Ellipse AHRS/INS

High-Performance, Miniature Inertial Sensors

Firmware Reference Manual

Document	ELLIPSEFWM.1
Revision	1 – Oct 21, 2014

Support support@sbg-systems.com
+33 1 80 88 45 00

SBG Systems
3bis, chemin de la Jonchère
92500 Rueil-Malmaison
FRANCE

Revision history

Rev.	Date	Author	Information
1	Oct 21, 14	Alexis Guinamard	First public release
Pre2	Jun 26, 14	Raphaël Siryani	Added magnetic calibration related commands
Pre1	Jun 3, 2014	Alexis Guinamard	First version of this document

Index

1. Introduction

This firmware reference manual presents the input and output protocols supported by the Ellipse AHRS and Ellipse INS series.

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

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

1.3. Endianness

The Ellipse and all provided libraries use little endian data format. However, the sbgECom communication library should be compatible with big endian platforms.

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

1.4. Conventions and units

The Ellipse uses 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 interfaces

The Ellipse embeds several RS-232 or RS-422 ports. Some are available in full duplex, other are just Rx ports.

All serial ports use the following transmission format for communications:

- Available baudrates: 4800, 9600, 19200, 38400, 57600, 115200, 230400, 460800, 921600
- 8 bits data
- 1 stop bit
- No parity
- No control flow

1.6. CAN interface

The protocol described in this documentation is used to communicate with an Ellipse 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 Ellipse CAN implementation supports both CAN 2.0A and CAN 2.0B standards in a very versatile manner.

This documentation contains all information needed to configure and integrate an Ellipse to a CAN bus.

1.6.1. Specifications

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 calcCRC(const void *pBuffer, uint16 bufferSize)
{
      const uint8 *pBytesArray =  (const uint8*)pBuffer;
      uint16 poly = 0x8408;
      uint16 crc = 0;
      uint8 carry;
      uint8 i_bits;
      uint16 j;

      for (j =0; j < bufferSize; j++)
      {
            crc = crc ^ pBytesArray[j];
            for (i_bits = 0; i_bits < 8; i_bits++)
            {
                  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.

SbgEComClass enum definitions

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 protocol input/output log messages.
SBG_ECOM_CLASS_LOG_NMEA_0	0x02	Class that contains NMEA (and NMEA like) output logs. **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 third party output logs. **Note:** This class is only used for identification purpose and does not contain any sbgECom message.

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. There are input logs and output logs.

2.1.3. Continuous / Triggered mode

This mode is used for data input and output. Data computed by the Ellipse are sent in a uni-directional mode and require no answer from receiver side. In the same time, aiding data provided by third party equipments can be sent to the Ellipse. No answer will be sent by the Ellipse at such message reception.

All output logs can be configured for continuous/triggered output independently.

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 Ellipse 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 (0x00)`

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	Error Code	-	enum	2	2
				Total size	4

SbgErrorCode enum definition

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

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	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	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	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	2	0
				Total size	2

Device response format:

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

2.2.3. Commands List

The Ellipse series is 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 (0x00)	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_SET_MOTION_PROFILE (06)	Set a new motion profile
SBG_ECOM_CMD_MOTION_PROFILE_INFO (07)	Get motion profile information
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_SET_MAGNETOMETER_MODEL (10)	Set a new magnetometer error model
SBG_ECOM_CMD_MAGNETOMETER_MODEL_INFO (11)	Get magnetometer error model information
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_SET_GNSS_1_MODEL (16)	Set a new GNSS model
SBG_ECOM_CMD_GNSS_1_MODEL_INFO (17)	Get GNSS model information
SBG_ECOM_CMD_GNSS_1_LEVER_ARM_ALIGNMENT (18)	GNSS installation configuration (lever arm, antenna alignments)
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_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_LEGACY_CONT_OUTPUT_CONF (31)	Legacy serial output mode configuration
SBG_ECOM_CMD_ADVANCED_CONF (32)	Advanced settings configuration

2.2.4. Special settings commands

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

2.2.4.1. 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
setingAction	Special setting action to perform		enum	1	0
				Total size	1

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

SbgEComSettingAction enum definition

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.

2.2.4.2. 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
transferCmd	Transfer protocol CMD	-	enum	2	0
transferPayload	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.

2.2.4.3. 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
transferCmd	Transfer protocol CMD	-	enum	2	0
transferPayload	Transfer protocol payload	-	-	0 → n	1
				Total size	2+n

2.2.5. Information commands

2.2.5.1. 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

2.2.6. Sensor parameters

2.2.6.1. 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.

2.2.6.2. SBG_ECOM_CMD_SET_MOTION_PROFILE (06)

This command is used to send a new motion profile to be applied.

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
transferCmd	Transfer protocol CMD	-	enum	2	0
transferPayload	Transfer protocol payload	-	-	0 → n	2
				Total size	2+n

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

2.2.6.3. SBG_ECOM_CMD_MOTION_PROFILE_INFO (07)

You can check which motion profile is currently in use by asking for the motion profile details.

Sending this frame without payload will generate the device answer:

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

2.2.6.4. 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	-	enum	1	0
axisDirectionY	IMU Y axis direction in vehicle frame	-	enum	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.

AxisDirection enum definition

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

2.2.6.5. 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
gnss1ModulePortAssignment	GNSS module port assignment	-	enum	1	0
gnss1ModuleSyncAssignment	GNSS module sync assignment	-	enum	1	1
reserved	Reserved field for future use	-	-	6	2
rtcmPortAssignment	RTCM input port assignment for Ellipse-N DGPS		enum	1	8
reserved	Reserved field for future use	-	-	1	9
odometerPinAssignment	Odometer module pin assignment	-	enum	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.

Note 2: GNSS1 module configuration can only be set to an external port on Ellipse-E version. Ellipse-N users must set this module to MODULE_INTERNAL. On the other hand, rtcmModule is only available for Ellipse-N users. This module must be set to MODULE_DISABLED for other users.

SbgEComModulePortAssignment enum definition

Name	Value	Description
SBG_ECOM_MODULE_DISABLED	255	Module is disabled
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_INTERNAL	5	Module is connected internally

SbgEComModuleSyncAssignment enum definition

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

SbgEComOdometerPinAssignment enum definition

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.

2.2.7. Magnetometer configuration

2.2.7.1. SBG_ECOM_CMD_SET_MAGNETOMETER_MODEL (10)

This command is used to send a specific magnetometer configuration.

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
transferCmd	Transfer protocol CMD	-	enum	2	0
transferPayload	Transfer protocol payload	-	-	0 → n	1
				Total size	2+n

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

2.2.7.2. SBG_ECOM_CMD_MAGNETOMETER_MODEL_INFO (11)

You can check which magnetometer model is currently in use by asking for the model details.

Sending this frame without payload will generate the device answer:

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

2.2.7.3. 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	-	enum	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.

SbgEComRejectionMode enum definition

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

2.2.7.4. 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.

2.2.8. 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.

2.2.8.1. 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. It could be 3D or 2D.	-	enum	1	0
bandwidth	Tell the device that we should have low, medium or high dynamics during the magnetic calibration process.	-	enum	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.

SbgEComMagCalibMode enum definition

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.

SbgEComMagCalibBandwidth enum definition

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.

2.2.8.2. 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. This indicator is computed based on the magnetic field norm deviation after calibration.	-	enum	1	0
confidence	Confidence indicator that should be read to interpret the quality indicator. 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	1	1
advancedStatus	Set of bit masks used to report advanced information on the magnetic calibration status. Read this status to get more details on a magnetic calibration that has failed.	-	Mask	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
maxNumPoints	Maximum number of magnetic field points that can be stored internally.	-	uint16	2	40
numPoints	Number of magnetic field points stored internally and used to compute the magnetic calibration.	-	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

SbgEComMagCalibQuality 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.

