IG Devices Serial Protocol Specifications

Document : IGDSPS.10
Revision : 10 - Nov 22, 2013
Covers Firmware V2.1

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

Email : support@sbg-systems.com
Phone : +33 (0)1 80 88 45 00

Revision history

Rev.	Date	Author	Information
10	Nov 22, 2013	Nicolas MICHEL	Corrected command ID SBG_VALIDATE_MP_BUFFER Added Heave period configuration in SBG_SET/GET_HEAVE_CONF Fixed typo SBG_RET_DEFAULT_OUTPUT_MASK Updated logo Added HEHDT, HEHDM, KVHEXT, PSXN outputs definition
9	May 28, 2012	Alexis GUINAMARD	Protocol update for IG-500 Firmware V2 Added a chapter for IG-20/30 specific commands Added Heave and delta angle outputs Added NMEA / ASCII protocol output definitions. Added commands: SBG_SAVE_SETTINGS Motion profile commands: SBG_SEND_MP_BUFFER, SBG_VALIDATE_MP_BUFFER, SBG_GET_MP_INFO SBG_SET/GET_RET_HEAVE_CONF SBG_SET/GET/RET_VIRTUAL_ODO_CONF SBG_SET/GET/RET_ADVANCED_OPTIONS SBG_SET/GET/RET_ASCII_OUTPUT_CONF Updated commands: SBG_SET/GET_ODO_CONFIG SBG_SEND_HEADING, VELOCITY, POSITION) SBG_RESTORE_DEFAULT_SETTINGS SBG_SET_LOW_POWER_MODE (0x1C) SBG_GET/SET_SYNC_OUT_CONF SYNC_IN_CONF SBG_SET_HEADING_SOURCE SBG_SET/GET/RET_USER_BUFFER Removed obsolete commands and configurations: Removed Permanent setting SET commands Removed SBG_SET/GET_VELOCITY_CONSTRAINTS Removed SBG_GET/SET_FILTER_ATTITUDE_ERRORS Removed Error log commands Removed Internal magnetometer calibration methods Fixed default Reference pressure value Fixed Gyro Temperatures Raw output definition. Error Codes updated
8	Mar 21, 2011	Alexis GUINAMARD	Various minor fixes Added IG-500E specific configurations and outputs Added Synchronization features Added odometer features Updated set user buffer command specifications Updated new protocol overflow handling
7	May 3, 2010	Alexis GUINAMARD	Added triggered output specifications Added manual magnetometers calibration commands Fixed velocity constraints commands Updated company address details Fixed internal magnetic calibration settings
6	Nov 16, 2009	Alexis GUINAMARD	Removed obsolete Filter frequency output Added device status specifications Added Velocity constraints management Added error log command Updated magnetometers calibration data in default/specific output Updated SBG_CALIB_GYRO_BIAS command Reorganized overall commands organization in documentation

Rev.	Date	Author	Information
5	Sep 21, 2009	Alexis GUINAMARD	Minor fixes in the protocol: Baro altitude unit (cm) Updated Temperature and gyro temperatures explanations. Updated SBG_GET_FILTER_FREQUENCIES for new firmware. Corrected SBG_OUTPUT_BARO_PRESSURE name.
4	Jul 15, 2009	Alexis GUINAMARD	Minor corrections on types used in outputs. Fixed wrong command number of SBG_GET_SV_INFOS command Updated some default settings in GPS options Added new features : User buffer, UTC Time reference and altitude above MSL or above Ellipsoid Updated Kalman filter frequency behavior Updated Time since reset explanation Updated SBG_SET_FILTER_FREQUENCIES
3	9. Feb. 2009	Raphaël SIRYANI	Correction for SBG_SET_OUTPUT_MODE and SBG_RET_OUTPUT_MODE. The output mode argument was declared as a uint32 instead of a uint8. Command definition corrected for SBG_GET_SPECIFIC_OUTPUT. Various corrections for Output Mask and Output Buffer. Document renamed and now fits with the complete IG devices line.
2	1. Dec. 2008	Raphaël SIRYANI	Added IG-500N support.
1	11. July 2008	Alexis GUINAMARD	First version of this document

Index

1. General information about protocol

1.1. Serial port configuration

This protocol is used either on a standard RS232, RS422, or on a TTL format. Transmission format is the following:

- Default speed : 115200 bps
- 8 bits data
- 1 stop bit
- No parity
- No control flow

1.2. Supported protocols

Two protocols can be used with the IG-Devices.

The main protocol is the binary protocol which is defined below. It allows all the configuration and data retrieving.

In addition to that binary protocol, an ASCII / NMEA protocol is implemented in output only in order to facilitate IG-Devices integration into existing systems. Using the main binary protocol is still necessary to configure the ASCII outputs.

This document will describe the IG-Devices binary protocol. Section 4 will cover ASCII/NMEA outputs.

Both binary and ASCII protocols can be used in the same time.

1.3. Frames format

All frames sent through the serial port have a common format, which is described below :

Field	SYNCX	STX	CMD	LEN (MSB)	LEN (LSB)	DATA	CRC (MSB)	CRC (LSB)	ETX
Size (bytes)	1	1	1	1	1	0 to 504	1	1	1
Description	Sync. byte	Start of Tx byte	Command	Length of the DATA section		Additional data	CRC		End of Tx byte
Value	0xFF	0x02	-	-	-	-	-	-	0x03

***Note 1 :** The LENGTH field contains the DATA section length. A 0 value implies that no DATA section is present in the frame. Maximum length value is 504 so that maximum frame size is 512.*

***Note 2 :**CRC is calculated on the [CMD ; LENGTH ; DATA] fields. You can see below an example in C used to calculate the CRC:*

```c
// Calculate a CRC for the specified buffer
uint16 calcCRC(const uint8 *pBuffer, uint16 bufferSize)
{
      uint16 poly = 0x8408;
      uint16 crc = 0;
      uint8 carry;
      uint8 i_bits;
      uint16 j;

      for (j=0; j<bufferSize; j++)
      {
            crc = crc ^ pBuffer[j];
            for (i_bits=0; i_bits<8; i_bits++)
            {
                  carry = crc & 1;
                  crc = crc / 2;
                  if (carry)
                  {
                        crc = crc^poly;
                  }
            }
      }
      return crc;
}
```

1.4. Data format, Endianness

The device can be configured to output data in Big endian or in Little endian. You can also choose to transmit real numbers in either float/double values or in fixed32/fixed64 values.

The factory default configuration is the following :

- Big endian bytes ordering
- Float/Double real numbers

Note : *The endianness setting only affects values stored in the DATA field of the frame.*

1.5. Basic communications with the device

The device has two types of communication : Normal mode which is basically a Question/Answer system, and Continuous mode, in which the device will output regularly its data such as orientation estimate.

In addition, IG-30G and IG-500N include also a triggered mode which the device can send automatically new data as soon as it arrives. It is useful with GPS data which are updated at a low rate. If a triggered output is set, the device is then able to send the new raw GPS position each time a new data is received.

1.5.1. Normal mode

In normal mode, the host asks a question to the device, and the device answers to that question. Two types of answers can be defined.

1.5.1.1. Acknowledge

An acknowledge is sent by the device as a standard answer, when no specific response is needed. User is informed of how were executed the command, with the use of an error code. The error code which is returned in the DATA field can have several values :

Error Code	Value	Description
`SBG_NO_ERROR`	0x00	The command could properly be executed
`SBG_ERROR`	0x01	Command could not be executed properly
`SBG_NULL_POINTER`	0x02	A pointer equaled NULL
`SBG_INVALID_CRC`	0x03	A frame with an invalid CRC were received
`SBG_INVALID_FRAME`	0x04	The frame send 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 to 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_CALIB_MAG_NOT_ENOUGH_POINTS`	0x0C	The calibration procedure could not end correctly due to insufficient point number.
`SBG_CALIB_MAG_INVALID_TAKE`	0x0D	Not enough points were available to perform magnetometers calibration
`SBG_CALIB_MAG_SATURATION`	0x0E	The calibration procedure could not be properly executed due to noise measurement, or strong external and independent magnetic fields
`SBG_CALIB_MAG_POINTS_NOT_IN_A_PLANE`	0x0F	2D calibration procedure could not be performed because the rotation is not performed in a real plane
`SBG_INCOMPATIBLE_HARDWARE`	0x13	Hence valid; the command cannot be executed because of hardware incompatibility

The standard acknowledge frame format is the following :

Field	SYNCX	STX	CMD	LEN	DATA	CRC	ETX
Value	0xFF	0x02	SBG_ACK (0x01)	0x00 ; 0x01	SbgErrorCode (1 byte)	0xXX ; 0xXX	0x03

In case of a proper execution of the command , with SBG_NO_ERROR parameter, the frame sent will be:

```
FF 02 01 00 01 00 05 63 03
```

For example, if the error code is SBG_INVALID_PARAMETER, the frame sent will be (in hex):

```
FF 02 01 00 01 04 43 47 03
```

Standard error codes can be returned by all functions in case of a wrong transmission:

- `SBG_INVALID_CRC`, in case of an error during the CRC check,
- `SBG_INVALID_FRAME` when an invalid command is received. Possible reasons are:
  - The command does not exists
  - The parameters of the command do not have the right length → No interpretation can be done.

