all previous bytes except $
4	End of frame	<CR><LF>	Carriage return and line feed

3.3. SBG_ECOM_CLASS_LOG_NMEA_1

This message class contains all proprietary NMEA messages and other non standard messages that use the NMEA formatting.

NMEA_1 logs contain Kalman filtered navigation, velocity and attitude data. External (GNSS) aiding data are not used to generate these logs.

3.3.1. NMEA Logs Overview

Name (MSG ID)	Description
SBG_ECOM_LOG_NMEA_1_PRDID (00)	RDI proprietary sentence. Pitch, Roll, Heading
SBG_ECOM_LOG_NMEA_1_PSBGI (01)	SBG Systems proprietary sentence. Rotation rates, accelerations.
SBG_ECOM_LOG_NMEA_1_PASHR (02)	Ashtech NMEA message – Roll, pitch, Heading, heave
SBG_ECOM_LOG_NMEA_1_PSBGB (04)	SBG Systems proprietary sentence. Attitude, heading, heave, angular rates, velocity
SBG_ECOM_LOG_NMEA_1_PHINF (05)	Ixblue NMEA like log used to output Status information
SBG_ECOM_LOG_NMEA_1_PHTRO (06)	Ixblue NMEA like log used to output attitude and ship motion
SBG_ECOM_LOG_NMEA_1_PHLIN (07)	Ixblue NMEA like log used to output Roll and Pitch
SBG_ECOM_LOG_NMEA_1_PHOCT (08)	Ixblue NMEA like log used to output Surge, Sway and Heave
SBG_ECOM_LOG_NMEA_1_INDYN (09)	Ixblue NMEA like log used to output position, heading, attitude, rate and velocity
SBG_ECOM_LOG_NMEA_1_GGK(10)	Trimble NMEA like log with Time, Latitude, Longitude, Ellipsoidal height

3.3.2. NMEA Logs Availability

The table below details the NMEA logs availability according to a specific product family and product type. For example, if you have an ELLIPSE-E-G4A2-B1, you should refer to the column 'E' in the main column 'ELLIPSE'. Each available message is indicated with the ● symbol.

	ELLIPSE					High Performance INS					
Name (Message ID)	I	A	E	N	D	A	E	N	D	M	U
SBG_ECOM_LOG_NMEA_1_PRDID (00)		●	●	●	●	●	●	●	●	●	●
SBG_ECOM_LOG_NMEA_1_PSBGI (01)	●	●	●	●	●	●	●	●	●	●	●
SBG_ECOM_LOG_NMEA_1_PASHR (02)		●	●	●	●	●	●	●	●	●	●
SBG_ECOM_LOG_NMEA_1_PSBGB (04)						●	●	●	●	●	●
SBG_ECOM_LOG_NMEA_1_PHINF (05)						●	●	●	●	●	●
SBG_ECOM_LOG_NMEA_1_PHTRO (06)						●	●	●	●	●	●
SBG_ECOM_LOG_NMEA_1_PHLIN (07)						●	●	●	●	●	●
SBG_ECOM_LOG_NMEA_1_PHOCT (08)						●	●	●	●	●	●
SBG_ECOM_LOG_NMEA_1_INDYN (09)		●	●	●	●						
SBG_ECOM_LOG_NMEA_1_GGK(10)							●	●	●		●

3.3.3. PRDID Message

Name in sbgECom convention (msg ID): **SBG_ECOM_LOG_NMEA_1_PRDID (00)**

This Teledyne RDI proprietary message outputs the vessel pitch, roll and true heading angles in degrees. It uses an NMEA style formatting.

Message format

```
$PRDID,-012.39,+002.14,366.91*7A<CR><LF>
```

Field	Name	Format	Description
0	$PRDID	string	Message ID – Teledyne RDI proprietary NMEA identifier
1	Pitch	fff.ff	Signed vessel pitch in degrees, positive bow up.
2	Roll	fff.ff	Signed vessel roll in degrees, positive port up.
3	Heading	fff.ff	Vessel true heading in degrees [0 - 360]
4	Check sum	*cs	Xor of all previous bytes except $
5	End of frame	<CR><LF>	Carriage return and line feed

3.3.4. PSBGI Message

Name in sbgECom convention (msg ID): **SBG_ECOM_LOG_NMEA_1_PSBGI (01)**

This message is a SBG Systems proprietary NMEA log that provides accelerations and rotation rates in the sensor frame. It also includes accurate UTC time stamping.

Message format

```
$PSBGI,003944.74,-0.08,0.07,0.00,-0.02,0.06,-9.72,*42<CR><LF>
```

Field	Name	Format	Description
0	$PSBGI	string	Message ID – SBG Systems proprietary NMEA identifier
1	timeUTC	hhmmss.sss	Current UTC time
2	gyroX	fff.fff	X Rotation rate from -999.999°/s to +999.999°/s
3	gyroY	fff.fff	Y Rotation rate from -999.999°/s to +999.999°/s
4	gyroZ	fff.fff	Z Rotation rate from -999.999°/s to +999.999°/s
5	accelX	fff.fff	X acceleration in m/s^2 from -999.999 to +999.999
6	accelY	fff.fff	Y acceleration in m/s^2 from -999.999 to +999.999
7	accelZ	fff.fff	Z acceleration in m/s^2 from -999.999 to +999.999
8	Check sum	*cs	Xor of all previous bytes except $
9	End of frame	<CR><LF>	Carriage return and line feed

3.3.5. PASHR Message

Name in sbgECom convention (msg ID): **SBG_ECOM_LOG_NMEA_1_PASHR (02)**

This message is a NMEA proprietary log that provides roll, pitch, heading and heave outputs.

Message format

```
$PASHR,085335.000,224.19,T,-01.26,+00.83,+00.00,0.101,0.113,0.267,1,0*06<CR><LF>
```

Field	Name	Format	Description
0	$PASHR	string	Message ID – Ixblue proprietary NMEA identifier
1	timeUTC	hhmmss.ss	Current UTC time
2	heading	fff.ff	Heading angle, in decimal degrees [0 – 360]
3	T	char	True Heading
4	roll	fff.ff	Roll in decimal degrees. From [-180 – 180°]
5	pitch	fff.ff	Pitch in decimal degrees. From [-90 – 90°]
6	heave	fff.ff	Heave, in meters, positive down
7	rollStd	fff.ff	Roll angle standard deviation in decimal degrees
8	pitchStd	fff.ff	Pitch angle standard deviation in decimal degrees
9	headingStd	fff.ff	Heading angle standard deviation in decimal degrees
10	posStatus	char	Position Quality status 0 = No position 1 = All non-RTK fixed integer positions 2 = RTK fixed integer position
11	imuStatus	char	IMU & Sensor Status 0 = IMU is working correctly 1 = IMU sensor error
12	Check sum	*cs	Xor of all previous bytes except $
13	End of frame	<CR><LF>	Carriage return and line feed

3.3.6. PSBGB Message

Name in sbgECom convention (msg ID): **SBG_ECOM_LOG_NMEA_1_PSBGB (04)**

This NMEA proprietary log provides UTC time of day, heading, roll, pitch, heave, angular rates, and body velocities as well as standard deviations and status.

Message format

```
$PSBGB,1,000344.000,0,3.529,-12.821,6.122,0.101,0.098,10.117,0,0,0.004,0.050,2,0.772,0.004,-0.017
,1.043,4.476,0.171,866.025,0,*53<CR><LF>
```

Field	Name	Format	Description
0	$PSBGB	string	Message ID – SBG Systems proprietary NMEA identifier
1	1	char	Version of this message – 1
2	timeUTC	hhmmss.sss	Current UTC time
3	utcStatus	char	UTC time validity indicator: '0' = Invalid '1' = Valid and PPS synchronized '2' = Valid but no PPS synchronized '3' = Unknown leap second and PPS synchronized '4' = Unknown leap second and no PPS synchronized
4	roll	fff.fff	Roll angle in decimal degrees, positive port side up [-180; 180°].
5	pitch	fff.fff	Pitch in decimal degrees, positive bow up [-90; 90°]. [-90; 90°]
6	heading	fff.fff	True Heading angle, in decimal degrees [0; 360°]
7	rollStd	fff.fff	Roll standard deviation in decimal degrees [0; 180°]
8	pitchStd	fff.fff	Pitch standard deviation in decimal degrees [0; 180°]
9	headingStd	fff.fff	True Heading standard deviation in decimal degrees [0; 180°]
10	rollPitchStatus	char	Roll and Pitch validity flag: '0' = Invalid (Standard deviation above threshold) '1' = Optimal accuracy '2' = Degraded accuracy, alignment in progress
11	headingStatus	char	Heading validity flag: '0' = Invalid (Standard deviation above threshold) '1' = Optimal accuracy '2' = Degraded accuracy, alignment in progress
12	heave	ff.fff	Heave, in meters, positive down.
13	heaveStd	ff.fff	Heave standard deviation in meters. – Fixed to 5cm
14	heaveStatus	char	Heave status flag: '0' = Invalid or initializing '1' = Valid and velocity aided '2' = Valid, standalone
15	rollRate	fff.fff	Roll angular rate in °/s, positive port side up.
16	pitchRate	fff.fff	Pitch angular rate in °/s, positive bow up.
17	yawRate	fff.fff	Yaw angular rate in °/s, positive clockwise.
18	velocityX	fff.fff	Velocity in X body axis in m/s, positive forward.
19	velocityY	fff.fff	Velocity in Y body axis in m/s, positive starboard.
20	velocityZ	fff.fff	Velocity in Z body axis in m/s, positive down.
21	velocityStd	fff.fff	Norm of X, Y and Z velocity standard deviation in m/s.
22	velocityStatus	char	Velocity validity flag: '0' = Invalid (Standard deviation above threshold) '1' = Optimal accuracy '2' = Degraded accuracy, alignment in progress
23	Check sum	*cs	Xor of all previous bytes except $

24	End of frame	<CR><LF>	Carriage return and line feed

3.3.7. Ixblue PHINF

Name in sbgECom convention (msg ID): **SBG_ECOM_LOG_NMEA_1_PHINF (05)**

This Ixblue proprietary message outputs the general system status. It uses an NMEA style formatting.

Message format

```
$PHINF,08030027*7B<CR><LF>
```

Field	Name	Format	Description
0	$PHINF	string	Message ID – Ixblue proprietary NMEA identifier
1	Status	hhhhhhhh	Hexadecimal value of the INS status
2	Check sum	*cs	Xor of all previous bytes except $
3	End of frame	<CR><LF>	Carriage return and line feed

STATUS definition

The OCTANS Status is a 32 bits word which acts as a built-in test and control of system. Different OCTANS modes as well as optical and electrical sub-parts status are monitored in real time through dedicated flags. Each flag is a bit which is set to “1” when the flag is ON and set to “0” when the flag is OFF.

As these status are defined for a different product than an SBG Systems AHRS or INS, some status couldn't be directly translated.