SbgEComMagCalibConfidence enum definition

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.

SbgEComMagCalibAdvStatus mask definition

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 interferences 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.

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.

2.2.9. GNSS configuration

2.2.9.1. SBG_ECOM_CMD_SET_GNSS_1_MODEL (16)

This command is used to send a specific GNSS receiver configuration.

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
transferCmd	Transfer protocol CMD	-	enum	2	0
transferPayload	Transfer protocol payload	-	-	0 → n	1
				Total size	2+n

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

2.2.9.2. SBG_ECOM_CMD_GNSS_1_MODEL_INFO (17)

You can check which GNSS model is currently in use by asking for the model details. Sending this frame without payload will generate the device answer:

Field	Description	Unit	Format	Size	Offset
gnssModelId	GNSS model identifier	-	uint32	4	0
gnssModelRevision	GNSS model version number	-	uint32	4	4
				Total size	8

2.2.9.3. 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	Pitch 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.

2.2.9.4. 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	-	enum	1	0
velRejectMode	Rejection mode for velocity	-	enum	1	1
CourseRejectMode	Rejection mode for course over ground	-	enum	1	2
hdtRejectMode	Rejection mode for true heading	-	enum	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.

Note 2: Check SbgEComRejectionMode enum definition section for further details about rejection modes.

2.2.10. Odometer configuration

2.2.10.1. SBG_ECOM_CMD_ODO_CONF (20)

Sent without parameter, this frame retrieves the odometer configuration. 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.

2.2.10.2. 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.

2.2.10.3. 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	-	enum	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.

> **Note 2:** Check SbgEComRejectionMode enum definition section for further details about rejection modes.

2.2.11. Interfaces configuration

2.2.11.1. 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 be checked	-	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 be checked	-	enum	1	0
baudRate	Baudrate to use for communications. Possible values are: • 4800 • 9600 • 19200 • 38400 • 115200 • 230400 • 460800 • 921600	bps	uint32	4	1
mode	UART mode	-	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.

SbgEComPortID enum definition

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

SbgEComPortMode enum definition

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

> **Note:** There may be limitations regarding the ports configuration: PORT A is the main interface and cannot be turned OFF. PORT B and PORT D are only available in RS-232 mode. Finally, PORT C and PORT D are only available on Ellipse-E version.

2.2.11.2. 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. Possible values are: • 1000 • 500 • 250 • 125 • 100 • 50 • 20 • 10 • 0 (disabled)	kbps	uint16	2	0
				Total size	2

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.

2.2.11.3. 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	-	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	-	enum	2	0
outputMode	Output mode	-	enum	2	2
userID	User defined ID	-	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.

Check section SbgEComOutputMode enum definition for more details about the output mode.

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. Events configuration

2.2.12.1. 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	-	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	-	enum	1	0
sensitivity	Logic input corresponding sensitivity	-	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.

SbgEComSyncInID enum definition

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

Note: Sync IN C and D are only available on Ellipse-E version.

SbgEComSyncInSensitivity enum definition

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)

2.2.12.2. 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	-	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	-	enum	1	0
reserved	-	-	uint8	1	1
outputFunction	Main function of the sync Out pin	-	enum	2	2
polarity	Polarity of corresponding logic output	-	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.

SbgEComSyncOutID enum definition

Name	Value	Description
SBG_ECOM_SYNC_OUT_A	0	Synchronization output A on Main connector
SBG_ECOM_SYNC_OUT_B	1	Synchronization output on Auxiliary connector

Note: Sync OUT B is only available on Ellipse-E version.

SbgEComSyncOutFunction enum definition

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_VIRTUAL_ODO	10002	Output is generated when a new virtual odometer event occurs
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

SbgEComSyncOutPolarity enum definition

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

2.2.12.3. SBG_ECOM_CMD_VIRTUAL_ODOMETER_CONF (28)

Sent without any parameter, this frame retrieves the virtual odometer configuration. Sent with full payload, this frame configures the virtual odometer distance between pulses.

Field	Description	Unit	Format	Size	Offset
distance	Distance between two pulses	m	float	4	0
				Total size	4

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.

2.2.13. Output configuration

2.2.13.1. 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
portId	Port Name	-	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
portId	Port Name	-	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.

2.2.13.2. 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
portId	Port Name	-	enum	1	0
msgId	Output message identifier		enum	1	1
classId	Output class identifier		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
portId	Port Name	-	enum	1	0
msgId	Output message identifier		enum	1	1
classId	Output class identifier		enum	1	1
outputMode	Output mode	-	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.

Check sections SbgEComPortID enum definition and SbgEComClass enum definitions for more details about the Ports and Class IDs used in this message.

Check section SBG_ECOM_CLASS_LOG_ECOM_0 for more details about available sbgECom log messages.

Check sections NMEA Logs list and Third party Logs list for more details about available messages IDs.

SbgEComOutputMode enum definition

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_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_VIRTUAL_ODO	10002	Output is generated when a new virtual odometer event occurs
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

2.2.13.3. SBG_ECOM_CMD_LEGACY_CONT_OUTPUT_CONF (31)

This command enables legacy IG-Devices protocol continuous output for backward compatibility with previous IG-Devices. It is not recommended to use this communication mode for new designs.

Sent with limited payload, this message asks for current configuration with specified PORT ID.

Limited payload definition

Field	Description	Unit	Format	Size	Offset
portId	Port Name	-	enum	1	0
				Total size	1

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

Full Payload definitions

Field	Description	Unit	Format	Size	Offset
portId	Port Name	-	enum	1	0
outputMask	Output mask as defined in IG-devices Serial Protocol Specifications	-	uint32	4	1
outputFormat	Format of the output (float or fixed point)	-	enum	1	5
outputEndian	Endianness of the output.	-	enum	1	6
outputMode	Output mode	-	enum	2	7
				Total size	9

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

Check sections SbgEComLegacyFormat enum definition and SbgEComLegacyEndian enum definition for more details about the format and the endian fields used in this message.

Check section SbgEComSyncOutFunction enum definition for more details about the output mode used in this message.

SbgEComLegacyFormat enum definition

Name	Value	Description
SBG_ECOM_LEGACY_FORMAT_FLOAT	0	Data is generated in floating point format
SBG_ECOM_LEGACY_FORMAT_FIXED	1	Data is generated in fixed point format

SbgEComLegacyEndian enum definition

Name	Value	Description
SBG_ECOM_LEGACY_LITTLE_ENDIAN	0	Data is generated in little endian
SBG_ECOM_LEGACY_BIG_ENDIAN	1	Data is generated in big endian

Check IG-Devices Serial Protocol Specifications for more details about the output mask and the generated continuous output definition.

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

2.2.14. Advanced configuration

2.2.14.1. 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	-	enum	1	0
				Total size	1

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

TimeReference enum definition:

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 GPS PPS signal, (see GPS module assignment).

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.

2.3. 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.