1.5.1.2. Specific response

For some commands, user needs to get an answer with more complex information, such as orientation data, or some configuration settings returned by the device. For each command, we will describe the specific response that is expected. If for some reason, the specific command could not be executed, an ACK containing an error code will be returned.

1.5.2. Continuous mode

When the device is configured to output data in continuous mode, it will send regularly a frame containing orientation/sensor data (as configured by user) without the user asking for it. This frame expects no answer from the user.

Normal mode is still functional when continuous mode is enabled and normal mode has always the priority compared to continuous frames. If user asks for specific tasks to the device, it may be possible that some continuous frames are skipped because of an overflow in the serial buffer. A high serial port speed will considerably decrease this risk. User can still check if frames have been skipped.

1.5.3. Triggered Output Mode

On the IG-500N and IG-30G, the triggered output mode is similar to the continuous mode but sends only newly computed data. It's useful for outputting raw GPS measurements or barometer data only when a new data is available.

The triggered output is generated by some triggers but is sent at the end of a computation loop.

Four trigger conditions can be configured. Each condition can be activated by one or many triggers. An output mask is configured corresponding to each trigger condition.

When several trigger conditions are activated, a single triggered output frame will be generated containing all data needed for good interpretation:

- A trigger mask containing all triggers that have been activated
- An output mask that defines which outputs have been included in the frame
- A standard output buffer

1.5.4. Types definition

1.5.4.1. Scalar types

Type	Description
`bool`	8 bits boolean. 0x00 is FALSE. 0X01 is TRUE
`uint8`	8 bits unsigned int
`int8`	8 bits signed int
`uint16`	16 bits unsigned int
`int16`	16 bits signed int
`uint32`	32 bits unsigned int
`int32`	32 bits signed int
`uint64`	64 bits unsigned int
`int64`	64 bits signed int
`float`	32 bits single floating point, standard IEEE 754 format
`double`	64 bits double floating point, standard IEEE 754 format
`fixed32`	32 bits fixed point number : 1 sign bit; 11 bits for integer part, 20 bits for fractional part
`fixed64`	64 bits fixed point number : 1 sign bit; 31 bits for integer part, 32 bits for fractional part
`real32`	32 bits number. Can be either float or fixed32, depending on the device configuration.
`real64`	64 bits number. Can be either double or fixed64, depending on the device configuration.

Note 1: *Scalar types can be represented in memory in Big or Little endian. The default configuration for endianness is Big endian.*

Note 2: *Default representation format of real numbers is float/double.*

1.5.4.2. Vectors and matrices

Vectors are stored in a 1D array.

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

Vector V is stored in memory this way :

V_0	V_1	V_2

All matrices are stored in a 1D array. They are expressed in vector column format.
For example, with a $M_{3\times 3}$ matrix :

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

M is stored in memory this way:

U_0	U_1	U_2	V_1	V_2	V_3	W_0	W_1	W_2

1.6. Output mask and output buffer

1.6.1. Output mask

The IG protocol uses masks to set the format of the buffer that will be outputted by the device. User can activate or inhibit any of the following outputs.

It is then possible to pool all information needed in a unique frame.

Activation mask	Description	Order	Mask value
SBG_OUTPUT_QUATERNION	Estimate of attitude in quaternion form	1	0x00000001
SBG_OUTPUT_EULER	Estimate of attitude in Euler angles form	2	0x00000002
SBG_OUTPUT_MATRIX	Estimate of attitude in Matrix form	3	0x00000004
SBG_OUTPUT_GYROSCOPES	Calibrated values of gyroscopes	4	0x00000008
SBG_OUTPUT_ACCELEROMETERS	Calibrated values of accelerometers	5	0x00000010
SBG_OUTPUT_MAGNETOMETERS	Calibrated values of magnetometers	6	0x00000020
SBG_OUTPUT_TEMPERATURES	Calibrated values of temperatures	7	0x00000040
SBG_OUTPUT_GYROSCOPES_RAW	Raw values of gyroscopes	8	0x00000080
SBG_OUTPUT_ACCELEROMETERS_RAW	Raw values of accelerometers	9	0x00000100
SBG_OUTPUT_MAGNETOMETERS_RAW	Raw values of magnetometers	10	0x00000200
SBG_OUTPUT_TEMPERATURES_RAW	Raw values of temperatures	11	0x00000400
SBG_OUTPUT_TIME_SINCE_RESET	Time elapsed since the reset of the device	12	0x00000800
SBG_OUTPUT_DEVICE_STATUS	Device status Bit field	13	0x00001000
SBG_OUTPUT_GPS_POSITION	Raw GPS position in WGS84 format	14	0x00002000
SBG_OUTPUT_GPS_NAVIGATION	Raw GPS velocity in NED, heading	15	0x00004000
SBG_OUTPUT_GPS_ACCURACY	Raw GPS horizontal, vertical and heading accuracies	16	0x00008000
SBG_OUTPUT_GPS_INFO	Raw GPS information such as available satellites	17	0x00010000
SBG_OUTPUT_BARO_ALTITUDE	Barometric altitude referenced to a user defined value	18	0x00020000
SBG_OUTPUT_BARO_PRESSURE	Absolute pressure in pascals	19	0x00040000
SBG_OUTPUT_POSITION	Kalman enhanced 3d position	20	0x00080000
SBG_OUTPUT_VELOCITY	Kalman enhanced 3d velocity	21	0x00100000
SBG_OUTPUT_ATTITUDE_ACCURACY	Kalman estimated attitude accuracy	22	0x00200000
SBG_OUTPUT_NAV_ACCURACY	Kalman estimated position and velocity accuracy	23	0x00400000
SBG_OUTPUT_GYRO_TEMPERATURES	Calibrated internal gyro temperatures sensors output	24	0x00800000
SBG_OUTPUT_GYRO_TEMPERATURES_RAW	Raw internal gyro temperatures sensors	25	0x01000000
SBG_OUTPUT_UTC_TIME_REFERENCE	UTC time reference	26	0x02000000
SBG_OUTPUT_MAG_CALIB_DATA	Enable magnetometers Calibration data output	27	0x04000000
SBG_OUTPUT_GPS_TRUE_HEADING	Enable raw true heading output	28	0x08000000
SBG_OUTPUT_ODO_VELOCITY	Enable Odometer raw velocity output	29	0x10000000
SBG_OUTPUT_DELTA_ANGLES	Enable delta angle output from coning integration	30	0x20000000
SBG_OUTPUT_HEAVE	Enable Heave output	31	0x40000000

Table 1: Output masks with corresponding order and values

1.6.1.1. Orientation outputs

SBG_OUTPUT_QUATERNION

Estimate of the orientation quaternion. The four coefficients are `real32` numbers. (16 bytes)

Memory representation :

q_0	q_1	q_2	q_3

SBG_OUTPUT_EULER

Estimate of the orientation in Euler angles form. Roll, Pitch and Yaw are three `real32` numbers (12 bytes), and are expressed in radians.

Euler angles are organized this way in memory:

Roll	Pitch	Yaw

SBG_OUTPUT_MATRIX

Estimate of orientation in matrix form. Matrix coefficients are `real32` numbers (36 bytes).

Memory organization is :

R_0	R_1	R_2	R_3	R_4	R_5	R_6	R_7	R_8

1.6.1.2. Calibrated Sensors output

SBG_OUTPUT_GYROSCOPES

Fully calibrated and Kalman unbiased gyroscopes values. G_x, G_y and G_z are three `real32` numbers (12 bytes), expressed in *rad* . s^{-1}.

Gyroscopes values are organized this way:

G_x	G_y	G_Z

SBG_OUTPUT_DELTA_ANGLES

Coning integration at 1KHz output. These ω_x, ω_y, and ω_z are three `real32` numbers (12 bytes), expressed in *rad* . s^{-1}.

Gyroscopes values are organized this way:

ω_x	ω_y	ω_Z

SBG_OUTPUT_ACCELEROMETERS

Fully calibrated and Kalman unbiased accelerometers values. A_x, A_y and A_z are three `real32` numbers (12 bytes), expressed in *m* . s^{-2}.