Bit	Name	Description
0	IXBLUE_STAT_HEADING_UNVALID	Set when the provided heading is invalid or still converging
1	IXBLUE_STAT_ROLL_UNVALID	Set when the provided roll is invalid or still converging
2	IXBLUE_STAT_PITCH_UNVALID	Set when the provided pitch is invalid or still converging
3	IXBLUE_STAT_HEAVE_INIT	Set when the heave filter is in initialization phase (not fully accurate)
4	Reserved	Reserved, not used
5	IXBLUE_STAT_ALIGNMENT	Set during alignment phase (not fully accurate)
6	IXBLUE_STAT_CONFIG_SAVED	Not implemented
7	IXBLUE_STAT_COMPUTATION_OVERLOAD	Set when the CPU is overloaded
8	IXBLUE_STAT_FOG_X1_ANOMALY	Set when the gyroscope X built in test has failed
9	IXBLUE_STAT_FOG_X2_ANOMALY	Set when the gyroscope Y built in test has failed
10	IXBLUE_STAT_FOG_X3_ANOMALY	Set when the gyroscope Z built in test has failed
11	IXBLUE_STAT_FOG_ACQ_ERROR	Set when at least one gyroscope is out of range
12	IXBLUE_STAT_ACC_X1_ANOMALY	Set when the accelerometer X built in test has failed
13	IXBLUE_STAT_ACC_X2_ANOMALY	Set when the accelerometer Y built in test has failed
14	IXBLUE_STAT_ACC_X3_ANOMALY	Set when the accelerometer Z built in test has failed
15	IXBLUE_STAT_SENSOR_ERROR	Set when at least one sensor is failing or is out of range
16	IXBLUE_STAT_SERIAL_IN_A_ERROR	Set when error are detected on the serial port A rx

Bit	Name	Description
17	IXBLUE_STAT_SERIAL_IN_B_ERROR	Set when error are detected on the serial port B rx
18	IXBLUE_STAT_SERIAL_IN_C_ERROR	Set when error are detected on the serial port C rx
19	IXBLUE_STAT_OUTPUT_OVERLOADED	Set when at least one serial output is overloaded
20	IXBLUE_STAT_SERIAL_OUT_A_FULL	Set when the serial output A is overloaded
21	IXBLUE_STAT_SERIAL_OUT_B_FULL	Set when the serial output B is overloaded
22	IXBLUE_STAT_SERIAL_OUT_C_FULL	Set when the serial output C is overloaded
23	IXBLUE_STAT_SERIAL_OUT_D_FULL	Not implemented
24	IXBLUE_STAT_MANUAL_LOG_USED	Not implemented
25	IXBLUE_STAT_MANUAL_LAT_USED	Not implemented
27	IXBLUE_STAT_HRP_INVALID	Set if at least one roll, pitch and heading component is invalid
31	IXBLUE_STAT_RESTART_SYSTEM	Not implemented

3.3.8. Ixblue PHTRO

Name in sbgECom convention (msg ID): **`SBG_ECOM_LOG_NMEA_1_PHTRO (06)`**

This Ixblue proprietary message outputs the unit pitch and roll angles. It uses an NMEA style formatting.

Message format

```
$PHTRO,0.03,P,0.22,T*56<CR><LF>
```

Field	Name	Format	Description
0	$PHTRO	string	Message ID – Ixblue proprietary NMEA identifier
1	Pitch	x.xx	Pitch angle in degrees
3	Pitch sign	a	'M' for bow up and 'P' for bow down
4	Roll	y.yy	Roll angle in degrees
5	Roll sign	b	'B' for port down and 'T' for port up
6	Check sum	*cs	Xor of all previous bytes except
7	End of frame	<CR><LF>	Carriage return and line feed

3.3.9. Ixblue PHLIN

Name in sbgECom convention (msg ID): **SBG_ECOM_LOG_NMEA_1_PHLIN (07)**

This Ixblue proprietary message outputs surge, sway and heave. It uses an NMEA style formatting.

Message format

```
$PHTRO,0.03,P,0.22,T*56<CR><LF>
```

Field	Name	Format	Description
0	$PHLIN	string	Message ID – Ixblue proprietary NMEA identifier
1	Surge	x.xxx	Signed surge in meters (positive forward)
3	Sway	y.yyy	Signed sway in meters (positive left)
4	Heave	z.zzz	Signed heave in meters (positive up)
5	Check sum	*cs	Xor of all previous bytes except
6	End of frame	<CR><LF>	Carriage return and line feed

Warning: Ixblue uses different conventions for ship motion measurements. In this frame, both sway and heave values are reversed compared to SBG Systems convetions.

3.3.10. Ixblue PHOCT

Name in sbgECom convention (msg ID): **SBG_ECOM_LOG_NMEA_1_PHOCT (08)**

This Ixblue proprietary message outputs time, attitude, heading and ship motion data. It uses an NMEA style formatting.

Message format

```
$PHOCT,01,000201.000,E,00,356.592,E,+000.225,E,+00.039,E,+00.023,T,
+00.023,+00.016,+00.003,+00.002,-00.001,+00.000,+0001.96*04<CR><LF>
```

Field	Name	Format	Description
0	$PHOCT	string	Message ID – Ixblue proprietary NMEA identifier
1	01	string	Protocol version identifier
2	Time	hhmmss.ss	UTC Time, current time
3	UTC Status	G	UTC Time status: 'T'=Valid 'E'=Invalid
4	Latency	AA	INS latency for heading, roll, pitch (Not implemented, always set to 0)
5	True Heading	HHH.HHH	True heading in degree (from 000.000 to 359.999)
6	Heading Status	N	True Heading status: 'T'=Valid; 'E'=Invalid; 'I'=Initializing
7	Roll Angle	RRR.RRR	Roll in degree (positive if port side up); from -180.000 to +180.000
8	Roll Status	L	Roll status: 'T'=Valid 'E'=Invalid 'I'=Initializing
9	Pitch Angle	PP.PPP	Pitch degree (positive if bow down), from -90.000 to +90.000
10	Pitch Status	K	Pitch status: 'T'=Valid 'E'=Invalid 'I'=Initializing
11	Primary Heave	FF.FFF	Heave at Primary Lever arm in meters (positive up), from -99.999 to +99.999
12	Heave Status	M	Heave status (also used for surge, sway & speed) : T=Valid E=Invalid I=Initializing
13	Heave	HH.HHH	Heave at desired lever arm in meters (positive up), from -99.999 to +99.999
14	Surge	SS.SSS	Surge with Lever arms applied in meters (positive forward), from -99.999 to +99.999
15	Sway	WW.WWW	Sway at desired lever arm in meters (Positive left) from -99.999 to +99.999
16	Heave Speed	ZZ.ZZZ	Heave speedat desired lever arm in m/s (Positive up) from -99.999 to +99.999
17	Surge Speed	YY.YYY	Surge speed at desired lever arm in m/s positive forward, from -99.999 to +99.999
18	Sway Speed	XX.XXX	Sway speed at desired lever arm in m/s (Positive left), from -99.999 to +99.999
19	Heading Rate	QQQQ.QQ	Heading rate of turn in °/min (Positive clockwize) from -9999.99 to +9999.99
20	Check sum	*cs	Xor of all previous bytes except
21	End of frame	<CR><LF>	Carriage return and line feed

Warning: Ixblue uses different conventions for ship motion measurements. In this frame, both sway and heave values are reversed compared to SBG Systems conventions.

3.3.11. Ixblue INDYN

Name in sbgECom convention (msg ID): **SBG_ECOM_LOG_NMEA_1_INDYN (09)**

This Ixblue proprietary message outputs position, heading, attitude, attitude rate and speed data. It uses an NMEA style formatting.

Message format

```
$INDYN,48.87949927,1.99962275,0.000,218.714,-0.909,0.291,-0.011,-0.073,-0.024,0.019*6A<CR><LF>
```

Field	Name	Format	Description
0	$INDYN	string	Message ID – Ixblue proprietary NMEA identifier
1	Latitude	x.xxxxxxxx	INS latitude in degrees with 8 decimals
2	Longitude	y.yyyyyyyy	INS longitude in degrees with 8 decimals
3	Altitude	z.zzz	INS altitude above Mean Sea Level in meters positive upward
4	True Heading	h.hhh	True heading in degrees (from 0.000 to 359.999)
5	Roll angle	r.rrr	Roll angle in degrees positive for port side up (from -180.000 to +180.000)
6	Pitch angle	p.ppp	Pitch angle in degrees positive when bow down (from -90.000 to +90.000)
7	Heading rate	a.aaa	Heading rate in °/s (positive when heading increases)
8	Roll rate	b.bbb	Roll rate in °/s (positive when roll increases)
9	Pitch rate	c.ccc	Pitch rate in °/s (positive when pitch increases)
10	Ground speed	s.sss	Horizon speed in m/s (positive toward the bow)
20	Check sum	*cs	Xor of all previous bytes except
21	End of frame	<CR><LF>	Carriage return and line feed

Note: The roll, pitch and heading rates are not the same as the gyroscopes values. Gyroscopes values are indeed expressed in body frame (X,Y,Z) and are not unbiased.

Warning: In this message, Ixblue uses a different convention for orientation. The pitch angle is reversed compared to SBG Systems conventions.

3.3.12. Trimble GGK

Name in sbgECom convention (msg ID): **SBG_ECOM_LOG_NMEA_1_GGK(10)**

This Trimble proprietary message outputs time, position, and GNSS related quality indicators. It uses an NMEA style formatting.

Message format

```
$PTNL,GGK,,,,,,,0,00,,,M*30<CR><LF>
$PTNL,GGK,161159.00,013020,4854.61758182,N,00210.08881241,E,1,07,8.3,EHT140.509,M*75<CR><LF>
```

Field	Name	Format	Description
0	$PTNL	string	Talker ID – Trimble proprietary NMEA identifier, always PTNL
1	GGK	string	Message ID – Trimble proprietary NMEA identifier, always GGK
2	Time	hhmmss.ss	UTC time of solution, hours must be two chars
3	Date	mmddyy	UTC date of solution, month must be two chars
4	Latitude	ddmm.mmmmmmmm	INS latitude in degrees from 0 to 90 and decimal minutes (8 decimal digits)
5	N/S	a	'N' for Northern hemisphere latitude, 'S' for Southern hemisphere latitude
6	Longitude	dddmm.mmmmmmmm	INS longitude in degrees from 0 to 180 and decimal minutes (8 decimal digits)
7	E/W	a	'E' for Eastern longitude, 'W' for Western longitude
8	Pos. Quality	x	Position quality indicator using INS standard deviation (see Quality indicators)
9	SV used	xx	Number of satellites used in solution 2 digits (latest received GNSS fix)
10	HDOP	z.z	Horizontal Dilution of Precision (1 decimal digit) (see Quality indicators)
11	Height	EHTx.xxx	Height above ellipsoid with 3 decimal digits and must start with EHT
12	Height Unit	M	Always M: Height above ellipsoid measured in meters
13	Check sum	*cs	Xor of all previous bytes except
14	End of frame	<CR><LF>	Carriage return and line feed

Note: The position quality and DOP are computed using the INS position estimated standard deviation. It is not related to the GNSS solution directly.

4. Third Party Logs description

In order to allow straightforward integration with third party equipments and software, SBG Systems inertial navigation systems handle several third party logs such as TSS1, Simrad or even Ixblue protocols.

4.1. SBG_ECOM_CLASS_LOG_THIRD_PARTY_0

This class contains Third party output logs.

4.1.1. Third Party Logs Overview

Name (MSG ID)	Description
SBG_ECOM_THIRD_PARTY_TSS1 (0x00)	Latitude, Longitude, Altitude, Quality indicator
SBG_ECOM_THIRD_PARTY_KVH (0x01)	KVH Roll, Pitch and heading output
SBG_ECOM_THIRD_PARTY_PD0 (0x02)	Teledyne RDI PD0 message
SBG_ECOM_THIRD_PARTY_SIMRAD_1000 (0x03)	Konsberg EM1000 proprietary frame that outputs Roll, Pitch and Heading
SBG_ECOM_THIRD_PARTY_SIMRAD_3000 (0x04)	Konsberg EM3000 proprietary frame that outputs Roll, Pitch and Heading
SBG_ECOM_THIRD_PARTY_SEAPATH_B26 (0x05)	Konsberg Seapth Binary Log 26 used for MBES FM mode
SBG_ECOM_THIRD_PARTY_DOLOG_HRP (0x06)	DOLOG Heading, Roll, Pitch and HRP rates
SBG_ECOM_THIRD_PARTY_AHRS_500 (0x07)	Crossbow AHRS-500 attitude, rate, acceleration and status
SBG_ECOM_THIRD_PARTY_ADA_01 (0x08)	ADA specific Data Packet with IMU/INS/Status data
SBG_ECOM_THIRD_PARTY_AT_ITINS (0x09)	Cobham AVIATOR UAV 200 AT command

4.1.2. Third Party Logs Availability

The table below details the Third Party logs availability according to a specific product family and product type. For example, if you have an ELLIPSE-E-G4A2-B1, you should refer to the column 'E' in the main column 'ELLIPSE'. Each available message is indicated with the ● symbol.