The following output logs are available:

Name (MSG ID)	Description
SBG_ECOM_LOG_STATUS (01)	Status general, clock, com aiding, solution, heave
SBG_ECOM_LOG_IMU_DATA (03)	Includes IMU status, acc., gyro, temp delta speeds and delta angles values
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_0 (09)	Heave, surge and sway and accelerations on each axis for up to 4 points
SBG_ECOM_LOG_UTC_TIME (02)	Provides UTC time reference
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)	GPS velocities from primary or secondary GPS receiver
SBG_ECOM_LOG_GPS1_POS (14)	GPS positions from primary or secondary GPS receiver
SBG_ECOM_LOG_GPS1_HDT (15)	GPS true heading from dual antenna system
SBG_ECOM_LOG_ODO_VEL (19)	Provides odometer velocity
SBG_ECOM_LOG_EVENT_A/B/C/D (24, 25, 26, 27)	Event markers sent when events are detected on a sync in pin
SBG_ECOM_LOG_PRESSURE (31)	Barometric altimeter output

2.3.1. General information, time

2.3.1.1. SBG_ECOM_LOG_STATUS (01)

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

This log is useful for advanced status information.

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_0	Reserved field for backward compatibility	-	uint16	2	6
COM STATUS	Communication status bitmask and enums.	-	uint32	4	8
AIDING STATUS	Aiding equipments status bitmask and enums.	-	uint32	4	12
RESERVED_1	Reserved field for backward compatibility	-	uint32	4	16
RESERVED	Reserved field for future use	-	uint16	2	20
				Total size	22

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.

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
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

Bit	Name	Type	Description
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
25	SBG_ECOM_CAN_RX_OK	Mask	Set to 0 in case of saturation on CAN Bus output buffer
26	SBG_ECOM_CAN_TX_OK	Mask	Set to 0 in case of saturation on CAN Bus input buffer
27-29	SBG_ECOM_CAN_BUS	Enum	Define the CAN Bus status (see the 1 below).

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 1: 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
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

2.3.1.2. SBG_ECOM_LOG_UTC_TIME (02)

Provides UTC time reference. This frame also provides a time correspondence between Ellipse TIME_STAMP value and actual UTC Time.

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 2: Clock Status enumeration below).
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 3: UTC time status enumeration below).

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 2: 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 3: UTC time status enumeration

2.3.2. Internal IMU data

2.3.2.1. SBG_ECOM_LOG_IMU_DATA (03)

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

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/s²	float	4	6
ACCEL_Y	Filtered Accelerometer – Y axis	m/s²	float	4	10
ACCEL_Z	Filtered Accelerometer – Z axis	m/s²	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/s²	float	4	34
DELTA_VEL_Y	Sculling output – Y axis	m/s²	float	4	38
DELTA_VEL_Z	Sculling output – Z axis	m/s²	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

2.3.3. 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 4 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 (data used and valid since 3s)
9	SBG_ECOM_SOL_MAG_REF_USED	Set to 1 if magnetometer is used in solution (data used and valid since 3s)
10	SBG_ECOM_SOL_GPS1_VEL_USED	Set to 1 if GPS velocity is used in solution (data used and valid since 3s)
11	SBG_ECOM_SOL_GPS1_POS_USED	Set to 1 if GPS Position is used in solution (data used and valid since 3s)
12	SBG_ECOM_SOL_GPS1_COURSE_USED	Set to 1 if GPS Course is used in solution (data used and valid since 3s)
13	SBG_ECOM_SOL_GPS1_HDT_USED	Set to 1 if GPS True Heading is used in solution (data used and valid since 3s)
14	SBG_ECOM_SOL_GPS2_VEL_USED	Set to 1 if GPS2 velocity is used in solution (data used and valid since 3s)
15	SBG_ECOM_SOL_GPS2_POS_USED	Set to 1 if GPS2 Position is used in solution (data used and valid since 3s)
16	SBG_ECOM_SOL_GPS2_COURSE_USED	Set to 1 if GPS2 Course is used in solution (data used and valid since 3s)
17	SBG_ECOM_SOL_GPS2_HDT_USED	Set to 1 if GPS2 True Heading is used in solution (data used and valid since 3s)
18	SBG_ECOM_SOL_ODO_USED	Set to 1 if Odometer is used in solution (data used and valid since 3s)

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 4: Solution modes enumeration

2.3.3.1. SBG_ECOM_LOG_EKF_EULER (06)

Provides computed orientation in Euler angles format.

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

2.3.3.2. SBG_ECOM_LOG_EKF_QUAT (07)

Provides orientation in quaternion format.

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

2.3.3.3. SBG_ECOM_LOG_EKF_NAV (08)

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

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.	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

2.3.3.4. SBG_ECOM_LOG_SHIP_MOTION_0 (09)

Provides ship motion data (heave), velocity and accelerations.

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 – Fixed to 0	m	float	4	8
SWAY	Sway at main location – Fixed to 0	m	float	4	12
HEAVE	Heave at main location (positive down)	m	float	4	16
ACCEL_X	Longitudinal acceleration – Fixed to 0	$m.s^{-2}$	float	4	20
ACCEL_Y	Lateral acceleration – Fixed to 0	$m.s^{-2}$	float	4	24
ACCEL_Z	Vertical acceleration (positive down)	$m.s^{-2}$	float	4	28
VEL_X	Longitudinal velocity – Fixed to 0	$m.s^{-1}$	float	4	32
VEL_Y	Lateral velocity – Fixed to 0	$m.s^{-1}$	float	4	36
VEL_Z	Vertical velocity (positive down)	$m.s^{-1}$	float	4	40
STATUS	Ship motion output status	-	uint16	2	44
				Total size	46

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
3	SBG_ECOM_PERIOD_AVAILABLE	Set to 1 if the swell period is provided in this output
4	SBG_ECOM_PERIOD_VALID	Set to 1 if the period returned is assumed to be valid or not.

2.3.4. Aiding sensors outputs

2.3.4.1. SBG_ECOM_LOG_MAG (04)

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

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_Y	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

2.3.4.2. SBG_ECOM_LOG_MAG_CALIB (05)

This log provides a RAW buffer for magnetic calibration procedure.

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

2.3.4.3. SBG_ECOM_LOG_GPS1_VEL (13)

Provides raw GNSS velocity from primary GNSS receiver.

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

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 5 below).
[6-11]	Enum	SBG_ECOM_GPS_VEL_TYPE	The raw GPS velocity type (see the 6 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 5: 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 6: Raw GPS velocity type enumeration

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

2.3.4.4. SBG_ECOM_LOG_GPS1_POS (14)

Provides GNSS position from primary GNSS receiver.

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

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	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
				Total size	52

GPS_POS_STATUS definition

Bit	Type	Name	Description
[0-5]	Enum	SBG_ECOM_GPS_POS_STATUS	The raw GPS position status (see the 7 below).
[6-11]	Enum	SBG_ECOM_GPS_POS_TYPE	The raw GPS position type (see the 8 below).
12	Mask	SBG_ECOM_GPS_POS_GPS_L1_USED	Set to 1 if GPS L1 is used in the solution
13	Mask	SBG_ECOM_GPS_POS_GPS_L2_USED	Set to 1 if GPS L2 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 L1 is used in the solution
16	Mask	SBG_ECOM_GPS_POS_GLO_L2_USED	Set to 1 if GLONASS L2 is used in the 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 7: 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_OMNISTAR_HP	Omnistar HP position.
9	SBG_ECOM_POS_OMNISTAR_XP	Omnistar XP position.
10	SBG_ECOM_POS_OMNISTAR_G2	Omnistar G2 position (XP + GLonass).

Table 8: Raw GPS position type enumeration

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

2.3.4.5. SBG_ECOM_LOG_GPS1_HDT (15)

Provides raw GPS true heading data from a dual antenna GPS system.

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

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
				Total size	26

GPS_HDT_STATUS definition

Bit	Type	Name	Description
[0-5]	Enum	SBG_ECOM_GPS_HDT_STATUS	The raw GPS true heading status (see the 9 below).