Memory representation is:

A_x	A_y	A_Z

SBG_OUTPUT_MAGNETOMETERS

Fully calibrated and normalized magnetometers values. M_x, M_y and M_z are three `real32` numbers (12 bytes), expressed in an arbitrary unit.

Memory representation is:

M_x	M_y	M_Z

SBG_OUTPUT_TEMPERATURES

Temperature measured by the accelerometer/magnetometer external temperature sensor, and the internal ADC temperature sensor. The two numbers are `real32` (8 bytes) expressed in *°C*.

Memory organization:

T_0	T_1

SBG_OUTPUT_GYRO_TEMPERATURES

Temperature measured by the three internal temperature sensors of gyroscopes. The three numbers are `real32` (12 bytes) expressed in *°C*.

Memory organization :

T_0	T_1	T_2

1.6.1.3. Raw sensors output

SBG_OUTPUT_GYROSCOPES_RAW

Raw gyroscopes values. Corresponds to the ADC readings. These values are three `uint16` numbers (6 bytes).

Gyroscopes values are organized this way:

RG_x	RG_y	RG_Z

SBG_OUTPUT_ACCELEROMETERS_RAW

Raw accelerometers values. Corresponds to the ADC readings. These values are three `uint16` numbers (6 bytes).

Memory representation is:

RA_x	RA_y	RA_Z

SBG_OUTPUT_MAGNETOMETERS_RAW

Raw magnetometers values. Corresponds to the ADC readings. These 3 values are in `uint16` (6 bytes).

Memory representation is:

RM_x	RM_y	RM_Z

SBG_OUTPUT_TEMPERATURES_RAW

Raw temperature measured by the accelerometer/magnetometer external temperature sensor, and the internal ADC temperature sensor.

These values are two `uint16` numbers (4 bytes)

Memory organization :

RT_0	RT_1

SBG_OUTPUT_GYRO_TEMPERATURES_RAW

Raw Temperature measured by the three internal temperature sensors of gyroscopes.. The three numbers are `uint16` (6 bytes) .

Memory organization :

RT_0	RT_1	RT_2

1.6.1.4. Time related outputs

SBG_OUTPUT_TIME_SINCE_RESET

This output represent the time in ms elapsed since the last reset. This time information is measured at each beginning of a sampling and calculation loop. This value is a `uint32` integer

Memory organization is:

time

SBG_OUTPUT_UTC_TIME

This structure contains the following UTC time referenced information:

- Year: Add 2000 to get the current UTC year. Range [0 – 255]
- Month of year (UTC) – Range [1 – 12]
- Day of month (UTC) – Range [1 – 31]
- Hour of day (UTC) – Range [0 – 23]
- Minutes of hour (UTC) – Range [0 – 60]
- Seconds of minute (UTC) – Range [0 – 60]
- Nanoseconds of second (UTC) – Range [0 – 999 999 999]

Each information is in `uint8` format except nanoseconds which require `uint32`.

Memory representation:

year	month	day	hour	min	sec	nanoseconds

1.6.1.5. Raw GPS outputs

SBG_OUTPUT_GPS_POSITION

3D position expressed in the WGS84 format : latitude, longitude and height. These 32 bits numbers (total: 12 bytes) are expressed in `int32` , $1 \times 10^{-7\,\circ}$ unit (latitude and longitude) and in `int32`, *mm* height above ellipsoid for altitude.

Memory representation:

lat.	long.	alt.

SBG_OUTPUT_GPS_NAVIGATION

3D velocity in North East Down as measured by the GPS receiver, and Heading.

- Velocity is three `int32` expressed in *cm*.s^{-1}.
- Heading is expressed in `int32`, within [0;360°] , with 1×10^{-5} ° per LSB.

Memory representation:

V_n	V_e	V_d	H

SBG_OUTPUT_GPS_TRUE_HEADING

This output is only valid on IG-500E. When the external device handles the True Heading output, the raw data can be retrieved here with the corresponding accuracy.

- Heading is expressed in `int32`, within [0;360°] , with 1×10^{-5} ° per LSB.
- Accuracy is expressed in `int32`, within [0;360°] , with 1×10^{-5} ° per LSB.

Memory representation:

H	Acc

SBG_OUTPUT_GPS_ACCURACY

Horizontal, vertical, speed and heading accuracies. These four `uint32` (16 bytes) are expressed in mm, for horizontal and vertical accuracies, in *cm.s*$^{-1}$, for speed accuracy and in *1×10*$^{-5}$ ° for heading accuracy.

Memory representation:

A_{Hor}	A_{Vert}	A_{Speed}	$A_{Heading}$

SBG_OUTPUT_GPS_INFO

This frame contains some basic information about the GPS Status :

- GPS Time of the week in *ms* (`uint32`) ,
- GPS Fix information (`uint8`),
- Number of satellites used by GPS for navigation solution (`uint8`).

Memory representation:

time	GPSFlags	NbSat

GPSFlags is composed of the flags described below:

Bit	**(MSB) 7**	**6**	**5**	**4**	**3**	**2**	**1**	**0 (LSB)**
Field	`Ext Fix info`		`GPS_TRUE_HEAD_VALID`	`GPS_VALID_UTC`	`GPS_VALID_WKN`	`GPS_VALID_TOW`	`GPS Fix info`	
Desc.	**00**=GPS_STD_FIX **01**=GPS_DGPS_FIX **10**=GPS_FRTK_FIX **11**=GPS_RTK_FIX		**1** = Valid true heading data	**1** = Valid UTC data (leap seconds known)	**1** = Week number known	**1** = Time of week known.	**00**=SBG_NO_FIX **01**=SBG_TIME_ONLY **10**=SBG_2D_FIX **11**=SBG_3D_FIX	

1.6.1.6. Barometric sensor outputs

SBG_OUTPUT_BARO_ALTITUDE

Altitude calculated using pressure information, relative to a reference pressure, configured by user. Default reference pressure is 101325 *Pa*. Altitude is an `int32` integer, expressed in *cm*.

Memory representation :

altitude

SBG_OUTPUT_BARO_PRESSURE

Pressure measured by the pressure sensor. This `uint32` integer is expressed in Pascals.

Memory representation :

P

1.6.1.7. Kalman Navigation outputs

SBG_OUTPUT_POSITION

3D position expressed in the WGS84 format : latitude, longitude and height. These three `real64` (24 bytes) are expressed in *degrees* for latitude and longitude and in *meters* for altitude.

Memory representation:

lat.	long.	alt.

SBG_OUTPUT_VELOCITY

3D velocity in device coordinate frame. These three `real32` (12 bytes) are expressed in *m.s*$^{-1}$.

Memory representation:

V_x	V_y	V_z

SBG_OUTPUT_HEAVE

High pass filtered Altitude. This real32 value is expressed in *m.*

Heave

1.6.1.8. Kalman accuracies outputs

SBG_OUTPUT_ATTITUDE_ACCURACY

Kalman estimated attitude accuracy. This `real32` value is expressed in *rads*.

Memory representation:

accuracy

SBG_OUTPUT_NAV_ACCURACY

Kalman estimated position and velocity accuracy. These two `real32` (8 bytes) are expressed in *meters* for the position accuracy and in *m.s*$^{-1}$ for the velocity accuracy.

Memory representation:

$Pos_{accuracy}$	$Vel_{accuracy}$

1.6.1.9. Other outputs

SBG_OUTPUT_DEVICE_STATUS

Bit mask (32 bits) containing useful status data concerning device initialization and health. Some bits are updated at initialization, and other at run time.