	ELLIPSE					High Performance INS					
Name (Message ID)	I	A	E	N	D	A	E	N	D	M	U
SBG_ECOM_THIRD_PARTY_TSS1 (0x00)		●	●	●	●	●	●	●	●	●	●
SBG_ECOM_THIRD_PARTY_KVH (0x01)		●	●	●	●						
SBG_ECOM_THIRD_PARTY_PD0 (0x02)							●	●	●		●
SBG_ECOM_THIRD_PARTY_SIMRAD_1000 (0x03)						●	●	●	●	●	●
SBG_ECOM_THIRD_PARTY_SIMRAD_3000 (0x04)						●	●	●	●	●	●
SBG_ECOM_THIRD_PARTY_SEAPATH_B26 (0x05)							●	●	●		●
SBG_ECOM_THIRD_PARTY_DOLOG_HRP (0x06)						●	●	●	●	●	●
SBG_ECOM_THIRD_PARTY_AHRS_500 (0x07)		●	●	●	●						
SBG_ECOM_THIRD_PARTY_ADA_01 (0x08)			●	●	●						
SBG_ECOM_THIRD_PARTY_AT_ITINS (0x09)			●	●	●						

4.1.3. TSS1

Name in sbgECom convention (msg ID): **SBG_ECOM_THIRD_PARTY_TSS1 (0x00)**

Proprietary log used for marine survey applications that provides heave, roll, pitch, as well as sway and heave accelerations.

This log is affected by the heave measurement point configured for each output interface. You can thus output a TSS1 frame for the main heave measurement point on the Port A and an other TSS1 frame on the Port B that measures the heave at the second monitoring point.

Warning: The TSS1 frame uses different conventions for Heave measurements. In this frame, Sway is when expressed positive left and heave is positive up.

Frame format

```
:XXAAAASMHHHHQMRRRRSMPPPP<CR><LF>
```

Field	Description
:	Start character
XX	Sway acceleration (hex value), in 3.835 cm/s², with a range from zero to 9.81 m/s²
AAAA	Vertical acceleration (hex value - 2's complement), in 0.0625 cm/s², with a range of -20.48 to +20.48 m/s²
S	Space character
M	Space if positive; minus if negative
HHHH	Heave measurement (ASCII value), in centimeters, with a range of -99.99 to +99.99 meters
Q	Status flag character (see table below)
M	Space if positive; minus if negative
RRRR	Roll, in units of 0.01 degrees (ex: 1000 = 10°), with a range of -99.99° to +99.99°
S	Space character
M	Space if positive; minus if negative
PPPP	Pitch, in units of 0.01 degrees (ex: 1000 = 10°), with a range of -99.99° to +99.99°
<CR><LF>	Carriage return, Line feed

TSS1 status flags

This flag is used to output status on algorithms used to compute the heave data. The Inertial System can use heading and velocity aiding data to improve the heave quality dramatically during ship maneuvers.

Value	Description
U	Unaided mode and stable measurements.
u	Unaided mode but unstable heave data.
G	Velocity aided mode and stable measurements.
g	Velocity aided mode but unstable data.
H	Heading aided mode and stable measurements.
h	Heading aided mode but unstable data.
F	Both velocity and heading aided mode and stable measurements.
f	Both velocity and heading aided mode but unstable measurements.

Frame example

```
:1A4770 -0016H 0429 -0680<CR><LF>
```

You can find below the explanation of each field:

- XX = 1A, Sway acceleration, which is 0.9971 $m.s^{-2}$
  (0x1A (hex) = 26 (decimal), multiplied by 0.03835 $m.s^{-2}$ yields to 0.9971 $m.s^{-2}$)
- AAAA = 4770, Heave acceleration, which is 11.43 $m.s^{-2}$
  (0x4770 (hex) = 18288 (decimal), multiplied by 0.000625 $m.s^{-2}$ yields to 11.43 $m.s^{-2}$)
- S = (space)
- M = (minus), meaning following heave value is negative
- HHHH = 0016, Heave value, which is 16 cm (-16 cm based on the M value)
- Q = H, status flag, which is stable heading aided mode
- M = (space), meaning following roll value is positive
- RRRR = 0429, roll, which is 4.29°
- S = (space)
- M = (minus), meaning following pitch value is negative
- PPPP = 0680, pitch, which is 6.80°

4.1.4. KVH

Name in sbgECom convention (msg ID): **SBG_ECOM_THIRD_PARTY_KVH (0x01)**

This format is very basic and provides pitch, roll, heading, and heading rate.

```
%pitch,roll,heading,headingrate<CR><LF>
```

Field	Name	Format	Description
0	%	string	Sentence Identifier
1	pitch	ddd	Pitch angle in tenths of degrees. Divide by 10 to get pitch in °. Positive when bow up
2	roll	ddd	Roll angle in tenths of degrees. Divide by 10 to get roll in °. Positive when port side down – **Warning**; Opposite sign of usual SBG Systems convention.
7	heading	ddd	Heading in tenths of degrees. From 0 to 3600.
8	End of frame	<CR><LF>	End of frame : carriage return and line feed.

Example frame

```
%10,-5,3489<CR><LF>
```

4.1.5. Teledyne RDI - PD0

Name in sbgECom convention (msg ID): **SBG_ECOM_THIRD_PARTY_PD0 (0x02)**

The PD0 binary frame is a Teledyne RDI proprietary log that output DVL data such as bottom tracking, water tracking and water profiling data. The INS can output this log only if valid PD0 frames are sent by a DVL to the INS. The PD0 frame is not altered or completed by the INS and is just forwarded as it is.

This frame is very useful to store into the internal datalogger or output in real time water profiling data that can be parsed with a third party software. Please refer to the Teledyne RDI documentations for a detailed PD0 frame definition.

4.1.6. Simrad EM1000 & Simrad EM3000

Name in sbgECom convention (msg ID): **`SBG_ECOM_THIRD_PARTY_SIMRAD_1000 (0x03)`**, **`SBG_ECOM_THIRD_PARTY_SIMRAD_3000 (0x04)`**

Proprietary binary log from Kongsberg used to input attitude and heave data to an echo sounder. The Simrad attitude log provides roll, pitch, heading as well as heave measurements.

This log is affected by the heave measurement point configured for each output interface. You can thus output a Simrad 1000/3000 frame for the main heave measurement point on the Port A and an other Simrad 1000/3000 frame on the Port B that measures the heave at the second monitoring point.

This binary log is 10 bytes long and all fields are stored in Little Endian (LSB first). Signed data uses the 2's complement representation. A resolution of 0.01° is used for roll, pitch and heading angles and 1 cm for heave measurements.

The only difference between Simrad 1000 and 3000 frames is the first byte that is used as a synchronization byte for Simrad 1000 and as a status flag for Simrad 3000.

Warning: The Simrad 1000/3000 frame uses different conventions for heave measurements. In this frame, heave is positive up.

Frame format

Field	Description	Unit	Format	Size	Offset
Sync Byte 1 / Status	Sync Byte 1 = 0x00 or Status flag for EM3000	-	uint8	1	0
Sync Byte 2	Sync Byte 2 = 0x90	-	uint8	1	1
Roll	Roll is positive with port side up ±179.99°	0.01°	int16	2	2
Pitch	Pitch is positive with bow up ±89.99°	0.01°	int16	2	4
Heave	Heave measurement positive up ±99.99 meters	cm	int16	2	6
Heading	Heading is positive clockwise [0 to 359.99°]	0.01°	int16	2	8
				Total size	10

Simrad 3000 status flags

This flag is used to output status on algorithms used to compute the orientation and heave data. This flag is only available for the Simrad 3000 protocol. For Simrad 1000 outputs, the first byte is used as a synchronization byte and is always set to 0x00.

Value	Description
0x90	Valid measurements with full accuracy.
0x91	Valid measurements with reduced accuracy (unaided mode).
0x9A	Invalid measurements, the device is aligning and/or the heave filter is not settled.
0xA0	Report an error with the motion sensor.

4.1.7. Konsberg Seapth Binary Log 26

Name in sbgECom convention (msg ID): **SBG_ECOM_THIRD_PARTY_SEAPATH_B26 (0x05)**

Proprietary binary log from Kongsberg Seapth that provides INS position, velocity, attitude, heave and rotation rates. It is used mainly with Konsberg MBES to support the FM mode.

This binary log is 52 bytes long and all fields are stored in Big Endian (MSB first). Signed data uses the 2's complement representation.

Frame format

Field	Description	Unit	Format	Size	Offset
Header 1	First header value, always 0xAA	-	uint8	1	0
Header 2	Second header value, always 0x55	-	uint8	1	1
Time, Seconds	UTC Time in seconds that dates INS data	Second	int32	4	2
Time, Fraction of Sec.	UTC Time fraction of second that dates INS data	0.0001 second	uint16	2	6
Latitude	INS latitude, positive North of the Equator.	2^{30} = 90 degrees	int32	4	8
Longitude	INS longitude, positive East of Greenwich.	2^{30} = 90 degrees	int32	4	12
Height	INS altitude above ellipsoid.	Centimeters	int32	4	16
Real time heave	Heave positive down.	Centimeters	int16	2	20
North velocity	INS North velocity, positive North	cm/s	int16	2	22
East velocity	INS East velocity, positive East	cm/s	int16	2	24
Down velocity	INS Down velocity, positive Down	cm/s	int16	2	26
Roll	Vessel roll angle, positive with port side up	2^{14} = 90 degrees	int16	2	28
Pitch	Vessel pitch angle, positive with bow up	2^{14} = 90 degrees	int16	2	30
Heading	Vessel true heading, positive clockwise [0 to 359.99°]	2^{14} = 90 degrees	uint16	2	32
Roll rate	Vessel roll rotation rate, positive with port side up	2^{14} = 90 °/s	int16	2	34
Pitch rate	Vessel pitch rotation rate, positive with bow up	2^{14} = 90 °/s	int16	2	36
Yaw rate	Vessel yaw rotation rate, positive clockwise	2^{14} = 90 °/s	int16	2	38
Delayed Heave Time, Seconds	UTC Time in seconds that dates the delayed heave information	Second	int32	4	40
Delayed Heave Time, Fraction of Sec.	UTC Time fraction of the second that dates the delayed heave information	0.0001 second	uint16	2	44
Delayed Heave	Delayed heave, positive down	Centimeters	int16	2	46
Status Word	Device status bitmask	-	uint16	2	48
Checksum		-	uint16	2	50
				Total size	52

Time Format

The UTC time is divided in an integer seconds part and a fractional second part. The integer seconds part of time is an UNIX time counted from 1970-01-01 UTC.

Checksum

Checksum is calculated as a 16-bit Block Cycle Redundancy Check of all bytes but not including the header and checksum. The CRC algorithm is using the polynom 0x8408 with an initial value of 0xFFFF.

Seapth Binary 26 Status Word

The Status Word consists of 16 single bit flags numbered from 0 to 15, where 0 position is the least significant bit. If the bit is set (1 value), it means:

Bit Position	Description
0	Reduced horizontal position and velocity performance.
1	Invalid horizontal position and velocity data.
2	Reduced heave and vertical velocity performance.
3	Invalid heave and vertical velocity data.
4	Reduced roll and pitch performance.
5	Invalid roll and pitch data
6	Reduced heading performance
7	Invalid heading data
8	Invalid delayed heave data

4.1.8. DOLOG HRP message

Name in sbgECom convention (msg ID): **SBG_ECOM_THIRD_PARTY_DOLOG_HRP (0x06)**

Proprietary binary message providing Euler angles (roll, pitch, heading) and Euler angle rates.

This binary log is 16 bytes long and all fields are stored in Big Endian (MSB first). Signed data uses the 2's complement representation.

Frame format

Field	Description	Unit	Format	Size	Offset
Header	Start of sentence – always 0x02	-	uint8	1	0
Status	Status bitmask, please refer to table below for details	-	uint8	1	1
Heading	Heading angle [0° to 359.99°]	2^{15} = 180°	uint16	2	2
Roll	Roll angle. positive when port side up [-90 to 89.99°]	2^{15} = 90°	int16	2	4
Pitch	Pitch angle. Positive when bow up [-90 to 89.99°]	2^{15} = 90°	int16	2	6
Heading Rate	Heading rate – [-45 to +44.99°/s]. Positive when heading increases	2^{15} = 45°/s	int16	2	8
Roll Rate	Roll rate – [-45 to +44.99°/s]. Positive when roll increases	2^{15} = 45°/s	int16	2	10
Pitch Rate	Pitch rate – [-45 to +44.99°/s]. Positive when pitch increases	2^{15} = 45°/s	int16	2	12
Checksum	Negative sum of all the bytes before checksum, ignoring overflow	-	uint8	1	14
Footer	End of frame byte – always 0x03	-	uint8	1	15
				Total size	16

Status bitmask definition

The Status Word consists of an 8 bit integer with various single bit flags numbered from 0 to 7, where 0 position is the least significant bit. If the bit is set (1 value), it means:

Bit Position	Description
0	Set if the Heading, Pitch & Roll measurements are valid (based on user defined thresholds).
1	Not Used
2	Not Used
3	Not Used
4	Set if the device is aligned and ready to deliver full accuracy data
5	Set if an error has been detected on the IMU (accelerometers or gyroscopes)
6	Not Used
7	Not Used

4.1.9. Crossbow AHRS-500 Data Packet

Name in sbgECom convention (msg ID): **SBG_ECOM_THIRD_PARTY_AHRS_500 (0x07)**

This proprietary binary message provides a compatibility with Crossbow AHRS-500 series. This data packet outputs the AHRS attitude (roll, pitch, yaw), the rotation rates, the accelerations as well as device status.