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_INSUFFISENT_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 9: Raw GPS true heading status enumeration

2.3.4.6. SBG_ECOM_LOG_ODO_VEL (19)

Provides raw Odometer velocity.

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

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.

2.3.4.7. SBG_ECOM_LOG_PRESSURE (31)

Altimeter log provides altitude above reference level & pressure.

Altitude is referenced to a standard 1013 hPa.

Field	Description	Unit	Scaling	Format	Size	Offset
TIME_STAMP	Time since sensor is powered up	µs	-	uint32	4	0
ALTIMETER_STATUS	Altimeter status	-	-	uint16	2	4
PRESSURE	Pressure measured by the sensor	Pa	1	float	4	6
ALTITUDE	Altitude computed from altitude	m	1	float	4	10
					Total size	14

ALTIMETER_STATUS definition

Bit	Name	Description
0 (LSB)	SBG_ECOM_PRESSURE_VALID	Set to 1 if the altimeter equipment was correctly initialized,
1	SBG_ECOM_ALTITUDE_VALID	Set to 1 if the altitude output is valid

2.3.5. Miscellaneous logs

2.3.5.1. SBG_ECOM_LOG_EVENT_A/B/C/D (24, 25, 26, 27)

The Ellipse can detect events markers at up to 1 kHz on Sync In A, Sync In B, Sync In C, Sync In D input signals. For each input synchronization signal, the Ellipse 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
- 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.

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 Ellipse series supports 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.

3. NMEA Protocol description

The Ellipse AHRS and Ellipse INS series provide a NMEA support for both aiding input and data output.

3.1. NMEA sentences format

The NMEA sentences implemented in the Ellipse 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 Ellipse, 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 '-' is prepended to represent a negative decimal number.

Example

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

3.2. SBG_ECOM_CLASS_LOG_NMEA_0

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

NMEA logs contain Kalman filtered navigation, velocity and attitude data. External NMEA aiding data are not used to generate these logs.

3.2.1. Quality indicators

NMEA messages provide different quality indicators. The following table explains on which criterion the Ellipse sensor will provide an indicator or another one.

Quality indicators are not based on GPS input, and only rely on the estimated position accuracy. You can find in the table below the position accuracy threshold for each quality value.

Horizontal Accuracy	100m	10m	1m	0.6m	0.2m	
GGA quality indicator NMEA meaning	0 Invalid Fix	6 Dead Reckoning	1 Standalone Fix	2 DGPS Fix	5 Floating RTK	4 Fixed RTK
RMC Pos. mode NMEA meaning	N Invalid Sol.	E Dead Reckoning	A Standalone Fix	D DGPS Fix	F Floating RTK	R Fixed RTK
RMC Navigational status	V Not valid	C Caution	S Caution	S Safe	S Safe	S Safe

3.2.2. NMEA Logs list

Name (MSG ID)	Description
SBG_ECOM_LOG_NMEA_GGA (0x00)	Latitude, Longitude, Altitude, Quality indicator
SBG_ECOM_LOG_NMEA_RMC (0x01)	Latitude, Longitude, velocity, course over ground
SBG_ECOM_LOG_NMEA_ZDA (0x02)	UTC Time
SBG_ECOM_LOG_NMEA_HDT (0x03)	Heading (True)
SBG_ECOM_LOG_NMEA_PRDID (0x04)	RDI proprietary sentence. Pitch, Roll, Heading

3.2.3. GGA message

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

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

Message format

```
$##GGA,hhmmss.ss,ddmm.mmmmm,N,ddd.mm.mmmmm,E,i,ii,ff.f,ffff.fff,M,fff.fff,M,,*cs<CR><LF>
```

Field	Name	Format	Description
0	$##GGA	string	Message ID – GGA frame
1	Time	hhmmss.ss	UTC Time, current time
2	Latitude	ddmm.mmmmm	Latitude: degree + minutes
3	N/S	char	North / South indicator
4	Longitude	dddmm.mmmmm	Longitude: degree + minutes
5	E/W	char	East / West indicator
6	Quality	i	Fix status (see definition below)
7	SV used	ii	Number of satellites used in solution
8	Horizontal DOP	ff.f	Horizontal dilution of precision, 1 (ideal) to > 20 (poor)
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

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

3.2.4. RMC message

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

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

```
$##RMC,hhmmss.ss,A,ddmm.mmmmm,N,dddmm.mmmmm,E,fff.f,fff.f,ddmmyy,fff.ff,E,R,S*cs<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.mmmmm	Latitude: degree + minutes
4	N/S	char	North / South indicator
5	longitude	dddmm.mmmmm	Longitude: degree + minutes
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 table below).
13	navStatus	char	Navigational status indicator (see table below).
14	Check sum	*cs	Xor of all previous bytes except $
15	End of frame	<CR><LF>	Carriage return and line feed

Message example

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

3.2.5. ZDA message

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

This message contains UTC time and date information.

Message format

```
$##ZDA,hhmmss.ss,day,month,year,ltzh,ltzn*cs<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)
11	Ltzh	0	Local zone minutes (not supported, fixed 00)
13	Check sum	*cs	Xor of all previous bytes except $
14	End of frame	<CR><LF>	Carriage return and line feed

Message example

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

3.2.6. HDT Message

Name in sbgECom convention (msg ID): **SBG_ECOM_LOG_NMEA_HDT (0x03)**

The message output the Ellipse 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

```
$##HDT,fff.ff,T*cs<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

Message example

```
$GPHDT,fff.ff,T*cs<CR><LF>
```

3.2.7. PRDID

Name in sbgECom convention (msg ID): **SBG_ECOM_LOG_NMEA_PRDID (0x04)**

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,+fff.ff,-fff.ff,fff.ff*cs<CR><LF>
```

Field	Name	Format	Description
0	$PRDID	string	Message ID – RDI proprietary heading, pitch and roll
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

Message example

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

3.3. SBG_ECOM_CLASS_LOG_THIRD_PARTY_0

This class contains Third party output logs.

3.3.1. Third party Logs list

Name (MSG ID)	Description
SBG_ECOM_THIRD_PARTY_TSS1 (0x00)	Latitude, Longitude, Altitude, Quality indicator

3.3.2. 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/s^2, with a range from zero to 9.81 m/s^2
AAAA	Vertical acceleration (hex value - 2's complement), in 0.0625 cm/s^2, with a range of -20.48 to +20.48 m/s^2
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 Ellipse 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. Input protocols

4.1.1. 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 as well as vertical velocity
- RMC is used to handle horizontal velocity in NED frame aiding.
- 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 Ellipse NMEA integration manual to quickly connect a NMEA GNSS receiver to the Ellipse.

4.1.2. UBX protocol

The Ublox UBX binary protocol can be used to provide best performance when connecting an external Ublox GPS/GNSS receiver to an Ellipse-E.

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. CAN protocol specifications

5.1.1. 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

5.2. Output Logs

Following CAN messages are provided by the Ellipse.