Memory representation:

Status

The following bits are stored in the bit-mask:

Bit	Name	Description
LSB	SBG_CALIB_INIT_STATUS_MASK	Set to 1 if the calibration structure is well initialized
1	SBG_SETTINGS_INIT_STATUS_MASK	Set to 1 if the settings structure is well initialized
[2 – 4]	SBG_ACCEL_*_SELF_TEST_STATUS_MASK	Set to 1 if the X accelerometer has passed self test
5	SBG_ACCEL_RANGE_STATUS_MASK	Set to 1 if the readings of accelerometers do not exceed operating range.
[6 – 8]	SBG_GYRO_*_SELF_TEST_STATUS_MASK	Set to 1 if the X gyroscope has passed self test
9	SBG_GYRO_RANGE_STATUS_MASK	Set to 1 if the readings of gyroscope do not exceed operating range.
10	SBG_MAG_CALIBRATION_STATUS_MASK	Set to 1 if the magnetic field calibration looks OK
11	SBG_ALTI_INIT_STATUS_BIT_MASK	Set to 1 if altimeter could initialize
12	SBG_GPS_INIT_STATUS_BIT_MASK	Set to 1 if GPS receive could initialize properly
13	SBG_G_MEASUREMENT_VALID_MASK	Set to 1 if gravity is observable sufficiently for proper Kalman filter operation
14	SBG_HEADING_MEASUREMENT_VALID_MASK	Set to 1 if heading is observable sufficiently for proper Kalman filter operation
15	SBG_VEL_MEASUREMENT_VALID_MASK	Set to 1 if velocity is observable sufficiently for proper Kalman filter operation
16	SBG_POS_MEASUREMENT_VALID_MASK	Set to 1 if position is observable sufficiently for proper Kalman filter operation
17	SBG_UTC_VALID_MASK	Bit mask for GPS UTC Validation: Leap Seconds already known
18	SBG_UTC_ROUGH_ACCURACY_MASK	Bit mask for UTC time validation with a 0.25s accuracy
19	SBG_UTC_FINE_ACURACY_MASK	Bit mask for UTC time synchronization with < 1µs accuracy
20	SBG_PROTOCOL_OUTPUT_STATUS_MASK	Bit mask for output buffer saturation indication. Set to 1 if the output buffer is in normal operation. Set to 0 if a buffer saturation occurred.
[21 – 31]	-	Reserved – Set to 0

SBG_OUTPUT_MAG_CALIB_DATA

This 12 bytes buffer contains data used to calibrate hard and soft Iron effects.

Memory representation:

CD_0	CD_1	CD_2	CD_3	CD_4	CD_5	CD_6	CD_7	CD_8	CD_9	CD_{10}	CD_{11}

SBG_OUTPUT_ODO_VELOCITY

This output is only available on the IG-500E. It provides the odometer velocity on the two available channels: These two `real32` (8 bytes) are expressed in $m.s^{-1}$.

Memory representation:

ODO_1	ODO_2

1.6.2. Output buffer generation

Depending on the output mask, an output buffer will be generated by the device. The wanted outputs will be serialized in ascending order (order of each output can be seen in the Table 1:)

Considering the following output mask:

```
SBG_OUTPUT_QUATERNION | SBG_OUTPUT_BARO_PRESSURE | SBG_OUTPUT_GYROSCOPES |
SBG_OUTPUT_GPS_POSITION
```

The output buffer generated will be :

Name	**q_0**	**q_1**	**q_2**	**q_3**	**G_x**	**G_y**	**G_z**	**lat**	**long**	**Alt**	**P**
Size (bytes)	4	4	4	4	4	4	4	4	4	4	4
	16				12			12			4
	44										

2. List of commands

2.1. Configure your device

Most commands are stored in a volatile memory. Once the device is fully configured it's then possible to save the configuration to non volatile flash memory. Note that due to technical requirements, some commands automatically save parameters to flash memory and perform a device reboot. This is specified in the command definition.

2.1.1. General settings

2.1.1.1. Get device information

It is used to retrieve some information about the connected device: product code, device number, firmware revision, hardware revisions.

Command availability:

IG-20	IG30A	IG-30G	IG-500A	IG-500N	IG-500E
yes	yes	yes	yes	yes	yes

SBG_GET_INFOS (0x10)

This command has no input parameter. The device returns a specific answer : SBG_RET_INFOS (0x11). There is no possible error with this command.

Field	CMD	LEN	DATA					
Value	SBG_RET_INFO 0x11	44	Product code (string)	Device number (uint32)	Firmware rev. (uint32)	Calibration data rev. (uint32)	Main board hard. Rev. (uint32)	GPS/Com board rev. (uint32)
Size (bytes)	1	2	32	4	4	4	4	4

2.1.1.2. User ID

Commands used to set/get a User ID for the device. It makes it possible to identify the device.

Command availability:

IG-20	IG30A	IG-30G	IG-500A	IG-500N	IG-500E
yes	yes	yes	yes	yes	yes

SBG_SET_USER_ID (0x18)

Configure the device user ID.

Field	CMD	LEN	DATA	
Value	SBG_SET_USER_ID 0x18	5	Reserved. Leave to 0 (uint8)	User ID (uint32)
Size (bytes)	1	2	1	4

The device returns an ACK.

SBG_GET_USER_ID (0x19)

This command has no parameter and is used to retrieve the user configured ID.

Device's answer is : SBG_RET_USER_ID (0x1A)

Field	CMD	LEN	DATA
Value	SBG_RET_USER_ID 0x1A	4	User ID (uint32)
Size (bytes)	1	2	4

2.1.1.3. Restore default settings

Command availability:

IG-20	IG30A	IG-30G	IG-500A	IG-500N	IG-500E
yes	yes	yes	yes	yes	yes

SBG_RESTORE_DEFAULT_SETTINGS (0x1B)

This command is designed to reset the system in its initial configuration.

The command performs the following actions:

1. Reset all configuration settings (not calibration settings),
2. Save the settings to Flash memory,
3. Reboot the device.

Field	CMD	LEN	DATA
Value	SBG_RESTORE_DEFAULT_SETTING 0x1B	1	Reserved. Leave to 0 (uint8)
Size (bytes)	1	2	1

The device should return an ACK.

Note : *The acknowledge returned is transmitted with the current configuration. Default settings are effectively applied just after the acknowledge is sent. It belongs to user to switch his protocol settings to the default ones after reception of the ACK.*

2.1.1.4. Saving the settings

SBG_SAVE_SETTINGS (0x24)

This function saves all IG-500 parameters to non volatile memory. It has no parameter.

Device should answer with an ACK.

Note: *Calibration configuration is NOT saved with this command. The gyro and magnetometer calibration saving is made with existing dedicated functions.*

2.1.1.5. User Buffer management

These commands allows, as a complement to the User ID, having a read write access to a dedicated 64 bytes buffer located in the device's flash memory. Writing to user buffer is similar to standard configurations, and requires to be saved to flash memory if data has to be stored for future use.

Command availability:

IG-20	IG30A	IG-30G	IG-500A	IG-500N	IG-500E
yes	yes	yes	yes	yes	yes

SBG_SET_USER_BUFFER (0x1F)

Write operation in the User Buffer. User sets the first address to be written into the user buffer, and the size of the buffer.

- Max address: 0x40 (64 bytes available in the buffer)
- Max size: 64

Field	CMD	LEN	DATA			
Value	SBG_SET_USER_BUFFER 0x1F	5+ bufferSize	Reserved Set to 0x01	address (uint16)	bufferSize (uint16)	Buffer to be written
Size (bytes)	1	2	1	2	2	bufferSize

The device returns an ACK.

Possible Error Codes	Possible Reason
SBG_INVALID_PARAMETER	Check parameters : size of the buffer is not consistent
SBG_INVALID_BUFFER_OVERFLOW	Check the buffer size and address. User buffer must stay in authorized limits of 64 bytes.

SBG_GET_USER_BUFFER (0x20)

This command is used to retrieve the user buffer previously written. User sets the address in the user buffer to be read, and the size of the buffer.

- Max address: 0x40 (64 bytes available in the buffer)
- Max size: 64

Field	CMD	LEN	DATA	
Value	SBG_GET_USER_BUFFER (0x20)	4	Address (uint16)	BufferSize (uint16)
Size (bytes)	1	2	2	2

If the transaction were done without error, the device answers SBG_RET_USER_BUFFER (0x21)

Field	CMD	LEN	DATA
Value	SBG_RET_USER_BUFFER (0x21)	bufferSize	Buffer asked by user
Size (bytes)	1	2	bufferSize

In case of a problem (wrong parameter, etc), the devices answers with an ACK.

Possible Errors	Possible Reason
SBG_INVALID_PARAMETER	Check parameters : size of the buffer may be to high for the uart protocol
SBG_INVALID_BUFFER_OVERFLOW	Check the buffer size and address. User buffer must stay in bounds of memory.

2.1.1.6. Low Power Configuration

Set of commands used to turn on/off the GPS receiver, or get GPS receiver power mode. The following power modes are available (may vary depending on hardware revision):

Possible values for Imu Power are:

```
SBG_IG_MAX_PERF     (0x00)       Max performance mode. Reduces latency,
                                 allows heave computation
SBG_IG_NORMAL_MODE  (0x02)       Normal mode; Limited power consumption
                                 but we cannot get heave computation on this mode
```

Possible values for GPS Power are:

```
SBG_GPS_MAX_PERF    (0x00)       Max performance mode
SBG_GPS_ECO_MODE_1  (0x01)       Current peaks are limited. startup might be longer
                                 than in MAX_PERF mode. (Only supported with
                                 GPS board hardware V.3 and above)
SBG_GPS_ECO_MODE_2  (0x02)       Current consumption is reduced when GPS has a good fix.
                                 (Only supported with GPS board hardware V.3 and above)
SBG_GPS_OFF_MODE    (0x05)       GPS is turned off
```

Command availability:

IG-20	IG30A	IG-30G	IG-500A	IG-500N	IG-500E
yes	yes	yes	yes	yes	yes

SBG_SET_LOW_POWER_MODE (0x1C)

This command can be used for setting both IMU and GPS power modes.