All values are stored in Big Endian (MSB first then LSB) and encoded on 16 bits signed integer in 2's complement representation. The status indications have to be converted to comply with SBG Systems AHRS/INS technology so please take care to test behavior differences.

Each data packet begins with a two-bytes header (0xAA 0x55) and a two-bytes checksum is used to ensure data integrity.

The checksum is computed using the following guidelines:

- Sum all packet contents except the header and checksum bytes
- Mod the sum by 0xFFFF to get the checksum

Frame format

Field	Description	Resolution	Unit	Format	Size	Offset
Header #1	First header char, always - 0xAA	-	-	uint8	1	0
Header #2	Second header char, always - 0x55	-	-	uint8	1	1
Roll	Roll angle [-180 to 180°]	$180/2^{15}$	°	int16	2	2
Pitch	Pitch angle [-90 to 90°]	$180/2^{15}$	°	int16	2	4
Heading	Heading angle [-180 to 180°]	$180/2^{15}$	°	int16	2	6
Rate X	X body rotation rate [-1200 to +1200°/s]	$1200/2^{15}$	°/s	int16	2	8
Rate Y	Y body rotation rate [-1200 to +1200°/s].	$1200/2^{15}$	°/s	int16	2	10
Rate Z	Z body rotation rate [-1200 to +1200°/s]	$1200/2^{15}$	°/s	int16	2	12
Accel X	X body acceleration [-15 to +15g] – With 1g = 9.80 m.s^-2	$15/2^{15}$	G	int16	2	14
Accel Y	Y body acceleration [-15 to +15g] – With 1g = 9.80 m.s^-2	$15/2^{15}$	G	int16	2	16
Accel Z	Z body acceleration [-15 to +15g] – With 1g = 9.80 m.s^-2	$15/2^{15}$	G	int16	2	18
Model #	Model number, always set to 226	-	-	uint16	2	20
Status BIT	Device status, please refer to Table 43: Status BIT mask definition	-	-	uint16	2	22
Checksum	Sum all bytes except headers (0xAA, 0x55) and mod by 0xFFFF	-	-	uint16	2	24
				Total size		26

Note: SBG Systems products and AHRS-500 are using the same North East Down frame definition. Euler angles, rotation rates and accelerations are using the same conventions as sbgECom logs.

SBG SYSTEMS

Status BIT definitions

Several status BIT are returned to monitor the health of the product. A bit set to 1 means there is an error or a performance degradation. A bit set to 0 is used to report normal / optimal operations.

Please keep in mind that Crossbow AHRS-500 status can't be directly transposed to SBG Systems products due to architecture differences. As a result, in this table, we also describe the expected behavior and conversions done between SBG Systems status and Crossbow ones.

Bit	Name	Description
0	Hard Failure	0: The unit is working correctly 1: A fatal error has been detected and the unit data shouldn't be used
1	Soft Failure	Not implemented, always set to 0
2	Reserved	Reserved, always set to 0
3	Power Failure	Not implemented, always set to 0
4	Comm Error	0: Port A serial communication is correctly working with no saturation 1: Port A serial communication has errors or is saturated
5	Reboot Detect	Not implemented, always set to 0
6	Calibration Failure	0: Sensor calibration table is valid and thus sensor data is valid 1: Sensor calibration table is corrupted and sensor data is invalid
7	External Aiding	0: Receiving and using external air data to aid the AHRS/INS solution 1: External air data not present or not used in the AHRS/INS solution
8	Coordinated Turn	Not implemented, always set to 0
9	Reserved	Reserved, always set to 0
10	Algorithm Status	0: Roll and Pitch are valid and device alignment is done 1: Solution not ready, alignment in progress or roll/pitch is invalid
11	Mag Align Status	0: Heading is valid and aligned 1: Heading is not valid such as free drifting or not aligned yet
12	Reserved	Reserved, always set to 0
13	Mag Calibration	0: Magnetometers are valid and used in the AHRS/INS solution 1: Magnetometers are not used in AHRS/INS solution and probably not well calibrated
14	Reserved	Reserved, always set to 0
15	Remote Mag	Not implemented, always set to 0

Table 43: Status BIT mask definition

4.1.10. ADA Data Packet

Name in sbgECom convention (msg ID): **SBG_ECOM_THIRD_PARTY_ADA_01 (0x08)**

This proprietary binary message provides all INS, IMU and status data in one single compact frame. This message uses only integers representations with specific scaling making processing time efficient for low end processing platforms.

All values are stored in Big Endian (MSB first then LSB) and encoded on 16/32 bits signed integer in 2's complement representation or unsigned integers.

The status indications are exactly the same as the ones defined in sbgECom log messages. The exact same behavior is thus expected between standard sbgECom logs and this log.

Each data packet begins with a two-bytes header (0xAA 0x5A) and a two-bytes checksum is used to ensure data integrity.

The checksum is computed using the following guidelines:

- Sum all packet contents except the header and checksum bytes
- Mod the sum by 0xFFFF to get the checksum

Frame format

Field	Description	Resolution	Unit	Format	Size	Offset
Header #1	First header char, always - 0xAA	-	-	uint8	1	0
Header #2	Second header char, always - 0x5A	-	-	uint8	1	1
Roll	INS Roll angle [-180 to 180°]	$180/2^{15}$	°	int16	2	2
Pitch	INS Pitch angle [-90 to 90°]	$180/2^{15}$	°	int16	2	4
Heading	INS Heading angle [-180 to 180°]	$180/2^{15}$	°	int16	2	6
Rate X	X body rotation rate [-1200 to +1200°/s]	$1200/2^{15}$	°/s	int16	2	8
Rate Y	Y body rotation rate [-1200 to +1200°/s].	$1200/2^{15}$	°/s	int16	2	10
Rate Z	Z body rotation rate [-1200 to +1200°/s]	$1200/2^{15}$	°/s	int16	2	12
Accel X	X body acceleration [-15 to +15g] – With 1g = 9.80 m.s^-2	$15/2^{15}$	G	int16	2	14
Accel Y	Y body acceleration [-15 to +15g] – With 1g = 9.80 m.s^-2	$15/2^{15}$	G	int16	2	16
Accel Z	Z body acceleration [-15 to +15g] – With 1g = 9.80 m.s^-2	$15/2^{15}$	G	int16	2	18
Temperature	Internal INS temperature [-50 to +100°]	$100/2^{15}$	°C	int16	2	20
Velocity N	INS North Velocity [-1500 to +1500°/s]	$1500/2^{31}$	m/s	int32	4	24
Velocity E	INS East Velocity [-1500 to +1500°/s]	$1500/2^{31}$	m/s	int32	4	28
Velocity D	INS Down Velocity [-1500 to +1500°/s]	$1500/2^{31}$	m/s	int32	4	32
Latitude	INS Latitude positive North [-90 to +90°]	$180/2^{31}$	°	int32	4	36
Longitude	INS Longitude positive East [-180 to +180°]	$180/2^{31}$	°	int32	4	40
Altitude	INS Altitude above MSL positive upward [-100000 to +100000 m]	$100000/2^{31}$	m	int32	4	44
Roll Std	INS Roll accuracy 1σ standard deviation [0 to +180°]	$180/2^{15}$	°	uint16	2	46
Pitch Std	INS Pitch accuracy 1σ standard deviation [0 to +180°]	$180/2^{15}$	°	uint16	2	48
Heading Std	INS Heading accuracy 1σ standard deviation [0 to +180°]	$180/2^{15}$	°	uint16	2	50
Velocity N Std.	INS North Velocity accuracy 1σ standard deviation [0 to +200 m/s]	$100/2^{15}$	m/s	uint16	2	52
Velocity E Std.	INS East Velocity accuracy 1σ standard deviation [0 to +200 m/s]	$100/2^{15}$	m/s	uint16	2	54
Velocity D Std.	INS Down Velocity accuracy 1σ standard deviation [0 to +200 m/s]	$100/2^{15}$	m/s	uint16	2	56
Latitude Std.	INS Latitude accuracy 1σ standard deviation [0 to +3276 m]	0.05	m	uint16	2	58
Longitude Std.	INS Longitude accuracy 1σ standard deviation [0 to +3276 m]	0.05	m	uint16	2	60
Altitude Std.	INS Altitude accuracy 1σ standard deviation [0 to +3276 m]	0.05	m	uint16	2	62
Time stamp	Free running time since INS powered up	-	µs	uint32	4	64
General Status	See GENERAL_STATUS definition	-	-	uint16	2	68
Com Status	See COM_STATUS definition	-	-	uint32	4	70
Aiding Status	See AIDING_STATUS definition	-	-	uint32	4	74
Up Time	INS system up time since power on in seconds	-	s	uint32	4	78
IMU Status	See IMU_STATUS definition	-	-	uint16	2	82
Solution Status	See SOLUTION_STATUS definition	-	-	uint32	4	84
Reserved	Reserved bytes for future use	-	-	-	10	88
Checksum	Sum all bytes except headers (0xAA, 0x5A) and mod by 0xFFFF	-	-	uint16	2	98
				Total size		100

4.1.11. Cobham AT_ITINS

Name in sbgECom convention (msg ID): **SBG_ECOM_THIRD_PARTY_AT_ITINS (0x09)**

This proprietary ASCII message provides all navigation and orientation data required by the AVIATOR UAV 200 Satcom terminal to ensure correct steering towards the satellite.

The message starts with AT_ITINS= and ends with a <CR><LF>. Each parameter is separated by a comma and the full message length is always less or equals to 128 chars.

Example

```
AT_ITINS=48.9102,2.1677,66,112,2020-06-17/14:10:15,0,0,0,-1.6,0.2,0.1,127.3,126.3,0.1,0.2,0.0\r\n
```

Frame format

Field	Description	Decimal digits	Unit
AT_IT_INS=	AT-Command header	-	-
latitude	INS latitude positive North [-90 to +90]	4	°
longitude	INS longitude positive East [-180 to +180]	4	°
baroAltitude	Barometric altitude [-500 to +25000]	0	m
height	INS altitude above ellipsoid [-500 to +25000]	0	m
UTC Time	UTC Time as a string: *yyyy-mm-dd/hh:mm:ss*	-	-
velocityNorth	INS north velocity positive north [-500 to +500]	0	m/s
velocityEast	INS east velocity positive east [-500 to +500]	0	m/s
gndSpeed	INS ground 2D velocity [0 to +500]	0	m/s
trackAngle	INS track course positive clockwise [-180 to +180]	1	°
roll	INS roll angle positive right up [-180 to +180]	1	°
pitch	INS pitch angle positive nose up [-180 to +180]	1	°
heading	INS heading positive clockwise [-180 to +180]	1	°
magHeading	Magnetic heading positive clockwise [-180 to +180]	1	°
rollRate	Roll rate positive right up [-90 to +90]	1	°/s
pitchRate	Pitch rate positive nose up [-90 to +90]	1	°/s
yawRate	Yaw rate positive clockwise [-90 to +90]	1	°/s

Note 1: Roll, pitch, yaw rate are body rates equivalent to gyroscope X,Y,Z readings but compensated for bias and earth rotations

Note 2: Invalid or not available fields are represented using a zero value.

5. Input protocols

Several external equipment can be connected to an Inertial Navigation System to enhance it's accuracy by providing aiding information.

The most common aiding equipment for an INS is an external GNSS receiver. An external GNSS receiver will typically provide aiding information using NMEA standard protocol. However, for best accuracy and robustness, SBG Systems tries to support native binary protocols from leading third party equipment manufacturers.