Name (log ID)	Description
SBG_ECAN_LOG_STATUS_01 (0x100) SBG_ECAN_LOG_STATUS_02 (0x101) SBG_ECAN_LOG_STATUS_03 (0x102)	Status general, clock, com aiding, solution, heave
SBG_ECAN_LOG_IMU_INFO (0x120) SBG_ECAN_LOG_IMU_ACCEL (0x121) SBG_ECAN_LOG_IMU_GYRO (0x122) SBG_ECAN_LOG_IMU_DELTA_VEL (0x123) SBG_ECAN_LOG_DELTA_ANGLES (0x124)	Includes IMU status, acc., gyro, temp delta speeds and delta angles values
SBG_ECAN_LOG_EKF_INFO (0x130) SBG_ECAN_LOG_EKF_QUAT (0x131) SBG_ECAN_LOG_EKF_EULER (0x132) SBG_ECAN_LOG_EKF_ORIENTATION_ACC (0x133)	Includes roll, pitch, yaw, or quaternion output and their accuracies on each axis
SBG_ECAN_LOG_EKF_POS (0x134) SBG_ECAN_LOG_EKF_ALTITUDE (0x135) SBG_ECAN_LOG_EKF_POS_ACC (0x136) SBG_ECAN_LOG_EKF_VEL (0x137) SBG_ECAN_LOG_EKF_VEL_ACC (0x138)	Position and velocities in NED coordinates with the accuracies on each axis
SBG_ECAN_LOG_SHIP_MOTION_INFO (0x140) SBG_ECAN_LOG_SHIP_MOTION_0_0 (0x141) SBG_ECAN_LOG_SHIP_MOTION_0_1 (0x145)	Heave, surge and sway and accelerations on each axis for up to 4 points
SBG_ECAN_LOG_UTC_0 (0x110) SBG_ECAN_LOG_UTC_1 (0x111)	Provides UTC time reference
SBG_ECAN_LOG_MAG_0 (0x150) SBG_ECAN_LOG_MAG_1 (0x151) SBG_ECAN_LOG_MAG_2 (0x152)	Magnetic data with associated accelerometer on each axis
SBG_ECAN_LOG_GPS1_VEL_INFO (0x170) SBG_ECAN_LOG_GPS1_VEL (0x171) SBG_ECAN_LOG_GPS1_VEL_ACC (0x172) SBG_ECAN_LOG_GPS1_COURSE (0x173)	GPS velocities from primary or secondary GPS receiver
SBG_ECAN_LOG_GPS1_POS_INFO (0x174) SBG_ECAN_LOG_GPS1_POS (0x175) SBG_ECAN_LOG_GPS1_ALT (0x176) SBG_ECAN_LOG_GPS1_POS_ACC (0x177)	GPS positions from primary or secondary GPS receiver
SBG_ECAN_LOG_GPS1_HDT_INFO (0x178) SBG_ECAN_LOG_GPS1_HDT (0x179)	GPS true heading from dual antenna system
SBG_ECAN_LOG_ODOMETER_INFO (0x160) SBG_ECAN_LOG_ODOMETER_VELOCITY (0x161)	Provides odometer velocity
SBG_ECAN_LOG_EVENT_INFO_A/B/C/D (0x200, 0x202, 0x204, 0x206) SBG_ECAN_LOG_EVENT_TIME_A/B/C/D (0x201, 0x203, 0x205, 0x207)	Event markers sent when events are detected on a sync in pin

5.2.1. 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.

5.2.1.1. SBG_ECAN_LOG_STATUS_01 (0x100)

Field	Description	Unit	Format	Size	Offset
TIME_STAMP	Time since sensor is powered up.	µs	uint32	4	0
GENERAL_STATUS	Aiding status bit-mask.	-	uint16	2	4
CLOCK_STATUS	Clock status bit-mask.	-	uint16	2	6
				Total size	8

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_ECAN_GENERAL_MAIN_POWER_OK	Mask	Set to 1 when main power supply is OK.
1	SBG_ECAN_GENERAL_IMU_POWER_OK	Mask	Set to 1 when IMU power supply is OK.
2	SBG_ECAN_GENERAL_GPS_POWER_OK	Mask	Set to 1 when GPS power supply is OK.
3	SBG_ECAN_GENERAL_SETTINGS_OK	Mask	Set to 1 if settings where correctly loaded
4	SBG_ECAN_GENERAL_TEMPERATURE_OK	Mask	Set to 1 when temperature is within specified limits.
5	SBG_ECAN_GENERAL_DATALOGGER_OK	Mask	Set to 1 when the data-logger is working correctly.

CLOCK_STATUS definition

Provides status on the clock stability, error and synchronization.

Bit	Name	Type	Description
0	SBG_ECAN_CLOCK_STABLE_INPUT	Mask	Set to 1 when a clock input can be used to synchronize the internal clock.
[1-4]	SBG_ECAN_CLOCK_STATUS	Enum	Define the internal clock estimation status (see the table 10 below).
5	SBG_ECAN_CLOCK_UTC_SYNC	Mask	Set to 1 if UTC time is synchronized with a PPS
[6-9]	SBG_ECAN_CLOCK_UTC_STATUS	Enum	Define the UTC validity status (see the table 11 below).

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

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

Table 10: Clock Status enumeration

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

Table 11: UTC time status enumeration

5.2.1.2. SBG_ECAN_LOG_STATUS_02 (0x101)

Field	Description	Format	Size	Offset
COM_STATUS	Com status bit-mask.	uint32	4	0
AIDING_STATUS	Aiding status bit-mask.	uint32	4	4
			Total size	8

COM_STATUS definition

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

Bit	Name	Type	Description
0	SBG_ECAN_PORTA_VALID	Mask	Set to 0 in case of low level communication error.
1	SBG_ECAN_PORTB_VALID	Mask	Set to 0 in case of low level communication error.
2	SBG_ECAN_PORTC_VALID	Mask	Set to 0 in case of low level communication error.
3	SBG_ECAN_PORTD_VALID	Mask	Set to 0 in case of low level communication error.
5	SBG_ECAN_PORTA_RX_OK	Mask	Set to 0 in case of saturation on PORT A input
6	SBG_ECAN_PORTA_TX_OK	Mask	Set to 0 in case of saturation on PORT A output
7	SBG_ECAN_PORTB_RX_OK	Mask	Set to 0 in case of saturation on PORT B input
8	SBG_ECAN_PORTB_TX_OK	Mask	Set to 0 in case of saturation on PORT B output
9	SBG_ECAN_PORTC_RX_OK	Mask	Set to 0 in case of saturation on PORT C input
10	SBG_ECAN_PORTC_TX_OK	Mask	Set to 0 in case of saturation on PORT C output
11	SBG_ECAN_PORTD_RX_OK	Mask	Set to 0 in case of saturation on PORT D input
12	SBG_ECAN_PORTD_TX_OK	Mask	Set to 0 in case of saturation on PORT D output
25	SBG_ECAN_CAN_RX_OK	Mask	Set to 0 in case of saturation on CAN Bus output buffer
26	SBG_ECAN_CAN_TX_OK	Mask	Set to 0 in case of saturation on CAN Bus input buffer
27-29	SBG_ECAN_CAN_BUS	Enum	Define the CAN Bus status (see the 12 below).

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

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

Table 12: CAN Bus status enumeration

AIDING_STATUS definition

Tells which aiding data is received.

Bit	Name	Type	Description
0	SBG_ECAN_AIDING_GPS1_POS_RECV	Mask	Set to 1 when valid GPS 1 position data is received
1	SBG_ECAN_AIDING_GPS1_VEL_RECV	Mask	Set to 1 when valid GPS 1 velocity data is received
2	SBG_ECAN_AIDING_GPS1_HDT_RECV	Mask	Set to 1 when valid GPS 1 true heading data is received
3	SBG_ECAN_AIDING_GPS1_UTC_RECV	Mask	Set to 1 when valid GPS 1 UTC time data is received
4	SBG_ECAN_AIDING_GPS2_POS_RECV	Mask	Set to 1 when valid GPS 2 position data is received
5	SBG_ECAN_AIDING_GPS2_VEL_RECV	Mask	Set to 1 when valid GPS 2 velocity data is received
6	SBG_ECAN_AIDING_GPS2_HDT_RECV	Mask	Set to 1 when valid GPS 2 true heading data is received
7	SBG_ECAN_AIDING_GPS2_UTC_RECV	Mask	Set to 1 when valid GPS 2 UTC time data is received
8	SBG_ECAN_AIDING_MAG_RECV	Mask	Set to 1 when valid Magnetometer data is received
9	SBG_ECAN_AIDING_ODO_RECV	Mask	Set to 1 when Odometer pulse is received

5.2.1.3. SBG_ECAN_LOG_STATUS_03 (0x102)

Field	Description	Format	Size	Offset
SOLUTION_STATUS	Solution status bit-mask.	uint32	4	0
HEAVE_STATUS	Heave status bit-mask.	uint16	2	4
			Total size	6

HEAVE_STATUS definition

Tells if the heave is valid and if the heave computation is aided by velocity and heading data.