The following actions are performed at the frame reception:

1. Set new power modes
2. Send an Acknowledge
3. In case of good operation:
   1. Save all settings to flash memory
   2. Reboot the device

Field	CMD	LEN	DATA		
Value	SBG_SET_LOW_POWER_MODE (0x1C)	3	Reserved. Leave to 0 (uint8)	Imu Power. (uint8)	Gps Power
Size (bytes)	1	2	1	1	1

SBG_GET_LOW_POWER_MODE (0x1D)

Command used for retrieving the GPS power state. This command has no payload. As the user send it, the device answers with SBG_RET_LOW_POWER_MODE (0x1E).

Field	CMD	LEN	DATA	
Value	SBG_RET_LOW_POWER_MODE (0x1E)	2	Imu Power. (uint8)	Gps Power
Size (bytes)	1	2	1	1

2.1.2. Output settings

2.1.2.1. Default output Mask

Commands used to configure and retrieve the default output mask. (see output masks section for more information about output masks).

Command availability:

IG-20	IG30A	IG-30G	IG-500A	IG-500N	IG-500E
yes	yes	yes	yes	yes	yes

SBG_SET_DEFAULT_OUTPUT_MASK (0x50)

This command is used to configure the default output mask. :

Field	CMD	LEN	DATA	
Value	SBG_SET_DEFAULT_OUTPUT_MASK 0x50	5	Reserved. Leave to 0 (uint8)	Default output mask (uint32)
Size (bytes)	1	2	1	4

The device returns an ACK.

SBG_GET_DEFAULT_OUTPUT_MASK (0x51)

This command retrieves the default output mask. It has no argument.

The device answers with the SBG_RET_DEFAULT_OUTPUT_MASK (0x52) command.

Field	CMD	LEN	DATA
Value	SBG_RET_DEFAULT_OUTPUT_MASK 0x52	4	Default output mask (uint32)
Size (bytes)	1	2	4

2.1.2.2. Continuous Mode

Commands used to get continuous mode information or to configure it. Automatic frames are continuously sent at the Kalman/Main loop filter frequency divided by the continuous divider. The two parameters allow to configure/get the continuous mode type:

```
Mode:
SBG_CONT_TRIGGER_MODE_DISABLE    (0x00)
SBG_CONTINUOUS_MODE_ENABLE       (0x01)
SBG_TRIGGERED_MODE_ENABLE        (0x02)

Divider:
0      : Invalid parameter
1-255  : Divider applied on main loop frequency
```

Command availability:

IG-20	IG30A	IG-30G	IG-500A	IG-500N	IG-500E
yes	yes	yes	yes	yes	yes

***Note:** All products support the basic continuous mode, but only IG-500A, IG-30G and IG-500N support the triggered output mode.*

SBG_SET_CONTINUOUS_MODE (0x53)

This command enables or disables the continuous mode.

Field	CMD	LEN	DATA		
Value	SBG_SET_CONTINUOUS_MODE 0x53	3	Reserved. Leave to 0 (uint8)	Mode (uint8)	Divider (uint8)
Size (bytes)	1	2	1	1	1

The device should return an ACK.

SBG_GET_CONTINUOUS_MODE (0x54)

This commands retrieves from the device the continuous mode configuration. It has no parameter.

The device should return SBG_RET_CONTINUOUS_MODE (0x55):

Field	CMD	LEN	DATA	
Value	SBG_RET_CONTINUOUS_MODE 0x55	2	Mode (uint8)	Divider (uint8)
Size (bytes)	1	2	1	1

2.1.2.3. Triggered Output mode

Command availability:

IG-20	IG30A	IG-30G	IG-500A	IG-500N	IG-500E
yes	yes	yes	yes	yes	yes

Commands used to configure the 4 triggered output channels.

Channels

The four channels are numbered from 0 to 3.

Trigger mask

The following triggers can generate a triggered output. The triggers can be combined with a bitwize OR:

```
SBG_TRIGGER_MAIN_LOOP_DIVIDER  0x00000001   Trigger on the main loop frequency divider
SBG_TRIGGER_MAGNETOMETERS      0x00000002   Trigger on a new magnetometers data
SBG_TRIGGER_BAROMETER          0x00000004   Trigger on a new barometer data
SBG_TRIGGER_GPS_VELOCITY       0x00000008   Trigger on a new GPS velocity
SBG_TRIGGER_GPS_POSITION       0x00000010   Trigger on a new GPS position
SBG_TRIGGER_GPS_COURSE         0x00000020   Trigger on a new GPS course
SBG_TRIGGER_TIME_PULSE         0x00000040   Trigger on an input time pulse
SBG_TRIGGER_EXT_EVENT          0x00000080   Trigger on a new external event
SBG_TRIGGER_ODO_VELOCITY_0     0x00000100   Trigger on a new odo velocity (ch0) data
SBG_TRIGGER_ODO_VELOCITY_1     0x00000200   Trigger on a new odo velocity (ch1) data
SBG_TRIGGER_EXT_TRUE_HEADING   0x00000400   Trigger on a new true heading data
SBG_TRIGGER_VIRTUAL_ODOMETER   0x00000800   Trigger when the virtual odometer reached
                                            desired distance
```

Output mask

The output mask defined in the chapter 1.6 is used to define what data the device has to send when each channel is activated by a trigger.

SBG_SET_TRIGGERED_OUTPUT (0xB9)

This command configures a trigger channel with the corresponding trigger mask and output buffer.

Field	CMD	LEN	DATA			
Value	SBG_SET_TRIGGERED_OUTPUT 0xB9	10	Reserved Leave to 0	Channel 0 – 3 (uint8)	Trigger Mask (uint32)	Output mask (uint32)
Size (bytes)	1	2	1	1	4	4

The device should return an ACK.

Possible Error Code	Possible Reason
`SBG_INVALID_PARAMETER`	Check trigger channel

Note: *It is possible to disable a trigger channel by setting its trigger mask to 0x00000000.*

SBG_GET_TRIGGERED_OUTPUT (0xBA)

This command is used to retrieve a trigger channel configuration.

Field	CMD	LEN	DATA
Value	SBG_GET_TRIGGERED_OUTPUT (0xBA)	1	Channel (uint8)
Size (bytes)	1	2	1

If the transaction were done without error, the device answers with SBG_RET_TRIGGERED_OUTPUT (0xBB)

Field	CMD	LEN	DATA	
Value	SBG_RET_TRIGGERED_OUTPUT (0xBB)	8	Trigger Mask (uint32)	Output Mask (uint32)
Size (bytes)	1	2	4	4

In case of a problem (wrong parameter, etc), the devices answers with an ACK.

Possible Errors	Possible Reason
`SBG_INVALID_PARAMETER`	Check parameters : check trigger channel

2.1.2.4. ASCII / NMEA outputs configuration

NMEA and ASCII outputs can be configured by a single command.

Command availability:

IG-20	IG30A	IG-30G	IG-500A	IG-500N	IG-500E
yes	yes	yes	yes	yes	yes

The following frames can be configured through these commands. Please check section 4 for more details about these sentences.

SBG_OUTPUT_NMEA_GPGGA	0x00
SBG_OUTPUT_NMEA_GPRMC	0x01
SBG_OUTPUT_NMEA_GPZDA	0x02
SBG_OUTPUT_ASCII_SBG01	0x03
SBG_OUTPUT_NMEA_HEHDT	0x04
SBG_OUTPUT_ASCII_KVHEXT	0x05
SBG_OUTPUT_NMEA_HEHDM	0x06
SBG_OUTPUT_NMEA_PSXN23	0x07

Each frame can be configured with a divider, set to reduce main loop frequency. Setting the divider to 0 is ignored and replaced by 1.

A trigger mask is the same as for triggered output mode and allows generating ASCII outputs upon special event reception.

A trigger mask set to 0 disables the corresponding frame.

SBG_SET_ASCII_OUTPUT_CONF (0xDE)

This command is used to enable/disable an ASCII output.

Field	CMD	LEN	DATA		
Value	SBG_SET_ASCII_OUTPUT_CONF 0xDE	6	Frame Id (uint8)	Divider (uint8)	Trigger mask (uint32)
Size (bytes)	1	2	1	1	4

Device should answer an ACK.