In this section, you will find details about third party equipment and protocols support. Each product will support different aiding equipment and protocols so please refer to the section Input Protocols Support below for input protocols availability for your specific device.

5.1.1. Input Protocols Support

The table below details the support of Third Party equipment and protocols according to a specific product family and product type. For example, if you have an ELLIPSE-E-G4A2-B1, you should refer to the column 'E' in the main column 'ELLIPSE'. Each available message is indicated with the ● symbol.

	ELLIPSE					High Performance INS					
Name (Message ID)	I	A	E	N	D	A	E	N	D	M	U
sbgECom Input Protocol			●	●	●						
NMEA Protocol			●	●	●	●	●	●	●	●	●
UBX protocol			●	●	●						
Novatel Binary Protocol			●	●	●	●	●	●	●	●	●
Septentrio SBF binary protocol			●	●	●	●	●	●	●	●	●
Trimble GSOF binary protocol						●	●	●	●	●	●
Teledyne RDI PD0 protocol						●	●	●	●	●	●
Teledyne RDI PD6 protocol						●	●	●	●	●	●
Crossbow AHRS-500 Air Data			●	●	●						

5.1.2. sbgECom Input Protocol

Some aiding information such as external velocity or altitude (airdata), can be sent to the INS using the specific binary sbgECom protocol.

The sbgECom protocol is compact and secured and as such is ideal to inject in real time aiding information. The following sbgECom logs are accepted by the device according to aiding assignment configuration:

- SBG_ECOM_LOG_AIR_DATA (36) is used by the AirData module to inject an external barometric altitude as well as true airspeed (please refer to Operating Handbooks for more information)

5.1.3. NMEA Protocol

This protocol is used as GPS aiding data in a read only mode.

Currently several sentences are required for proper operation:

- GGA is used to handle position aiding
- GST is used for position accuracy
- RMC is used to handle velocity input
- HDT is used to get true heading from dual antenna systems.
- ZDA is used for UTC synchronization and it is usually sent at 1 Hz

Note: Please refer to the NMEA GNSS Integration Operating Handbook for more details.

5.1.4. UBX protocol

The Ublox UBX binary protocol can be used to provide best performance when connecting an external Ublox GPS/GNSS receiver to the INS. UBlox receivers are miniature and low cost GPS receivers so this protocol is only supported for ELLIPSE series.

The following messages are handled:

- NAV PVT is used to handle the full Position, Velocity and Time solution (required)
- NAV SAT is used to get advanced signal tracking details such as constellations in use (optional).

5.1.5. Novatel Binary Protocol

The Novatel protocol (binary form) can be used to provide best performance when connecting an external Novatel GNSS receiver to the INS.

Currently several sentences are handled using Novatel Binary protocol:

- BESTPOS is used to handle position aiding.
- PSRXYZ is used to handle Doppler velocity with associated accuracy.
- HEADING is used to get true heading from dual antenna systems.
- TIME is used for UTC synchronization and is usually sent at 1 Hz

The following messages should be sent as well for raw data output (post-processing):

The required configuration for this purpose is the following:

- RANGECMPB ontime 1
- RAWEPHEMB onchanged
- GLOEPHEMERISB onchanged
- HEADINGB onnew

Note: Please refer to the Novatel GNSS Integration Operating Handbook for more details.

5.1.6. Septentrio SBF binary protocol

The INS implements Septentrio SBF binary protocol for GPS data aiding input. The following messages are used for Septentrio → INS integration:

- PVTGeodetic: For position and velocity input
- PosCovGeodetic: For position advanced error management
- VelCovGeodetic: For velocity advanced error management
- AttEuler: Orientation for dual antenna systems
- AttCovEuler: Orientation accuracy
- ReceiverTime: Timing data
- xPPSOffset: Timing data
- RawData for post processing

Note: Please refer to the Septentrio GNSS Integration Operating Handbook for more details.

5.1.7. Trimble GSOF binary protocol

The Trimble General Serial Output Format (GSOF) can provide all needed information for the INS to compute a real time navigation solution. For post processing, the device supports RT-27 frame parsing and storage.

The following GSOF messages are supported:

- Time: For the time of the navigation solution
- LLH: For the latitude, longitude and altitude above ellipsoid
- Velocity: For the 3D velocity information
- UTC: For UTC time information
- Attitude: For dual antenna based heading
- All SV Detail: For space vehicle information

Note: Please refer to the Trimble GNSS Integration Operating Handbook for more details.

5.1.8. Teledyne RDI PD0 protocol

The PD0 binary frame is a Teledyne RDI proprietary log that output DVL data such as bottom tracking, water tracking and water profiling data.

It is accepted as input for DVL aiding purpose.

5.1.9. Teledyne RDI PD6 protocol

The PD6 ascii frame is a Teledyne RDI proprietary log that output DVL data such as bottom tracking and water tracking.

It is accepted as input for DVL aiding purpose.

5.1.10. Crossbow AHRS-500 Air Data

SBG Systems INS can accept Air Data aiding information such as true airspeed and barometric altitude to enhance the INS performance in case of GNSS outages. It also greatly improves the pitch accuracy for high performance aircraft if no GNSS aiding is available.

The AHRS-500 defines an external aiding packet used to input an external GNSS, airspeed and barometric altitude to the INS. However, currently, only the airspeed and barometric altitude information are supported and used to improve the AHRS / INS solution.

The external aiding packet should be sent at least at 1 Hz to be used by the device. You can use higher update rates and up to 5 Hz but make sure the serial interface is not saturated.

Frame format

Field	Description	Resolution	Unit	Format	Size	Offset
Header #1	First header char, always - 0xAA	-	-	uint8	1	0
Header #2	Second header char, always - 0x55	-	-	uint8	1	1
GNSS Latitude	Not used, leave 0xFFFFFFFF	-	-	int32	4	2
GNSS Longitude	Not used, leave 0xFFFFFFFF	-	-	int32	4	6
GNSS Altitude	Not used, leave 0xFFFFFFFF	-	-	int32	4	10
GNSS Ground Speed	Not used, leave 0xFFFF	-	-	uint16	2	14
GNSS Track	Not used, leave 0xFFFF	-	-	int16	2	16
GNSS Vertical Velocity	Not used, leave 0xFFFF	-	-	int16	2	18
True Airspeed	True Airspeed in the range of 0 - 2048 knots. Invalid indicated by a value of 0xFFFF.	$2048/2^{15}$	knots	uint16	2	20
Baro Altitude	Pressure Altitude in the range of ±131072ft. Resolution is 17 bits. Invalid indicated by a value of 0xFFFFFFFF.	$131072/2^{17}$	feet	int32	4	22
Baro Vertical Velocity	Not used, leave 0xFFFF	-	-	int16	2	26
Magnetic Variation	Not used, leave 0xFFFF	-	-	int16	2	28
Checksum	Sum all bytes except headers (0xAA, 0x55) and mod by 0xFFFF	-	-	uint16	2	30
				Total size		**32**

6. CAN protocol specifications

6.1. Introduction

The CAN protocol implements almost all information returned by the sbgECom serial protocol. Even if the two protocols are not the same, they share as much definitions and data structures as possible.

6.2. Conventions and principles

6.2.1. Status and enum definitions

For example, all status, enum, bitmask used in the CAN messages are the same as the ones used in the sbgECom serial protocol. For the sake of simplicity and maintainability, please refer to sbgECom serial logs to get status, bitmask and enums definitions.

6.2.2. Value scaling and units

In order to be as compatible with third party CAN software as possible, each parameter can be converted with a simple offset and scale factor. For example, if a parameter has a scale factor of 10^{-2}, it means that the value returned by the device should be multipled by 10^{-2} to get the correct unit specified in the message.

6.2.3. Message Identification

Every CAN message uses a unique identifier encoded on 11 bits for a CAN 2.0A standard message or on 29 bits for a CAN 2.0B extended message. In order to avoid incompatibilities with other materials, every CAN message id must be individually defined or even disabled using the following special id:

SBG_DISABLED_FRAME 0x000003FF

6.3. Output Logs

6.3.1. CAN Messages Overview

The following list, provides a quick overview of all available CAN messages. It briefly describle which parameters are contained in each CAN message.

Name (log ID)	Description
SBG_ECAN_MSG_STATUS_01 (0x100) SBG_ECAN_MSG_STATUS_02 (0x101) SBG_ECAN_MSG_STATUS_03 (0x102)	General status such as clock, com aiding, solution, heave
SBG_ECAN_MSG_UTC_0 (0x110) SBG_ECAN_MSG_UTC_1 (0x111)	Provides UTC time reference
SBG_ECAN_MSG_IMU_INFO (0x120) SBG_ECAN_MSG_IMU_ACCEL (0x121) SBG_ECAN_MSG_IMU_GYRO (0x122) SBG_ECAN_MSG_IMU_DELTA_VEL (0x123) SBG_ECAN_MSG_IMU_DELTA_ANGLE (0x124)	Includes IMU status, acc., gyro, temp delta speeds and delta angles values
SBG_ECAN_MSG_EKF_INFO (0x130) SBG_ECAN_MSG_EKF_QUAT (0x131) SBG_ECAN_MSG_EKF_EULER (0x132) SBG_ECAN_MSG_EKF_ORIENTATION_ACC (0x133)	Includes roll, pitch, yaw, or quaternion output and their accuracies on each axis

Name (log ID)	Description
SBG_ECAN_MSG_EKF_POS (0x134) SBG_ECAN_MSG_EKF_ALTITUDE (0x135) SBG_ECAN_MSG_EKF_POS_ACC (0x136) SBG_ECAN_MSG_EKF_VEL_NED (0x137) SBG_ECAN_MSG_EKF_VEL_NED_ACC (0x138) SBG_ECAN_MSG_EKF_VEL_BODY (0x139)	Position and velocities computed by the INS Kalman Filter. The positions are expressed in LLA and the velocities in either NED or INS body frame. Estimated standard deviation are also provided.
SBG_ECAN_MSG_AUTO_TRACK_SLIP_CURV (0x220)	These outputs are related to automotive applications with information such as slip angle or curvature radius.
SBG_ECAN_MSG_SHIP_MOTION_INFO (0x140) SBG_ECAN_MSG_SHIP_MOTION_0 (0x141) SBG_ECAN_MSG_SHIP_MOTION_1 (0x145) SBG_ECAN_MSG_SHIP_MOTION_2 (0x149)	Real time ship motion parameters such as surge, sway and heave
SBG_ECAN_MSG_SHIP_MOTION_HP_INFO (0x14A) SBG_ECAN_MSG_SHIP_MOTION_HP_0 (0x14B) SBG_ECAN_MSG_SHIP_MOTION_HP_1 (0x14C) SBG_ECAN_MSG_SHIP_MOTION_HP_2 (0x14D)	Delayed heave measurements.
SBG_ECAN_MSG_MAG_0 (0x150) SBG_ECAN_MSG_MAG_1 (0x151) SBG_ECAN_MSG_MAG_2 (0x152)	Magnetic data with associated accelerometer on each axis
SBG_ECAN_MSG_ODO_INFO (0x160) SBG_ECAN_MSG_ODO_VEL (0x161)	Provides odometer velocity
SBG_ECAN_MSG_AIR_DATA_INFO (0x162) SBG_ECAN_MSG_AIR_DATA_ALTITUDE (0x163) SBG_ECAN_MSG_AIR_DATA_AIRSPEED (0x164)	Barometric pressure & altitude True airspeed from a pitot tube
SBG_ECAN_MSG_DEPTH_INFO (0x166) SBG_ECAN_MSG_DEPTH_ALTITUDE (0x167)	Subsea water pressure and depth below surface
SBG_ECAN_MSG_GPS1_VEL_INFO (0x170) SBG_ECAN_MSG_GPS1_VEL (0x171) SBG_ECAN_MSG_GPS1_VEL_ACC (0x172) SBG_ECAN_MSG_GPS1_COURSE (0x173)	Velocity and course from primary GNSS receiver
SBG_ECAN_MSG_GPS1_POS_INFO (0x174) SBG_ECAN_MSG_GPS1_POS (0x175) SBG_ECAN_MSG_GPS1_POS_ALT (0x176) SBG_ECAN_MSG_GPS1_POS_ACC (0x177)	Positions from primary GNSS receiver
SBG_ECAN_MSG_GPS1_HDT_INFO (0x178) SBG_ECAN_MSG_GPS1_HDT (0x179)	True heading from primary dual antenna GNSS receiver
SBG_ECAN_MSG_GPS2_VEL_INFO (0x180) SBG_ECAN_MSG_GPS2_VEL (0x181) SBG_ECAN_MSG_GPS2_VEL_ACC (0x182) SBG_ECAN_MSG_GPS2_COURSE (0x183)	Velocity and course from secondary GNSS receiver
SBG_ECAN_MSG_GPS2_POS_INFO (0x184) SBG_ECAN_MSG_GPS2_POS (0x185) SBG_ECAN_MSG_GPS2_POS_ALT (0x186) SBG_ECAN_MSG_GPS2_POS_ACC (0x187)	Positions from secondary GNSS receiver
SBG_ECAN_MSG_GPS2_HDT_INFO (0x188) SBG_ECAN_MSG_GPS2_HDT (0x189)	True heading from secondary dual antenna GNSS receiver
SBG_ECAN_MSG_EVENT_INFO_A (0x200) SBG_ECAN_MSG_EVENT_TIME_A (0x201)	Event marker sent when a signal is detected on Sync In A pin
SBG_ECAN_MSG_EVENT_INFO_B (0x202) SBG_ECAN_MSG_EVENT_TIME_B (0x203)	Event marker sent when a signal is detected on Sync In B pin