Bit	Name	Description
0 (LSB)	SBG_ECAN_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_ECAN_HEAVE_VEL_AIDED	Set to 1 if heave output is compensated for transient accelerations

SOLUTION_STATUS definition

Provides 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_ECAN_SOLUTION_MODE	Defines the Kalman filter computation mode (see the table 13 below)
4 (LSB)	SBG_ECAN_SOL_ATTITUDE_VALID	Set to 1 if Attitude data is reliable (Roll/Pitch error < 0,5°)
5	SBG_ECAN_SOL_HEADING_VALID	Set to 1 if Heading data is reliable (Heading error < 1°)
6	SBG_ECAN_SOL_VELOCITY_VALID	Set to 1 if Velocity data is reliable (velocity error < 1.5 m/s)
7	SBG_ECAN_SOL_POSITION_VALID	Set to 1 if Position data is reliable (Position error < 10m)
8	SBG_ECAN_SOL_VERT_REF_USED	Set to 1 if vertical reference is used in solution (data used and valid since 3s)
9	SBG_ECAN_SOL_MAG_REF_USED	Set to 1 if magnetometer is used in solution (data used and valid since 3s)
10	SBG_ECAN_SOL_GPS1_VEL_USED	Set to 1 if GPS velocity is used in solution (data used and valid since 3s)
11	SBG_ECAN_SOL_GPS1_POS_USED	Set to 1 if GPS Position is used in solution (data used and valid since 3s)
12	SBG_ECAN_SOL_GPS1_COURSE_USED	Set to 1 if GPS Course is used in solution (data used and valid since 3s)
13	SBG_ECAN_SOL_GPS1_HDT_USED	Set to 1 if GPS True Heading is used in solution (data used and valid since 3s)
14	SBG_ECAN_SOL_GPS2_VEL_USED	Set to 1 if GPS2 velocity is used in solution (data used and valid since 3s)
15	SBG_ECAN_SOL_GPS2_POS_USED	Set to 1 if GPS2 Position is used in solution (data used and valid since 3s)
16	SBG_ECAN_SOL_GPS2_COURSE_USED	Set to 1 if GPS2 Course is used in solution (data used and valid since 3s)
17	SBG_ECAN_SOL_GPS2_HDT_USED	Set to 1 if GPS2 True Heading is used in solution (data used and valid since 3s)
18	SBG_ECAN_SOL_ODO_USED	Set to 1 if Odometer is used in solution (data used and valid since 3s)
19	SBG_ECAN_SOL_DVL_BT_USED	Set to 1 if DVL Bottom Tracking is used in solution (data used and valid since 3s)
20	SBG_ECAN_SOL_DVL_WT_USED	Set to 1 if DVL Water Layer is used in solution (data used and valid since 3s)

You can find below the different available solution modes:

Value	Name	Description
0	SBG_ECAN_SOL_MODE_UNINITIALIZED	The Kalman filter is not initialized and the returned data are all invalid.
1	SBG_ECAN_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_ECAN_SOL_MODE_AHRS	A heading reference is available, the Kalman filter provides full orientation but navigation data drift freely.
3	SBG_ECAN_SOL_MODE_NAV_VELOCITY	The Kalman filter computes orientation and velocity. Position is freely integrated from velocity estimation.
4	SBG_ECAN_SOL_MODE_NAV_POSITION	Nominal mode, the Kalman filter computes all parameters (attitude, velocity, position). Absolute position is provided.

Table 13: Solution modes enumeration

5.2.2. UTC time output

Provides UTC time reference. This frame also provides a time correspondence between Ellipse TIME_STAMP value and actual UTC Time.

5.2.2.1. SBG_ECAN_LOG_UTC_0 (0x110)

Field	Description	Unit	Format	Size	Offset
TIME_STAMP	Time since sensor is powered up.	µs	uint32	4	0
GPS_TOW	GPS Time of week.	ms	uint32	4	4
			Total size		8

5.2.2.2. SBG_ECAN_LOG_UTC_1 (0x111)

Field	Description	Scaling	Unit	Format	Size	Offset
YEAR	Year within the century (e.g. '10' means 2010)	1	year	uint8	1	0
MONTH	Month in Year [1 ... 12]	1	month	uint8	1	1
DAY	Day in Month [1 ... 31]	1	d	uint8	1	2
HOUR	Hour in day [0 ... 23]	1	h	uint8	1	3
MIN	Minute in hour [0 ... 59]	1	min	uint8	1	4
SEC	Second in minute [0 ... 60] Note 60 is when a leap second is added.	1	s	uint8	1	5
TENTHMS	Tenths of a millisecond in second.	10^{-4}	s	uint16	2	6
				Total size		8

5.2.3. Inertial Data output

5.2.3.1. SBG_ECAN_LOG_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:

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

Bit	Name	Description
0 (LSB)	SBG_ECAN_IMU_COM_OK	Set to 1 if the communication with the IMU is ok.
1	SBG_ECAN_IMU_STATUS_BIT	Set to 1 if internal IMU passes Built In Test (Calibration, CPU)
2	SBG_ECAN_IMU_ACCEL_X_BIT	Set to 1 if accelerometer X passes Built In Test
3	SBG_ECAN_IMU_ACCEL_Y_BIT	Set to 1 if accelerometer Y passes Built In Test
4	SBG_ECAN_IMU_ACCEL_Z_BIT	Set to 1 if accelerometer Z passes Built In Test
5	SBG_ECAN_IMU_GYRO_X_BIT	Set to 1 if gyroscope X passes Built In Test
6	SBG_ECAN_IMU_GYRO_Y_BIT	Set to 1 if gyroscope Y passes Built In Test
7	SBG_ECAN_IMU_GYRO_Z_BIT	Set to 1 if gyroscope Z passes Built In Test
8	SBG_ECAN_IMU_ACCELS_IN_RANGE	Set to 1 if accelerometers are within operating range
9	SBG_ECAN_IMU_GYROS_IN_RANGE	Set to 1 if gyroscopes are within operating range

5.2.3.2. SBG_ECAN_LOG_IMU_ACCEL (0x121)

Fully calibrated accelerometers values in meters per second squared.

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

5.2.3.3. SBG_ECAN_LOG_IMU_GYRO (0x122)

Fully calibrated gyroscopes values in radians per second. Multiply each component by 10^{-3} to get the value in radians per second.

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

5.2.3.4. SBG_ECAN_LOG_IMU_DELTA_VEL (0x123)

1KHz Sculling integration output. Delivers delta velocities in body coordinate frame.

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

5.2.3.5. SBG_ECAN_LOG_DELTA_ANGLES (0x124)

Coning integration output from the 1kHz gyroscopes integration. These values should be used as the gyroscopes values and should be multiplied by 10^{-3} to get the value in radians per second.