SBG_GET_ASCII_OUTPUT_CONF (0xDF)

This command is used to get the required frame configuration:

Field	CMD	LEN	DATA
Value	SBG_GET_ASCII_OUTPUT_CONF 0xDF	1	Frame Id (uint8)
Size (bytes)	1	2	1

Device should answer with the following frame:

Field	CMD	LEN	DATA		
Value	SBG_RET_ASCII_OUTPUT_CONF 0xE0	6	Frame Id (uint8)	Divider (uint8)	Trigger mask (uint32)

Size (bytes)	1	2	1	1	4

2.1.3. Protocol settings

2.1.3.1. Protocol mode

Commands used to set/get the baud rate of the device. Command availability:

IG-20	IG30A	IG-30G	IG-500A	IG-500N	IG-500E
yes	yes	yes	yes	yes	yes

The Uart Mode field is a 32 bit number composed of the following data

- Bit 31 (MSB): 0 = Normal (fast) mode
1 = EMI reduction active → Slew rate is reduced on Tx line, so baudrate is limited to 230400bps. If changed, please restart the device.
This parameter is only valid on new devices.

- Bit (0 – 30): Baudrate configured. Possible values are
[9600; 19200; 36400; 57600; 115200; 230400; 460800; 921600]

SBG_SET_PROTOCOL_MODE (0x12)

This commands allows to change the transmission speed on the serial line.

Field	**CMD**	**LEN**	**DATA**	
Value	SBG_SET_PROTOCOL_MODE 0x12	5	Reserved Leave to 0 (uint8)	Uart mode (uint32)
Size (bytes)	1	2	1	4

The device answers with a SBG_RET_PROTOCOL_MODE (0x14) frame.

Field	**CMD**	**LEN**	**DATA**
Value	SBG_RET_PROTOCOL_MODE 0x14	4	Uart mode (uint32)
Size (bytes)	1	2	4

In some cases, an error code can be returned :

Error	**Possible Reason**
`SBG_INVALID_PARAMETER`	1. User did not ask for a valid communication speed → *Choose a valid communication speed, with consistent EMI reduction parameter with actual baudrate*

Note : *The answer is transmitted in the current speed. The new setting is effectively applied just after the response of the device.*

SBG_GET_PROTOCOL_MODE (0x13)

This command has no argument. It is used to retrieve the configured baud rate used by the device.

The answer is SBG_RET_PROTOCOL_MODE (0x14). See previous paragraph to see its definition.

2.1.3.2. Output mode

These commands allows configure or check how data are transmitted: Endianess is configured, as well as the real numbers format (standard float/double, or fixed32/fixed64 point format).

Different values for output mode that can be ORed are:

```
SBG_OUTPUT_MODE_DEFAULT             0x00
SBG_OUTPUT_MODE_BIG_ENDIAN          0x00  (default)
SBG_OUTPUT_MODE_LITTLE_ENDIAN       0x01
SBG_OUTPUT_MODE_FLOAT               0x00  (default)
SBG_OUTPUT_MODE_FIXED               0x02
```

By default, the output mode is in Big endian, and Float/Double format. By specifying `SBG_OUTPUT_MODE_LITTLE_ENDIAN | SBG_OUTPUT_MODE_FIXED`, you change the output format to little endian and fixed32/fixed64 format.

***Note:** By nature, this configuration only affects the IG-Devices binary protocol.*

Command availability:

IG-20	IG30A	IG-30G	IG-500A	IG-500N	IG-500E
yes	yes	yes	yes	yes	yes

SBG_SET_OUTPUT_MODE (0x15)

Field	CMD	LEN	DATA	
Value	SBG_SET_OUTPUT_MODE 0x15	2	Reserved Leave to 0 (uint8)	Output mode (uint8)
Size (bytes)	1	2	1	1

SBG_GET_OUTPUT_MODE (0x16)

This command retrieves the output mode of the device. It has no parameter.

The device's answer is the SBG_RET_OUTPUT_MODE frame (0x17).

Field	CMD	LEN	DATA
Value	SBG_RET_OUTPUT_MODE 0x17	1	Output mode (uint8)
Size (bytes)	1	2	1

2.1.4. Kalman filter and sensors settings

2.1.4.1. Motion profile setting

Motion profile setting is allowed in a configuration level, but should not be performed in live, because updating a motion profile requires a device reset.

Command availability:

IG-20	IG30A	IG-30G	IG-500A	IG-500N	IG-500E
no	no	no	yes	yes	yes

SBG_SEND_MP_BUFFER (0xD1)

This sends a part of the motion profile buffer into the IG-Device:

The index where to place the data in the internal motion profile is specified. The buffer size is defined directly by the frame size.

Field	CMD	LEN	DATA	
Value	SBG_SEND_MP_ BUFFER (0xD1)	2 + X	index	Raw motion profile buffer
Size (bytes)	1	2	2	X

The sensor should answer with an ACK

SBG_VALIDATE_MP_BUFFER (0xD2)

This command has no parameter and is used to validate the buffer once it is fully sent.

The sensor should answer with an ACK.

The following actions are performed at reception of this frame:

1. Check that the buffer is consistent and compatible with current hardware / software
2. Send an Acknowledge with corresponding result
3. In case of good configuration
   1. Save all settings to flash memory
   2. Reboot the device

> ***Note:*** *As a reboot is initiated if the buffer Is valid, a long latency should be expected (< 2s) after the ACK is received.*

SBG_GET_MP_INFO (0xD3)

This command has no parameter and is used to retrieve the current motion profile info: Unique identifier, as well as version number.

The device should answer with a SBG_RET_MP_INFO (0xD4) frame:

Field	CMD	LEN	DATA	
Value	SBG_RET_MP_INFO (0xD4)	8	Motion profile ID (uint32)	Motion profile revision (uint32)
Size (bytes)	1	2	4	4

2.1.4.2. Heading Source Configuration

Different sources of heading measurement are available.

The different choices are :

```
SBG_HEADING_SOURCE_NONE                  0x00   (Not available for IG-500N / IG-500E)
SBG_HEADING_SOURCE_MAGNETOMETERS         0x01   (Default)
SBG_HEADING_SOURCE_GPS_COURSE            0x02   (Not available for IG-500A)
SBG_HEADING_GPS_ACCELERATIONS            0x03   (IG-500N and IG-500E)
SBG_HEADING_SOURCE_USER                  0x05
SBG_HEADING_SOURCE_REMOTE_MAG            0x06   (IG-500E only)
SBG_HEADING_SOURCE_REMOTE_TRUE_HEADING   0x07   (IG-500E only)
```

Command availability:

IG-20	IG30A	IG-30G	IG-500A	IG-500N	IG-500E
no	yes	yes	yes	yes	yes

SBG_SET_FILTER_HEADING_SOURCE (0x3D)

This command is used to configure the source of heading estimate.

Field	CMD	LEN	DATA	
Value	SBG_SET_FILTER_HEADING_SOURCE 0x3D	2	Reserved Leave to 0 (uint8)	Source (uint8)
Size (bytes)	1	2	1	1

Possible Error Codes	Possible Reason
SBG_INVALID_PARAMETER	User sent a source type that is not supported by the device

SBG_GET_FILTER_HEADING_SOURCE (0x3E)

This command accept no argument and is used to retrieve the heading source used by the Kalman filter.

The device may answer SBG_RET_FILTER_HEADING_SOURCE (0x3F)

Field	CMD	LEN	DATA
Value	SBG_RET_ FILTER_HEADING_SOURCE 0x3F	1	Source (uint8)
Size (bytes)	1	2	1

2.1.4.3. Magnetic Declination

This command is used to define/get the north magnetic declination. For IG-500N devices, when magnetometers are used for heading reference, this information is very important to ensure good results as navigation estimation is concerned.

The magnetic declination is expressed in radians. The valid range is between $[-\pi, +\pi]$

Command availability:

IG-20	IG30A	IG-30G	IG-500A	IG-500N	IG-500E
no	yes	yes	yes	yes	yes

SBG_SET_MAGNETIC_DECLINATION (0x40)

Configure magnetic declination.

Field	CMD	LEN	DATA	
Value	SBG_SET_MAGNETIC_DECLINATION 0x40	5	Reserved Leave to 0 (uint8)	declination (real32)
Size (bytes)	1	2	1	4

Possible Error Codes	Possible Reason
`SBG_INVALID_PARAMETER`	User sent an invalid magnetic declination.

SBG_GET_MAGNETIC_DECLINATION (0x41)

This command is used to retrieve the magnetic declination used by our Kalman filter. The returned value is expressed in radians.