Name (log ID)	Description
SBG_ECAN_MSG_EVENT_INFO_C (0x204) SBG_ECAN_MSG_EVENT_TIME_C (0x205)	Event marker sent when a signal is detected on Sync In C pin
SBG_ECAN_MSG_EVENT_INFO_D (0x206) SBG_ECAN_MSG_EVENT_TIME_D (0x207)	Event marker sent when a signal is detected on Sync In D pin
SBG_ECAN_MSG_EVENT_INFO_E (0x207) SBG_ECAN_MSG_EVENT_TIME_E (0x209)	Event marker sent when a signal is detected on Sync In E pin

6.3.2. Messages Availability

The table below details a CAN message availability according to a specific product family and product type. For example, if you have an ELLIPSE-E-G4A2-B1, you should refer to the column 'E' in the main column 'ELLIPSE'. Each available message is indicated with the • symbol.

	ELLIPSE					High Performance INS					
Name (Message ID)	I	A	E	N	D	A	E	N	D	M	U
SBG_ECAN_MSG_STATUS_01 (0x100) SBG_ECAN_MSG_STATUS_02 (0x101) SBG_ECAN_MSG_STATUS_03 (0x102)	•	•	•	•	•	•	•	•	•	•	•
SBG_ECAN_MSG_UTC_0 (0x110) SBG_ECAN_MSG_UTC_1 (0x111)	•	•	•	•	•	•	•	•	•	•	•
SBG_ECAN_MSG_IMU_INFO (0x120) SBG_ECAN_MSG_IMU_ACCEL (0x121) SBG_ECAN_MSG_IMU_GYRO (0x122) SBG_ECAN_MSG_IMU_DELTA_VEL (0x123) SBG_ECAN_MSG_IMU_DELTA_ANGLE (0x124)	•	•	•	•	•	•	•	•	•	•	•
SBG_ECAN_MSG_EKF_INFO (0x130) SBG_ECAN_MSG_EKF_QUAT (0x131) SBG_ECAN_MSG_EKF_EULER (0x132) SBG_ECAN_MSG_EKF_ORIENTATION_ACC (0x133)		•	•	•	•	•	•	•	•	•	•
SBG_ECAN_MSG_EKF_POS (0x134) SBG_ECAN_MSG_EKF_ALTITUDE (0x135) SBG_ECAN_MSG_EKF_POS_ACC (0x136) SBG_ECAN_MSG_EKF_VEL_NED (0x137) SBG_ECAN_MSG_EKF_VEL_NED_ACC (0x138) SBG_ECAN_MSG_EKF_VEL_BODY (0x139)			•	•	•		•	•	•		•
SBG_ECAN_MSG_AUTO_TRACK_SLIP_CURV (0x220)			•	•	•						
SBG_ECAN_MSG_SHIP_MOTION_INFO (0x140) SBG_ECAN_MSG_SHIP_MOTION_0 (0x141) SBG_ECAN_MSG_SHIP_MOTION_1 (0x145) SBG_ECAN_MSG_SHIP_MOTION_2 (0x149)		•	•	•	•	•	•	•	•	•	•
SBG_ECAN_MSG_SHIP_MOTION_HP_INFO (0x14A) SBG_ECAN_MSG_SHIP_MOTION_HP_0 (0x14B) SBG_ECAN_MSG_SHIP_MOTION_HP_1 (0x14C) SBG_ECAN_MSG_SHIP_MOTION_HP_2 (0x14D)						•	•	•	•	•	•
SBG_ECAN_MSG_MAG_0 (0x150) SBG_ECAN_MSG_MAG_1 (0x151) SBG_ECAN_MSG_MAG_2 (0x152)	•	•	•	•	•						
SBG_ECAN_MSG_ODO_INFO (0x160) SBG_ECAN_MSG_ODO_VEL (0x161)			•	•	•		•	•	•		
SBG_ECAN_MSG_AIR_DATA_INFO (0x162) SBG_ECAN_MSG_AIR_DATA_ALTITUDE (0x163) SBG_ECAN_MSG_AIR_DATA_AIRSPEED (0x164)			•	•	•						
SBG_ECAN_MSG_DEPTH_INFO (0x166) SBG_ECAN_MSG_DEPTH_ALTITUDE (0x167)											
SBG_ECAN_MSG_GPS1_VEL_INFO (0x170) SBG_ECAN_MSG_GPS1_VEL (0x171) SBG_ECAN_MSG_GPS1_VEL_ACC (0x172) SBG_ECAN_MSG_GPS1_COURSE (0x173)			•	•	•	•	•	•	•	•	•

	ELLIPSE					High Performance INS					
Name (Message ID)	I	A	E	N	D	A	E	N	D	M	U
SBG_ECAN_MSG_GPS1_POS_INFO (0x174) SBG_ECAN_MSG_GPS1_POS (0x175) SBG_ECAN_MSG_GPS1_POS_ALT (0x176) SBG_ECAN_MSG_GPS1_POS_ACC (0x177)			•	•	•	•	•	•	•	•	•
SBG_ECAN_MSG_GPS1_HDT_INFO (0x178) SBG_ECAN_MSG_GPS1_HDT (0x179)			•		•	•	•	•	•	•	•
SBG_ECAN_MSG_GPS2_VEL_INFO (0x180) SBG_ECAN_MSG_GPS2_VEL (0x181) SBG_ECAN_MSG_GPS2_VEL_ACC (0x182) SBG_ECAN_MSG_GPS2_COURSE (0x183)							•	•	•		•
SBG_ECAN_MSG_GPS2_POS_INFO (0x184) SBG_ECAN_MSG_GPS2_POS (0x185) SBG_ECAN_MSG_GPS2_POS_ALT (0x186) SBG_ECAN_MSG_GPS2_POS_ACC (0x187)							•	•	•		•
SBG_ECAN_MSG_GPS2_HDT_INFO (0x188) SBG_ECAN_MSG_GPS2_HDT (0x189)											
SBG_ECAN_MSG_EVENT_INFO_A (0x200) SBG_ECAN_MSG_EVENT_TIME_A (0x201)	•	•	•	•	•	•	•	•	•	•	•
SBG_ECAN_MSG_EVENT_INFO_B (0x202) SBG_ECAN_MSG_EVENT_TIME_B (0x203)		•	•	•	•	•	•	•	•	•	•
SBG_ECAN_MSG_EVENT_INFO_C (0x204) SBG_ECAN_MSG_EVENT_TIME_C (0x205)			•			•	•	•	•		
SBG_ECAN_MSG_EVENT_INFO_D (0x206) SBG_ECAN_MSG_EVENT_TIME_D (0x207)			•		•	•	•	•	•	•	•
SBG_ECAN_MSG_EVENT_INFO_E (0x207) SBG_ECAN_MSG_EVENT_TIME_E (0x209)						•	•	•	•	•	•

6.3.3. General Status output

These outputs combine all system status data, divided into six categories: General, Clock, Communications, Aiding, Solution and Heave. This log is useful for advanced status information.

Message name (ID)	SBG_ECAN_MSG_STATUS_01 (0x100)					
Field	Description	Scaling	Unit	Format	Size	Offset
TIME_STAMP	Time since sensor is powered up.	1	µs	uint32	4	0
GENERAL_STATUS	General status bit-mask.	-	-	uint16	2	4
CLOCK_STATUS	Clock status bit-mask.	-	-	uint16	2	6
				Total size		8

Message name (ID)	SBG_ECAN_MSG_STATUS_02 (0x101)					
Field	Description	Scaling	Unit	Format	Size	Offset
COM_STATUS	Com status bit-mask.	-	-	uint32	4	0
AIDING_STATUS	Aiding status bit-mask.	-	-	uint32	4	4
				Total size		8

Message name (ID)	SBG_ECAN_MSG_STATUS_03 (0x102)					
Field	Description	Scaling	Unit	Format	Size	Offset
SOLUTION_STATUS	Solution status bit-mask.	-	-	uint32	4	0
HEAVE_STATUS	Heave status bit-mask.	-	-	uint16	2	4
				Total size		6

STATUS definition

Please refer to the sbgECom serial log GENERAL_STATUS definition, CLOCK_STATUS definition, COM_STATUS definition, AIDING_STATUS definition, SOLUTION_STATUS definition and HEAVE_STATUS definition.

6.3.4. UTC time output

Provides UTC time reference. This frame also provides a time correspondence between the device TIME_STAMP value and the actual UTC Time. You thus have to use this frame if you would like to time stamp all data to an absolute UTC or GPS time reference.

Message name (ID)	SBG_ECAN_MSG_UTC_0 (0x110)					
Field	**Description**	**Scaling**	**Unit**	**Format**	**Size**	**Offset**
TIME_STAMP	Time since sensor is powered up.	1	µs	uint32	4	0
GPS_TOW	GPS Time of week.	1	ms	uint32	4	4
				Total size		8

Message name (ID)	SBG_ECAN_MSG_UTC_1 (0x111)					
Field	**Description**	**Scaling**	**Unit**	**Format**	**Size**	**Offset**
YEAR	Year within the century (e.g. '10' means 2010)	1	-	uint8	1	0
MONTH	Month in Year [1 ... 12]	1	-	uint8	1	1
DAY	Day in Month [1 ... 31]	1	-	uint8	1	2
HOUR	Hour in day [0 ... 23]	1	-	uint8	1	3
MIN	Minute in hour [0 ... 59]	1	-	uint8	1	4
SEC	Second in minute [0 ... 60] Note 60 is when a leap second is added.	1	s	uint8	1	5
MICRO_SEC	Microsecond within the current second	100	µs	uint16	2	6
				Total size		8

6.3.5. Inertial Data output

These CAN logs output the IMU calibrated accelerometers and gyroscopes values as well as delta angles and delta velocities that are computed using 1 kHz coning and sculling integrals.

Message name (ID)	SBG_ECAN_MSG_IMU_INFO (0x120)					
Field	Description	Scaling	Unit	Format	Size	Offset
TIME_STAMP	Time since sensor is powered up.	1	µs	uint32	4	0
IMU_STATUS	IMU Status bit-mask.	-	-	uint16	2	4
TEMPERATURE	IMU Temperature.	10^{-2}	°C	int16	2	6
				Total size		8

IMU_STATUS definition

Please refer to the sbgECom serial log IMU_STATUS definition.