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_Y	Coning integration Z.	10^{-3}	$rad.s^{-1}$	int16	2	4
				Total size		6

5.2.4. EKF output

5.2.4.1. SBG_ECAN_LOG_EKF_INFO (0x130)

Field	Description	Unit	Format	Size	Offset
TIME_STAMP	Time since sensor is powered up.	μs	uint32	4	0
			Total size		4

5.2.4.2. SBG_ECAN_LOG_EKF_QUAT (0x131)

Device 3d orientation represented with a normalized quaternion (q0, q1, q2, q3) expressed using frac16 numbers (frac16: 1 sign bit, 15 fractional bits).

Field	Description	Scaling	Format	Size	Offset
Q0	Orientation quaternion, q0 component.	32768^{-1}	int16	2	0
Q1	Orientation quaternion, q1 component.	32768^{-1}	int16	2	2
Q2	Orientation quaternion, q2 component.	32768^{-1}	int16	2	4
Q3	Orientation quaternion, q3 component.	32768^{-1}	int16	2	6
			Total size		8

5.2.4.3. SBG_ECAN_LOG_EKF_EULER (0x132)

Provides computed orientation in Euler angles format.

Field	Description	Scaling	Unit	Format	Size	Offset
ROLL	Roll angle.	10^{-4}	rad	int16	2	0
PITCH	Pitch angle.	10^{-4}	rad	int16	2	2
YAW	Yaw angle.	10^{-4}	rad	int16	2	4
				Total size		6

5.2.4.4. SBG_ECAN_LOG_EKF_ORIENTATION_ACC (0x133)

Provides estimated orientation standard deviation accuracy in Euler angles format.

Field	Description	Scaling	Unit	Format	Size	Offset
ROLL_ACC	1σ Roll angle accuracy.	10^{-4}	rad	uint16	2	0
PITCH_ACC	1σ Pitch angle accuracy.	10^{-4}	rad	uint16	2	2
YAW_ACC	1σ Yaw angle accuracy.	10^{-4}	rad	uint16	2	4
				Total size		6

5.2.4.5. SBG_ECAN_LOG_EKF_POS (0x134)

Latitude and longitude enhanced using inertial data and expressed in the WGS84 format.

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

5.2.4.6. SBG_ECAN_LOG_EKF_ALTITUDE (0x135)

Altitude in millimeters expressed either in height above ellipsoid or mean sea level according to the GPS configuration. Horizontal and vertical accuracies estimated by the Kalman filter are expressed in centimeters.

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

5.2.4.7. SBG_ECAN_LOG_EKF_POS_ACC (0x136)

Position accuracy from internal Kalman filter.

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

5.2.4.8. SBG_ECAN_LOG_EKF_VEL (0x137)

North, East and Down velocities, from Kalman filter output.

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

5.2.4.9. SBG_ECAN_LOG_EKF_VEL_ACC (0x138)

North, East and Down velocity accuracy, from Kalman filter output.

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

5.2.5. Heave output

5.2.5.1. SBG_ECAN_LOG_SHIP_MOTION_INFO (0x140)

Returns ship motion status as well as heave period at main monitoring point.

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
STATUS	Ship motion status	-	-	uint16	2	6
				Total size		8

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_SURGE_SWAY_AVAILABLE	Set to 1 if surge and sway channels are provided in this output.
3	SBG_ECOM_PERIOD_AVAILABLE	Set to 1 if the swell period is provided in this output
4	SBG_ECOM_PERIOD_VALID	Set to 1 if the period returned is assumed to be valid or not.

5.2.5.2. SBG_ECAN_LOG_SHIP_MOTION_0_0 (0x141)

Returns surge, sway & heave at monitoring point x (0 for main monitoring point, 1/2/3 for the others).

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

5.2.5.3. SBG_ECAN_LOG_SHIP_MOTION_0_1 (0x145)

Returns ship accelerations at monitoring point x (0 for main monitoring point, 1/2/3 for the others).

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

5.2.5.4. SBG_ECAN_LOG_SHIP_MOTION_0_2(0x149)

Returns ship velocity at monitoring point x (0 for main monitoring point, 1/2/3 for the others).

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

5.2.6. Magnetometer output

5.2.6.1. SBG_ECAN_LOG_MAG_0 (0x150)

Field	Description	Unit	Format	Size	Offset
TIME_STAMP	Time since sensor is powered up.	µs	uint32	4	0
MAG_STATUS	Magnetometer status bit-mask.	-	uint16	2	4
			Total size		6

MAG_STATUS definition

Bit	Name	Description
0(LSB)	SBG_ECAN_MAG_MAG_X_BIT	Set to 1 if the magnetometer X has passed the self test.
1	SBG_ECAN_MAG_MAG_Y_BIT	Set to 1 if the magnetometer Y has passed the self test.
2	SBG_ECAN_MAG_MAG_Z_BIT	Set to 1 if the magnetometer Z has passed the self test.
3	SBG_ECAN_MAG_ACCEL_X_BIT	Set to 1 if the accelerometer X has passed the self test.
4	SBG_ECAN_MAG_ACCEL_Y_BIT	Set to 1 if the accelerometer Y has passed the self test.
5	SBG_ECAN_MAG_ACCEL_Z_BIT	Set to 1 if the accelerometer Z has passed the self test.
6	SBG_ECAN_MAG_MAGS_IN_RANGE	Set to 1 if magnetometer is not saturated
7	SBG_ECAN_MAG_ACCELS_IN_RANGE	Set to 1 if accelerometer is not saturated
8	SBG_ECAN_MAG_CALIBRATION_OK	Set to 1 if magnetometer seems to be calibrated

5.2.6.2. SBG_ECAN_LOG_MAG_1 (0x151)

Fully calibrated and normalized magnetometers values in arbitrary units. Multiply each component by 10^{-3} to get the value in the A.U. If well calibrated, the norm of the magnetic vector should equal 1.

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

5.2.6.3. SBG_ECAN_LOG_MAG_2 (0x152)

Provides associated accelerometer values, in case of internal magnetometer, internal accelerometer values are returned.

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

5.2.7. Odometer output

5.2.7.1. SBG_ECAN_LOG_ODOMETER_INFO (0x160)

Odometer status information and Time since reset. This time is not necessary aligned with main loop as it dates the actual odometer data.

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
			Total size		6

ODO_STATUS definition

Bit	Name	Description
0 (LSB)	SBG_ECAN_ODO_REAL_MEAS	Set to 1 if this log comes from a real pulse measurement or from a timeout.

5.2.7.2. SBG_ECAN_LOG_ODOMETER_VELOCITY (0x161)

Odometers raw velocities only computed using detected pulses and odometer pulses per meter setting.

Field	Description	Unit	Format	Size	Offset
VELOCITY	Velocity in odometer direction.	$m.s^{-1}$	float	4	0
			Total size		4

5.2.8. GPS 1 and 2 outputs

5.2.8.1. SBG_ECAN_LOG_GPS1_VEL_INFO (0x170)

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

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 bit-mask.	-	uint32	4	4
			Total size		8

GPS_VEL_STATUS definition

Bit	Type	Name	Description
[0-5]	Enum	SBG_ECAN_GPS_VEL_STATUS	The raw GPS velocity status (see the 14 below).
[6-11]	Enum	SBG_ECAN_GPS_VEL_TYPE	The raw GPS velocity type (see the 15 below).

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

Value	Name	Description
0	SBG_ECAN_VEL_SOL_COMPUTED	A valid solution has been computed.
1	SBG_ECAN_VEL_INSUFFISENT_OBS	Not enough valid SV to compute a solution.
2	SBG_ECAN_VEL_INTERNAL_ERROR	An internal error has occurred.
3	SBG_ECAN_VEL_LIMIT	Velocity limit exceeded.