The device may answer SBG_RET_MAGNETIC_DECLINATION (0x42)

Field	CMD	LEN	DATA
Value	SBG_RET_ MAGNETIC_DECLINATION 0x42	4	declination (real32)
Size (bytes)	1	2	4

2.1.4.4. Heave configuration

These functions are used to configure the heave operation.

Command availability:

IG-20	IG30A	IG-30G	IG-500A	IG-500N	IG-500E
no	no	no	yes	yes	yes

SBG_SET_HEAVE_CONF (0xD8)

Used to enable or disable heave computation. Heave period should be set between 0.3 and 20.0.

If enabled, the output latency should be slightly increased due to added computations.

Field	CMD	LEN	DATA		
Value	SBG_SET_HEAVE_CONF (0xD8)	8	Enable Heave (bool)	Heave Period (s) (real32)	Reserved – padding; Leave to 0
Size (bytes)	1	2	1	4	3

Device should answer with an ACK.

SBG_GET_HEAVE_CONF (0xD9)

Used to get the heave configuration. Send the frame with an empty payload to retrieve setting.

Device should answer with the following frame:

Field	CMD	LEN	DATA		
Value	SBG_RET_HEAVE_CONF (0xDA)	8	Enable Heave (bool)	Heave Period (s) (real32)	Reserved – padding; Leave to 0
Size (bytes)	1	2	1	4	3

Device should answer with an ACK.

2.1.5. GPS , barometric sensor, and Navigation settings

2.1.5.1. Reference Pressure for Barometric Altitude

This command set/get the reference pressure used for barometric altitude calculations. The pressure is expressed in *Pa.*

Factory Default is set to 101 325 Pa.

Command availability:

IG-20	IG30A	IG-30G	IG-500A	IG-500N	IG-500E
no	no	yes	no	yes	yes

SBG_SET_REFERENCE_PRESSURE (0xA0)

Configure reference pressure. If 0 is passed as an argument, the current pressure is directly considered as the reference pressure.

Field	CMD	LEN	DATA	
Value	SBG_SET_REFERENCE_PRESSURE 0xA0	5	Reserved Leave to 0 (uint8)	refPressure (uint32)
Size (bytes)	1	2	1	4

Device's answer is an ACK.

SBG_GET_REFERENCE_PRESSURE (0xA1)

This command has no argument and is used to retrieve the reference pressure used for altitude calculation.

The device should answer SBG_RET_REFERENCE_PRESSURE (0xA2)

Field	CMD	LEN	DATA
Value	SBG_RET_REFERENCE_PRESSURE 0xA2	4	refPressure (uint32)
Size (bytes)	1	2	4

2.1.5.2. Navigation Velocity Source

This command configures or retries the input source for velocity aiding information used by the Kalman Filter to correct for velocity drift.

If your application couldn't use a GPS receiver and you have an other source of velocity information, you can feed the device with an external velocity using the command SBG_SEND_NAV_VELOCITY.

Possible values for source are:

```
SBG_VEL_SRC_GPS                                      0x00   (Default)
SBG_VEL_SRC_USER                                     0x02
SBG_VEL_SRC_ODO                                      0x03
```

Command availability:

IG-20	IG30A	IG-30G	IG-500A	IG-500N	IG-500E
no	no	no	no	yes	yes

SBG_SET_NAV_VELOCITY_SRC (0xA8)

Configure the Velocity source or measurement.

Field	CMD	LEN	DATA	
Value	SBG_SET_NAV_VELOCITY_SRC 0xA8	2	Reserved Leave to 0 (uint8)	source (uint8)
Size (bytes)	1	2	1	1

Device's answer is an ACK.

SBG_GET_NAV_VELOCITY_SRC (0xA9)

This command has no argument and is used to retrieve the input source for velocity aiding information.

The device answer `SBG_RET_NAV_VELOCITY_SRC (0xAA)`

Field	CMD	LEN	DATA
Value	SBG_RET_NAV_VELOCITY_SRC 0xAA	1	source (uint8)
Size (bytes)	1	2	1

2.1.5.3. Navigation Position Source

These commands configures/retrieve the input source for position aiding information used by the Kalman Filter to correct for position drift. The IG-500N includes a GPS receiver and a barometric sensor. With the default configuration SBG_SRC_GPS_AND_BARO, the Kalman Filter uses these two source of information to correct the position drift.

For some applications, it could be useful to disable the barometric sensor. For example, if the device is placed in a pressurized aircraft, the barometric information will be unusable and should be disabled.

If your application couldn't use a GPS receiver and you have an other source of position information, you can feed the device with an external position using the command SBG_SEND_NAV_POSITION.

Possible values for source are:

```
SBG_POS_SRC_GPS                         0x00   (Default on IG-500E)
SBG_POS_SRC_GPS_AND_BARO                0x01   (Default on IG-500N)
SBG_POS_SRC_USER                        0x02
```

Command availability:

IG-20	IG30A	IG-30G	IG-500A	IG-500N	IG-500E
no	no	no	no	yes	yes

SBG_SET_NAV_POSITION_SRC (0xAB)

Configure the position source.

Field	CMD	LEN	DATA	
Value	SBG_SET_NAV_POSITION_SRC 0xAB	2	Reserved Leave to 0 (uint8)	source (uint8)
Size (bytes)	1	2	1	1

Device's answer is an ACK.

SBG_GET_NAV_POSITION_SRC (0xAC)

This command has no argument and is used to retrieve the input source for position aiding information.

The device answer `SBG_RET_NAV_POSITION_SRC (0xAD)`

Field	CMD	LEN	DATA
Value	SBG_RET_NAV_POSITION_SRC 0xAD	1	source (uint8)
Size (bytes)	1	2	1

2.1.5.4. GPS Antenna Lever Arm

Use this command to specify the vector from the device to the GPS antenna. The lever arm is expressed in meters in the device coordinate system.

Command availability:

IG-20	IG30A	IG-30G	IG-500A	IG-500N	IG-500E
no	no	no	no	yes	yes

SBG_SET_GPS_LEVER_ARM (0xAE)

This command configures the GPS lever arm.

Field	CMD	LEN	DATA			
Value	SBG_SET_GPS_LEVER_ARM 0xAE	13	Reserved Leave to 0	armX (real32)	armY (real32)	armZ (real32)
Size (bytes)	1	2	1	4	4	4

Device's answer is an ACK.

SBG_GET_GPS_LEVER_ARM (0xAF)

This command has no argument and is used to retrieve the GPS lever arm. See the command SBG_SET_GPS_LEVER_ARM for arguments description.

The device answer `SBG_RET_GPS_LEVER_ARM (0xB0)`

Field	CMD	LEN	DATA		
Value	SBG_RET_GPS_LEVER_ARM 0xB0	12	armX (real32)	armY (real32)	armZ (real32)
Size (bytes)	1	2	4	4	4

2.1.5.5. Gravity Magnitude configuration

These commands get / set the local gravity magnitude. An invalid gravity magnitude can lead to significant errors in velocity and position calculation for IG-500N devices.

The gravity magnitude field is expressed in $m \,.\, s^{-2}$ and the default value is 9.809 $m \,.\, s^{-2}$.

Command availability:

IG-20	IG30A	IG-30G	IG-500A	IG-500N	IG-500E
no	no	no	yes	yes	yes

SBG_SET_GRAVITY_MAGNITUDE (0xB1)

Configure the gravity magnitude.

Field	CMD	LEN	DATA	
Value	SBG_SET_GRAVITY_MAGNITUDE 0xB1	5	Reserved Leave to 0 (uint8)	magnitude (real32)
Size (bytes)	1	2	1	4

Device's answer is an ACK.

SBG_GET_GRAVITY_MAGNITUDE (0xB2)

This command has no argument and is used to retrieve the local gravity magnitude. See the command SBG_GET_GRAVITY_MAGNITUDE for arguments description.

The device answer `SBG_RET_GRAVITY_MAGNITUDE (0xB3)`

Field	CMD	LEN	DATA
Value	SBG_RET_GRAVITY_MAGNITUDE 0xB3	4	magnitude (real32)
Size (bytes)	1	2	4

2.1.6. Coordinate frame transformation settings

The following commands are used to make sensor transformations, automatically or manually, directly on sensors outputs, or on the orientation output.

> ***Note:*** *Setting a Pre alignment (either automatic or manual) causes the sensor to perform the following actions:*
>
> *1) Apply the alignment*
>
> *2) Save all settings to flash memory*
>
> *3) Reboot the sensor*
>
> *This is why a large latency should be expected after these functions calls (< 2s)*

Commands availability:

IG-20	IG30A	IG-30G	IG-500A	IG-500N	IG-500E
yes	yes	yes	yes	yes	yes

SBG_SET_AUTO_ORIENTATION_OFFSET (0x39)

This command automatically align the device to the coordinate frame's axis (*Z*, *X* and *Y*, or *X Y* and *Z*).