Message name (ID)	SBG_ECAN_MSG_IMU_ACCEL (0x121)					
Field	Description	Scaling	Unit	Format	Size	Offset
ACCEL_X	Acceleration X	10^{-2}	$m.s^{-2}$	int16	2	0
ACCEL_Y	Acceleration Y	10^{-2}	$m.s^{-2}$	int16	2	2
ACCEL_Z	Acceleration Z	10^{-2}	$m.s^{-2}$	int16	2	4
				Total size		6

Message name (ID)	SBG_ECAN_MSG_IMU_GYRO (0x122)					
Field	Description	Scaling	Unit	Format	Size	Offset
GYRO_X	Rate of turn X	10^{-3}	$rad.s^{-1}$	int16	2	0
GYRO_Y	Rate of turn Y	10^{-3}	$rad.s^{-1}$	int16	2	2
GYRO_Z	Rate of turn Z	10^{-3}	$rad.s^{-1}$	int16	2	4
				Total size		6

Message name (ID)	SBG_ECAN_MSG_IMU_DELTA_VEL (0x123)					
Field	Description	Scaling	Unit	Format	Size	Offset
DELTA_VEL_X	Delta velocity X	10^{-2}	$m.s^{-2}$	int16	2	0
DELTA_VEL_Y	Delta velocity Y	10^{-2}	$m.s^{-2}$	int16	2	2
DELTA_VEL_Z	Delta velocity Z	10^{-2}	$m.s^{-2}$	int16	2	4
				Total size		6

Message name (ID)	SBG_ECAN_MSG_IMU_DELTA_ANGLE (0x124)					
Field	Description	Scaling	Unit	Format	Size	Offset
DELTA_ANGLE_X	Coning integration X	10^{-3}	$rad.s^{-1}$	int16	2	0
DELTA_ANGLE_Y	Coning integration Y	10^{-3}	$rad.s^{-1}$	int16	2	2
DELTA_ANGLE_Z	Coning integration Z	10^{-3}	$rad.s^{-1}$	int16	2	4
				Total size		6

6.3.6. EKF output

These logs output all parameters computed by the device Extended Kalman Filter such as the 3D attitude, velocity and inertially aided position.

6.3.6.1. General EKF info

Message name (ID)	SBG_ECAN_MSG_EKF_INFO (0x130)					
Field	Description	Scaling	Unit	Format	Size	Offset
TIME_STAMP	Time since sensor is powered up.	1	µs	uint32	4	0
				Total size		4

6.3.6.2. Orientation data

The EFK computed orientation data can be output using different representations. The 3D orientation can be represented using a normalized quaternion (q0, q1, q2, q3) or using so called Euler angles.

Message name (ID)	SBG_ECAN_MSG_EKF_QUAT (0x131)					
Field	Description	Scaling	Unit	Format	Size	Offset
Q0	Orientation quaternion, q0 component.	1/32767	-	int16	2	0
Q1	Orientation quaternion, q1 component.	1/32767	-	int16	2	2
Q2	Orientation quaternion, q2 component.	1/32767	-	int16	2	4
Q3	Orientation quaternion, q3 component.	1/32767	-	int16	2	6
				Total size		8

Message name (ID)	SBG_ECAN_MSG_EKF_EULER (0x132)					
Field	Description	Scaling	Unit	Format	Size	Offset
ROLL	Roll angle [-π to +π]	10^{-4}	rad	int16	2	0
PITCH	Pitch angle [-π/2 to +π/2]	10^{-4}	rad	int16	2	2
YAW	Yaw angle [-π to +π]	10^{-4}	rad	int16	2	4
				Total Size		6

Message name (ID)	SBG_ECAN_MSG_EKF_ORIENTATION_ACC (0x133)					
Field	Description	Scaling	Unit	Format	Size	Offset
ROLL_ACC	1σ Roll angle accuracy [0 to +π]	10^{-4}	rad	uint16	2	0
PITCH_ACC	1σ Pitch angle accuracy [0 to +π/2]	10^{-4}	rad	uint16	2	2
YAW_ACC	1σ Yaw angle accuracy [0 to +π]	10^{-4}	rad	uint16	2	4
				Total size		6

6.3.6.3. Position data

Inertially aided position data consists of a latitude, longitude and altitude expressed in the WGS84 datum. Estimated standard deviations are also returned by the Kalman Filter.

Message name (ID)	SBG_ECAN_MSG_EKF_POS (0x134)					
Field	Description	Scaling	Unit	Format	Size	Offset
LATITUDE	Latitude angle, positive north.	10^{-7}	°	int32	4	0
LONGITUDE	Longitude angle, positive east.	10^{-7}	°	int32	4	4
				Total size		8

Message name (ID)	SBG_ECAN_MSG_EKF_ALTITUDE (0x135)					
Field	Description	Scaling	Unit	Format	Size	Offset
ALTITUDE	Altitude above Mean Sea Level.	10^{-3}	m	int32	4	0
UNDULATION	Altitude difference between the geoid and the Ellipsoid.	0.005	m	int16	2	4
				Total size		6

Message name (ID)	SBG_ECAN_MSG_EKF_POS_ACC (0x136)					
Field	Description	Scaling	Unit	Format	Size	Offset
LATITUDE_ACC	1σ Latitude accuracy.	10^{-2}	m	uint16	2	0
LONGITUDE_ACC	1σ Longitude accuracy.	10^{-2}	m	uint16	2	2
ALTITUDE_ACC	1σ Vertical Position accuracy.	10^{-2}	m	uint16	2	4
				Total size		6

6.3.6.4. Velocity Data

Inertially aided velocity measurements that are either expressed in North, East, Down frame or in the INS body axis X,Y,Z. Estimated standard deviations are also returned by the Kalman Filter.

Message name (ID)	SBG_ECAN_MSG_EKF_VEL_NED (0x137)					
Field	Description	Scaling	Unit	Format	Size	Offset
VELOCITY_N	Velocity in North direction.	10^{-2}	$m.s^{-1}$	int16	2	0
VELOCITY_E	Velocity in East direction.	10^{-2}	$m.s^{-1}$	int16	2	2
VELOCITY_D	Velocity in Down direction.	10^{-2}	$m.s^{-1}$	int16	2	4
				Total size		6

Message name (ID)	SBG_ECAN_MSG_EKF_VEL_NED_ACC (0x138)					
Field	Description	Scaling	Unit	Format	Size	Offset
VELOCITY_ACC_N	1σ Velocity in North direction accuracy.	10^{-2}	$m.s^{-1}$	uint16	2	0
VELOCITY_ACC_E	1σ Velocity in East direction accuracy.	10^{-2}	$m.s^{-1}$	uint16	2	2
VELOCITY_ACC_D	1σ Velocity in Down direction accuracy.	10^{-2}	$m.s^{-1}$	uint16	2	4
				Total size		6

Message name (ID)	SBG_ECAN_MSG_EKF_VEL_BODY (0x139)					
Field	Description	Scaling	Unit	Format	Size	Offset
VELOCITY_X	Velocity in body/INS X direction.	10^{-2}	$m.s^{-1}$	int16	2	0
VELOCITY_Y	Velocity in body/INS Y direction.	10^{-2}	$m.s^{-1}$	int16	2	2
VELOCITY_Z	Velocity in body/INS Z direction.	10^{-2}	$m.s^{-1}$	int16	2	4
				Total size		6

6.3.7. Automotive Outputs

All the messages in this section have been especially designed for automotive applications.

Message name (ID)	SBG_ECAN_MSG_AUTO_TRACK_SLIP_CURV (0x220)					
Field	**Description**	**Scaling**	**Unit**	**Format**	**Size**	**Offset**
ANGLE_TRACK	Track course angle/direction of travel [-π to +π]	10^{-4}	rad	int16	2	0
ANGLE_SLIP	Vehicle slip angle [-π to +π]	10^{-4}	rad	int16	2	2
CURVATURE_RADIUS	Curvature radius in meters based on down rotation rate	10^{-2}	m	uint16	2	4
AUTO_STATUS	Status bitmasks as AUTO_STATUS definition	-	-	uint8	1	6
				Total size		7

AUTO_STATUS definition

This field must be checked in order to know which data is valid or not.

Bit	Name	Description
0	SBG_ECOM_AUTO_TRACK_VALID	Set to 1 if the track angle is valid
1	SBG_ECOM_AUTO_SLIP_VALID	Set to 1 if the slip angle is valid
2	SBG_ECOM_AUTO_CURVATURE_VALID	Set to 1 if the curvature radius is valid

6.3.8. Heave, Surge, Sway

Ship Motion data represents 3D displacement, velocities and accelerations for a vessel. In addition to heave, surge and sway measurements, main heave period as well as 3D velocities and accelerations can be output by the device.

Some devices can output both real time and delayed heave values, these two outputs share the exact same message format.

Message name (ID)	SBG_ECAN_MSG_SHIP_MOTION_INFO (0x140), SBG_ECAN_MSG_SHIP_MOTION_HP_INFO (0x14A)					
Field	Description	Scaling	Unit	Format	Size	Offset
TIME_STAMP	Time since sensor is powered up.	1	µs	uint32	4	0
PERIOD	Main heave period.	10^{-2}	s	uint16	2	4
HEAVE_STATUS	Ship motion status	-	-	uint16	2	6
				Total size		8

HEAVE_STATUS definition

Please refer to the sbgECom serial log HEAVE_STATUS definition.

Message name (ID)	SBG_ECAN_MSG_SHIP_MOTION_0 (0x141), SBG_ECAN_MSG_SHIP_MOTION_HP_0 (0x14B)					
Field	Description	Scaling	Unit	Format	Size	Offset
SURGE	Surge motion (positive forward).	10^{-3}	m	int16	2	0
SWAY	Sway motion (positive right).	10^{-3}	m	int16	2	2
HEAVE	Heave motion (positive down).	10^{-3}	m	int16	2	4
				Total size		6

Message name (ID)	SBG_ECAN_MSG_SHIP_MOTION_1 (0x145), SBG_ECAN_MSG_SHIP_MOTION_HP_1 (0x14C)					
Field	Description	Scaling	Unit	Format	Size	Offset
ACCEL_X	Longitudinal acceleration (positive forward).	10^{-2}	$m.s^{-2}$	int16	2	0
ACCEL_Y	Lateral acceleration (positive right).	10^{-2}	$m.s^{-2}$	int16	2	2
ACCEL_Z	Vertical acceleration (positive down).	10^{-2}	$m.s^{-2}$	int16	2	4
				Total size		6

Message name (ID)	SBG_ECAN_MSG_SHIP_MOTION_2 (0x149), SBG_ECAN_MSG_SHIP_MOTION_HP_2 (0x14D)					
Field	Description	Scaling	Unit	Format	Size	Offset
VEL_X	Longitudinal velocity (positive forward).	10^{-2}	$m.s^{-1}$	int16	2	0
VEL_Y	Lateral velocity (positive right).	10^{-2}	$m.s^{-1}$	int16	2	2
VEL_Z	Vertical velocity (positive down).	10^{-2}	$m.s^{-1}$	int16	2	4
				Total size		6

6.3.9. Magnetometer output

Fully calibrated and normalized magnetometers values in arbitrary units as well as associated accelerometer measurement. If well calibrated, the norm of the magnetic vector should equal 1.

Message name (ID)	SBG_ECAN_MSG_MAG_0 (0x150)					
Field	Description	Scaling	Unit	Format	Size	Offset
TIME_STAMP	Time since sensor is powered up.	1	µs	uint32	4	0
MAG_STATUS	Magnetometer status bit-mask.	-	-	uint16	2	4
				Total size		6

MAG_STATUS definition

Please refer to the sbgECom serial log MAG_STATUS definition.

Message name (ID)	SBG_ECAN_MSG_MAG_1 (0x151)					
Field	Description	Scaling	Unit	Format	Size	Offset
MAG_X	Magnetometer output, X axis.	10^{-3}	a.u.	int16	2	0
MAG_Y	Magnetometer output, Y axis.	10^{-3}	a.u.	int16	2	2
MAG_Z	Magnetometer output, Z axis.	10^{-3}	a.u.	int16	2	4
				Total size		6

Message name (ID)	SBG_ECAN_MSG_MAG_2 (0x152)					
Field	Description	Scaling	Unit	Format	Size	Offset
ACC_X	Accelerometer output, X axis.	10^{-2}	$m.s^{-2}$	int16	2	0
ACC_Y	Accelerometer output, Y axis.	10^{-2}	$m.s^{-2}$	int16	2	2
ACC_Z	Accelerometer output, Z axis.	10^{-2}	$m.s^{-2}$	int16	2	4
				Total size		6

6.3.10. Odometer output

Odometer status information, time since reset and raw velocity using detected odometer pulses. The time stamp field is not necessary aligned with main loop as it dates the actual odometer data.