Table 14: Raw GPS velocity status enumeration

Value	Name	Description
0	SBG_ECAN_VEL_NO_SOLUTION	No valid velocity solution available.
1	SBG_ECAN_VEL_UNKNOWN_TYPE	An unknown solution type has been computed.
2	SBG_ECAN_VEL_DOPPLER	A Doppler velocity has been computed.
3	SBG_ECAN_VEL_DIFFERENTIAL	A velocity has been computed between two positions.

Table 15: Raw GPS velocity type enumeration

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

5.2.8.2. SBG_ECAN_LOG_GPS1_VEL (0x171)

Provide raw GNSS velocity from primary or secondary GNSS receiver.

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

5.2.8.3. SBG_ECAN_LOG_GPS1_VEL_ACC (0x172)

Provide raw GNSS velocity accuracy from primary or secondary GNSS receiver.

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

5.2.8.4. SBG_ECAN_LOG_GPS1_COURSE (0x173)

Provide raw GNSS course data from primary or secondary GNSS receiver.

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

5.2.8.5. SBG_ECAN_LOG_GPS1_POS_INFO (0x174)

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

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 bit-mask.	-	uint32	4	4
			Total size		8

GPS_POS_STATUS definition

Bit	Type	Name	Description
[0-5]	Enum	SBG_ECAN_GPS_POS_STATUS	The raw GPS position status (see the table 16 below).
[6-11]	Enum	SBG_ECAN_GPS_POS_TYPE	The raw GPS position type (see the table 17 below).
12	Mask	SBG_ECAN_GPS_POS_GPS_L1_USED	Set to 1 if GPS L1 is used in the solution
13	Mask	SBG_ECAN_GPS_POS_GPS_L2_USED	Set to 1 if GPS L2 is used in the solution
14	Mask	SBG_ECAN_GPS_POS_GPS_L5_USED	Set to 1 if GPS L5 is used in the solution
15	Mask	SBG_ECAN_GPS_POS_GLO_L1_USED	Set to 1 if GLONASS L1 is used in the solution
16	Mask	SBG_ECAN_GPS_POS_GLO_L2_USED	Set to 1 if GLONASS L2 is used in the solution

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

Value	Name	Description
0	SBG_ECAN_POS_SOL_COMPUTED	A valid solution has been computed.
1	SBG_ECAN_POS_INSUFFISENT_OBS	Not enough valid SV to compute a solution.
2	SBG_ECAN_POS_INTERNAL_ERROR	An internal error has occurred.
3	SBG_ECAN_POS_HEIGHT_LIMIT	The height limit has been exceeded.

Table 16: Raw GPS position status enumeration

Value	Name	Description
0	SBG_ECAN_POS_NO_SOLUTION	No valid solution available.
1	SBG_ECAN_POS_UNKNOWN_TYPE	An unknown solution type has been computed.
2	SBG_ECAN_POS_SINGLE	Single point solution position.
3	SBG_ECAN_POS_PSRDIFF	Standard Pseudorange Differential Solution (DGPS).
4	SBG_ECAN_POS_SBAS	SBAS satellite used for differential corrections.
5	SBG_ECAN_POS_OMNISTAR	Omnistar VBS Position (L1 sub-meter).
6	SBG_ECAN_POS_RTK_FLOAT	Floating RTK ambiguity solution (20 cm RTK).
7	SBG_ECAN_POS_RTK_INT	Integer RTK ambiguity solution (2 cm RTK).
8	SBG_ECAN_POS_OMNISTAR_HP	Omnistar HP position.
9	SBG_ECAN_POS_OMNISTAR_XP	Omnistar XP position.
10	SBG_ECAN_POS_OMNISTAR_G2	Omnistar G2 position (XP + GLonass).

Table 17: Raw GPS position type enumeration

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

5.2.8.6. SBG_ECAN_LOG_GPS1_POS (0x175)

Provide GNSS position from primary or secondary GNSS receiver.

Field	Description	Scaling	Unit	Format	Size	Offset
LATITUDE	Latitude, positive Noth.	10^{-7}	°	int32	4	0
LONGITUDE	Longitude, positive East.	10^{-7}	°	int32	4	4
				Total size		8

5.2.8.7. SBG_ECAN_LOG_GPS1_ALT (0x176)

Provide GNSS altitude from primary or secondary GNSS receiver.

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	2
				Total size		6

5.2.8.8. SBG_ECAN_LOG_GPS1_POS_ACC (0x177)

Provide GNSS position accuracy from primary or secondary GNSS receiver.

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
				Total size		6

5.2.8.9. SBG_ECAN_LOG_GPS1_HDT_INFO (0x178)

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

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
			Total size		6

GPS_HDT_STATUS definition

Bit	Type	Name	Description
[0-5]	Enum	SBG_ECAN_GPS_HDT_STATUS	The raw GPS true heading status (see the 18 below).

You can find below the GPS true heading status enumeration:

Value	Name	Description
0	SBG_ECAN_HDT_SOL_COMPUTED	A valid solution has been computed.
1	SBG_ECAN_HDT_INSUFFISENT_OBS	Not enough valid SV to compute a solution.
2	SBG_ECAN_HDT_INTERNAL_ERROR	An internal error has occurred.
3	SBG_ECAN_HDT_HEIGHT_LIMIT	The height limit has been exceeded.

Table 18: Raw GPS true heading status enumeration

5.2.8.10. SBG_ECAN_LOG_GPS1_HDT (0x179)

Provides raw GPS true heading data from a dual antenna GPS system.

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

5.2.9. User log event

The Ellipse can detect events markers at up to 1 kHz on Sync A, Sync B, Sync C, Sync D and Sync E input signals. For each input synchronization signal, the Ellipse 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.

5.2.9.1. SBG_ECAN_LOG_EVENT_INFO_A/B/C/D (0x200, 0x202, 0x204, 0x206)

Field	Description	Unit	Format	Size	Offset
TIME_STAMP	Time since sensor is powered up.	µs	uint32	4	0
EVENT_STATUS	Status bit-mask.	-	uint16	2	4
			Total size		6

EVENT_STATUS definition

Bit	Name	Description
0 (LSB)	SBG_ECAN_EVENT_OVERFLOW	Set to 1 if we have received events at a higher rate than 1 kHz.
1	SBG_ECAN_EVENT_OFFSET_0_VALID	Set to 1 if at least two events have been received.
2	SBG_ECAN_EVENT_OFFSET_1_VALID	Set to 1 if at least three events have been received.
3	SBG_ECAN_EVENT_OFFSET_2_VALID	Set to 1 if at least four events have been received.
4	SBG_ECAN_EVENT_OFFSET_3_VALID	Set to 1 if five events have been received.

5.2.9.2. SBG_ECAN_LOG_EVENT_TIME_A/B/C/D (0x201, 0x203, 0x205, 0x207)

Field	Description	Unit	Format	Size	Offset
TIME_OFFSET_0	Time offset for the second received event.	µs	uint16	2	0
TIME_OFFSET_1	Time offset for the third received event.	µs	uint16	2	2
TIME_OFFSET_2	Time offset for the fourth received event.	µs	uint16	2	4
TIME_OFFSET_3	Time offset for the fifth received event.	µs	uint16	2	6
			Total size		8

Note: The Ellipse series supports 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.

6. 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 Ellipse, we would be glad to help you, so please feel free to contact us. Please do not forget to mention your Ellipse Device ID (written on your Ellipse' label).

You can contact us by:

- Email: support@sbg-systems.com
- Phone: +33 1 80 88 45 00