Message name (ID)	SBG_ECAN_MSG_ODO_INFO (0x160)					
Field	Description	Scaling	Unit	Format	Size	Offset
TIME_STAMP	Time since sensor is powered up.	1	µs	uint32	4	0
ODO_STATUS	Odometer velocity status bit-mask.	-	-	uint16	2	4
				Total size		6

ODO_STATUS definition

Please refer to the sbgECom serial log ODO_VEL_STATUS definition.

Message name (ID)	SBG_ECAN_MSG_ODO_VEL (0x161)					
Field	Description	Scaling	Unit	Format	Size	Offset
VELOCITY	Velocity in odometer direction.	10^{-2}	$m.s^{-1}$	int16	2	0
				Total size		2

SBG SYSTEMS

6.3.11. AirData output

Outputs related to the AirData module such as barometric pressure, altitude, true airspeed, ...

Message name (ID)	SBG_ECAN_MSG_AIR_DATA_INFO (0x162)					
Field	Description	Scaling	Unit	Format	Size	Offset
TIME_STAMP	Time since sensor is powered up	-	µs	uint32	4	0
AIR_DATA_STATUS	Airdata module and sensor status	-	-	uint8	1	4
AIR_TEMPERATURE	Outside air temperature	10^{-2}	°C	int16	2	5
				Total size		7

AIR_DATA_STATUS definition

Please refer to the sbgECom serial log AIRDATA_STATUS definition.

Message name (ID)	SBG_ECAN_MSG_AIR_DATA_ALTITUDE (0x163)					
Field	Description	Scaling	Unit	Format	Size	Offset
PRESSURE_ABS	Absolute barometric pressure	10^{-3}	Pa	uint32	4	0
ALTITUDE	Barometric altitude computed positive upward.	10^{-3}	m	int32	4	4
				Total size		8

Message name (ID)	SBG_ECAN_MSG_AIR_DATA_AIRSPEED (0x164)					
Field	Description	Scaling	Unit	Format	Size	Offset
PRESSURE_DIFF	Differential pressure measured by a pitot tube	10^{-3}	Pa	int32	4	0
AIRSPEED	True airspeed positive forward	10^{-2}	$m.s^{-1}$	int16	2	4
				Total size		6

6.3.12. Depth Sensor output

Outputs related to the subsea Depth sensor module.

Message name (ID)	SBG_ECAN_MSG_DEPTH_INFO (0x166)					
Field	Description	Scaling	Unit	Format	Size	Offset
TIME_STAMP	Time since sensor is powered up	1	µs	uint32	4	0
DEPTH_STATUS	Depth module and sensor status	-	-	uint8	1	4
				Total size		5

DEPTH_STATUS definition

Please refer to the sbgECom serial log DEPTH_STATUS definition.

Message name (ID)	SBG_ECAN_MSG_DEPTH_ALTITUDE (0x167)					
Field	Description	Scaling	Unit	Format	Size	Offset
PRESSURE_ABS	Absolute water pressure	10^{-1}	Pa	uint32	4	0
DEPTH	Altitude from depth sensor positive upward.	10^{-3}	m	int32	4	4
				Total size		8

6.3.13. GNSS 1 and 2 outputs

6.3.13.1. GNSS Velocity

These logs return the GNSS velocity, velocity accuracy, course and status as computed by the primary or secondary GNSS receiver. The time stamp is not aligned on main loop but instead of that, it dates the actual GNSS velocity data.

Message name (ID)	SBG_ECAN_MSG_GPS1_VEL_INFO (0x170), SBG_ECAN_MSG_GPS2_VEL_INFO (0x180)					
Field	Description	Scaling	Unit	Format	Size	Offset
TIME_STAMP	Time since sensor is powered up.	1	µs	uint32	4	0
GPS_VEL_STATUS	GPS velocity fix and status bit-mask.	-	-	uint32	4	4
				Total size		8

GPS_VEL_STATUS definition

Please refer to the sbgECom serial log GPS_VEL_STATUS definition.

Message name (ID)	SBG_ECAN_MSG_GPS1_VEL (0x171), SBG_ECAN_MSG_GPS2_VEL (0x181)					
Field	Description	Scaling	Unit	Format	Size	Offset
VEL_N	Velocity in North direction.	10^{-2}	$m.s^{-1}$	int16	2	0
VEL_E	Velocity in East direction.	10^{-2}	$m.s^{-1}$	int16	2	2
VEL_D	Velocity in Down direction.	10^{-2}	$m.s^{-1}$	int16	2	4
				Total size		6

Message name (ID)	SBG_ECAN_MSG_GPS1_VEL_ACC (0x172) , SBG_ECAN_MSG_GPS2_VEL_ACC (0x182)					
Field	Description	Scaling	Unit	Format	Size	Offset
VEL_ACC_N	1σ Accuracy in North direction.	10^{-2}	$m.s^{-1}$	uint16	2	0
VEL_ACC_E	1σ Accuracy in East direction.	10^{-2}	$m.s^{-1}$	uint16	2	2
VEL_ACC_D	1σ Accuracy in Down direction.	10^{-2}	$m.s^{-1}$	uint16	2	4
				Total size		6

Message name (ID)	SBG_ECAN_MSG_GPS1_COURSE (0x173), SBG_ECAN_MSG_GPS2_COURSE (0x183)					
Field	Description	Scaling	Unit	Format	Size	Offset
COURSE	True direction of motion over ground (0 to 360°).	10^{-2}	°	uint16	2	0
COURSE_ACC	1σ course accuracy.	10^{-2}	°	uint16	2	2
				Total size		4

6.3.13.2. GNSS Position

These logs return the GNSS position, position accuracy and status as computed by the primary or secondary GNSS receiver. The time stamp is not aligned on main loop but instead of that, it dates the actual GNSS positioning data.

Message name (ID)	SBG_ECAN_MSG_GPS1_POS_INFO (0x174), SBG_ECAN_MSG_GPS2_POS_INFO (0x184)					
Field	Description	Scaling	Unit	Format	Size	Offset
TIME_STAMP	Time since sensor is powered up.	1	µs	uint32	4	0
GPS_POS_STATUS	GPS position fix and status bit-mask.	-	-	uint32	4	4
				Total size		8

GPS_POS_STATUS definition

Please refer to the sbgECom serial log GPS_POS_STATUS definition.

Message name (ID)	SBG_ECAN_MSG_GPS1_POS (0x175), SBG_ECAN_MSG_GPS2_POS (0x185)					
Field	Description	Scaling	Unit	Format	Size	Offset
LATITUDE	Latitude, positive North.	10^{-7}	°	int32	4	0
LONGITUDE	Longitude, positive East.	10^{-7}	°	int32	4	4
				Total size		8

Message name (ID)	SBG_ECAN_MSG_GPS1_POS_ALT (0x176), SBG_ECAN_MSG_GPS2_POS_ALT (0x186)					
Field	Description	Scaling	Unit	Format	Size	Offset
ALTITUDE	Altitude Above Mean Sea Level.	10^{-3}	m	int32	4	0
UNDULATION	Altitude difference between the geoid and the Ellipsoid.	0.005	m	int16	2	4
NUM_SV	Number of space vehicles used in solution	-	-	uint8	1	6
DIFF_CORR_AGE	Age of differential corrections (Set to 0xFF if not available)	1	s	uint8	1	7
				Total size		8

Message name (ID)	SBG_ECAN_MSG_GPS1_POS_ACC (0x177), SBG_ECAN_MSG_GPS2_POS_ACC (0x187)					
Field	Description	Scaling	Unit	Format	Size	Offset
LAT_ACC	1σ Latitude Accuracy.	10^{-2}	m	uint16	2	0
LONG_ACC	1σ Longitude Accuracy.	10^{-2}	m	uint16	2	2
ALT_ACC	1σ Altitude Accuracy.	10^{-2}	m	uint16	2	4
BASE_STATION_ID	ID of the base station used (Set to 0xFFFF if not available)	-	-	uint16	2	6
				Total size		8

6.3.13.3. GNSS True Heading

These logs return the GNSS true heading and pitch solutions as well as status as computed by the primary or secondary dual antenna GNSS receiver. The time stamp is not aligned on main loop but instead of that, it dates the actual GNSS true heading data.

Message name (ID)	SBG_ECAN_MSG_GPS1_HDT_INFO (0x178), SBG_ECAN_MSG_GPS2_HDT_INFO (0x188)					
Field	Description	Scaling	Unit	Format	Size	Offset
TIME_STAMP	Time since sensor is powered up.	1	µs	uint32	4	0
GPS_HDT_STATUS	GPS True Heading status.	-	-	uint16	2	4
				Total size		6

GPS_HDT_STATUS definition

Please refer to the sbgECom serial log GPS_HDT_STATUS definition.

Message name (ID)	SBG_ECAN_MSG_GPS1_HDT (0x179), SBG_ECAN_MSG_GPS2_HDT (0x189)					
Field	Description	Scaling	Unit	Format	Size	Offset
TRUE_HEADING	True heading angle (0 to 360°).	10^{-2}	°	uint16	2	0
TRUE_HEADING_ACC	1σ True heading estimated accuracy (0 to 360°).	10^{-2}	°	uint16	2	2
PITCH	Pitch angle from the master to the rover.	10^{-2}	°	int16	2	4
PITCH_ACC	1σ pitch estimated accuracy.	10^{-2}	°	uint16	2	6
				Total size		8

6.3.14. Event Markers

All SBG Systems Inertial Systems can detect events markers at up to 1 kHz on synchronization input signals such as Sync A, Sync B, Sync C, Sync D and Sync E. For each input synchronization signal, the device can output a binary log that returns the time of each received event during the last past 5 milliseconds (the maximum output rate is 200 Hz).

The TIME_STAMP field dates the first event that has been received during the last 5 ms. Other events received during the same time slot (5ms) are dated using a time offset to reduce the log size.

Example

If three events are received during the last 5 ms, each event will be dated using the following rules:

- First received event time is directly stored in TIME_STAMP
- Second received event time is TIME_STAMP + TIME_OFFSET_0
- Third received event time is TIME_STAMP + TIME_OFFSET_1

The other time offset fields will be set to 0 and the EVENT_STATUS flag will reflect which time offset fields are valid.

Message name (ID)	SBG_ECAN_MSG_EVENT_INFO_A (0x200), SBG_ECAN_MSG_EVENT_INFO_B (0x202), SBG_ECAN_MSG_EVENT_INFO_C (0x204), SBG_ECAN_MSG_EVENT_INFO_D (0x206), SBG_ECAN_MSG_EVENT_INFO_E (0x207),					
Field	Description	Scaling	Unit	Format	Size	Offset
TIME_STAMP	Time since sensor is powered up.	1	µs	uint32	4	0
EVENT_STATUS	Status bit-mask.	-	-	uint16	2	4
				Total size		6

EVENT_STATUS definition

Please refer to the sbgECom serial log EVENT_STATUS definition.

Message name (ID)	SBG_ECAN_MSG_EVENT_TIME_A (0x201), SBG_ECAN_MSG_EVENT_TIME_B (0x203), SBG_ECAN_MSG_EVENT_TIME_C (0x205), SBG_ECAN_MSG_EVENT_TIME_D (0x207), SBG_ECAN_MSG_EVENT_TIME_E (0x209)					
Field	Description	Scaling	Unit	Format	Size	Offset
TIME_OFFSET_0	Time offset for the second received event.	1	µs	uint16	2	0
TIME_OFFSET_1	Time offset for the third received event.	1	µs	uint16	2	2
TIME_OFFSET_2	Time offset for the fourth received event.	1	µs	uint16	2	4
TIME_OFFSET_3	Time offset for the fifth received event.	1	µs	uint16	2	6
				Total size		8

Note: The device can support events markers at up to 1 kHz. If too much events are sent, it may overload the internal CPU leading to decreased performance and reliability.

Warning: Never leave an activated Sync In signal unconnected as noise on the line may trigger spurious events at very high rates.

7. Support

Our goal is to provide the best experience to our customers. If you have any question, comment or problem with the use of your product, we would be glad to help you, so feel free to contact us:

EMEA:

SBG Systems S.A.S.
1 avenue Eiffel
78420 Carrières-sur-Seine
FRANCE

Phone: +33 1 80 88 43 70
support@sbg-systems.com

Americas:

SBG Systems North America, Inc
5932 Bolsa Avenue, Suite #103
Huntington Beach, CA 92649
USA

Phone: +1 (657) 549-5807
support@sbg-systems.com