Two types of alignment can be done, a pre-alignment, the whole device frame including sensors outputs are aligned with the new coordinate system or a post-alignement, only the calculated orientation will be rotated.

For IG-500N devices, only the pre-alignement options are valid. Indeed, the IG-500N need a consistent information between calculated outputs and sensors measurements.

Different alignment type are :

```
SBG_OFFSET_PRE_ROT_Z_RESET        (0x03)
SBG_OFFSET_PRE_ROT_XY_RESET       (0x04)
SBG_OFFSET_PRE_ROT_XYZ_RESET      (0x05)
SBG_OFFSET_POST_ROT_Z_RESET       (0x06)
SBG_OFFSET_POST_ROT_XY_RESET      (0x07)
SBG_OFFSET_POST_ROT_XYZ_RESET     (0x08)
```

> ***Note :*** *See IG-500 user manual to get a detailed explanation on available alignment type.*

Field	CMD	LEN	DATA	
Value	SBG_SET_AUTO_ORIENTATION_OFFSET 0x39	2	Reserved Leave to 0	Offset type (uint8)
Size (bytes)	1	2	1	1

Device's answer is an ACK.

Possible Error Codes	Possible Reason
`SBG_INVALID_PARAMETER`	User did not chose one of the listed parameters above.

SBG_SET_MANUAL_ORIENTATION_OFFSET (0x3A)

This command configures a manual rotation that has to be applied on sensors input (pre-rotation), or in orientation output (post-rotation).

As explained above, you can't specify an post rotation on IG-500N devices.

Different choices for offset type are :

```
SBG_OFFSET_PRE_ROT  (0x01)
SBG_OFFSET_POST_ROT (0x02)
```

***Note :** See IG-500 user manual to get the matrix organization in memory*

Field	CMD	LEN	DATA		
Value	SBG_SET_MANUAL_ORIENTATION_OFFSET 0x3A	38	Reserved Leave to 0	Offset type (uint8)	Rot. matrix (9 * real32)
Size (bytes)	1	2	1	1	36

Device's answer is an ACK. Possible error codes are :

Error	Possible Reason
`SBG_INVALID_PARAMETER`	1. User did not chose one of the listed parameters above. → *Chose a valid rotation type* 2. User did not input a real rotation matrix → *Check the rotation matrix : The three vectors must be normalized, and they must be orthogonal to each other.*

SBG_GET_ORIENTATION_OFFSET (0x3B)

Retrieves the current orientation offset in matrix form. The only one parameter is the type of offset we are asking for.
Possible value are:

```
SBG_OFFSET_PRE_ROT  (0x01)
SBG_OFFSET_POST_ROT (0x02)
```

Field	CMD	LEN	DATA
Value	SBG_GET_ORIENTATION_OFFSET 0x3B	1	Offset type (uint8)
Size (bytes)	1	2	1

The device may answer the SBG_RET_ORIENTATION_OFFSET (0x3C) frame.

Field	CMD	LEN	DATA
Value	SBG_RET_ORIENTATION_OFFSET 0x3C	36	Rotation matrix (9 * real32)
Size (bytes)	1	2	36

2.1.7. Devices synchronization

These commands are used to configure the new IG-500 hardware synchronization features.

> ***Note:*** *These commands are available on GPS / Connector board version 3.0.0.0 and above. The 4 pins box version do not include these pins in the output interface, but the signals are still present internally.*

Command availability:

IG-20	IG30A	IG-30G	IG-500A	IG-500N	IG-500E
yes	yes	yes	yes	yes	yes

2.1.7.1. Synchronization inputs

Depending on the device, up to two synchronization inputs can be configured independently.

The following configurations can be made:

1. Channel: the input channel, from 0 to 1.
2. Input type, which defines the event type

```
SBG_IN_DISABLED           0x00          Input channel disabled
SBG_IN_EVENT              0x01          General purpose event trigger
SBG_IN_TIME_PULSE         0x03          GPS PPS time input
SBG_IN_ODOMETER           0x05          Odometer signal input
SBG_IN_ODOMETER_DIRECTION 0x06          Odometer direction sensing
```

3. Input sensitivity:

```
SBG_IN_FALLING_EDGE       0x00          The trigger will be activated by a falling edge
SBG_IN_RISING_EDGE        0x01          The trigger will be activated by a rising edge
SBG_IN_LEVEL_CHANGE       0x02          The trigger is activated by a level change
```

4. Input nanosecond delay before event actually occurs.

> ***Note:*** *Input sensitivity and delay are not used when the logic input is configured as odometer direction. In this case, only the input pin level is as general purpose input when a new odometer pulse is detected.*

SBG_SET_SYNC_IN_CONF (0xC1)

This command is used to configure the IG-Device synchronization Inputs.

Field	CMD	LEN	DATA				
Value	SBG_SET_SYNC_IN_CONF 0xC1	8	Res. Set to 0	Channel (uint8)	InputType (uint8)	Sensitivity (uint8)	Nanoseconds Delay (uint32)
Size (bytes)	1	2	1	1	1	1	4

Device's answer is an ACK.

Possible Error Codes	Possible Reason
`SBG_INVALID_PARAMETER`	User did not chose one of the listed parameters above.

SBG_GET_SYNC_IN_CONF (0xC2)

This command has only one parameter and is used to retrieve the device synchronization input configuration.

Field	CMD	LEN	DATA
Value	SBG_GET_SYNC_IN_CONF 0xC2	8	Channel (uint8)
Size (bytes)	1	2	1

The device should answer with the following frame: SBG_RET_SYNC_IN_CONF (0xC3)

Field	CMD	LEN	DATA			
Value	SBG_RET_SYNC_IN_CONF 0xC3	8	Channel (uint8)	InputType (uint8)	Sensitivity (uint8)	Nanoseconds Delay (uint32)
Size (bytes)	1	2	1	1	1	4

2.1.7.2. Synchronization Output

Depending on the device, a synchronization output pin might be available.

The following configuration may be done on this output pin:

1. Channel: For now, it is always 0.
2. Output type, which defines the event that generates the output:

SBG_OUT_DISABLED	0x00	Input channel disabled
SBG_OUT_MAIN_LOOP_START	0x01	Main loop starting trigger
SBG_OUT_MAIN_LOOP_DIVIDER	0x02	Trigger activated at the beginning of each main loop where a continuous output is generated
SBG_OUT_TIME_PULSE_COPY	0x03	Copy of the GPS time pulse input trigger
SBG_OUT_VIRTUAL_ODO	0x05	Virtual odometer logic output: Enabled each x meters of travel

3. Output sensitivity:

SBG_OUT_FALLING_EDGE	0x00,	The output pin will generate a falling edge
SBG_OUT_RISING_EDGE	0x01,	The output pin will generate a rising edge
SBG_OUT_TOGGLE	0x02	The pulse is a level change

4. Milliseconds pulse duration

SBG_SET_SYNC_OUT_CONF (0xC4)

This command is used to configure the IG-Device synchronization Output pin.

Field	CMD	LEN	DATA				
Value	SBG_SET_SYNC_OUT_CONF 0xC4	5	Res. Set to 0	Channel (uint8)	OutputType (uint8)	polarity (uint8)	Duration (ms) (uint8)
Size (bytes)	1	2	1	1	1	1	1

Device's answer is an ACK.

Possible Error Codes	Possible Reason
SBG_INVALID_PARAMETER	User did not chose one of the listed parameters above.

SBG_GET_SYNC_OUT_CONF (0xC5)

This command has only one parameter and is used to retrieve the device synchronization input configuration.

Field	CMD	LEN	DATA
Value	SBG_GET_SYNC_OUT_CONF 0xC5	8	Channel (uint8)
Size (bytes)	1	2	1

The device should answer with the following frame: SBG_RET_SYNC_OUT_CONF (0xC6)

Field	CMD	LEN	DATA			
Value	SBG_RET_SYNC_OUT_CONF 0xC6	5	Channel (uint8)	OutputType (uint8)	polarity (uint8)	Duration (ms) (uint8)
Size (bytes)	1	2	1	1	1	1

2.1.7.3. Virtual odometer configuration (associated with Sync Out)

SBG_SET_VIRTUAL_ODO_CONF (0xDB)

This function is used to configure the distance between two pulses when the sync out signal is used as a virtual odometer

Field	CMD	LEN	DATA	
Value	SBG_SET_VIRTUAL_ODO_CONF 0xDB	8	distance (real32)	Reserved; Leave to 0 (uint32)
Size (bytes)	1	2	4	4

Device's answer should be an ACK.

SBG_GET_VIRTUAL_ODO_CONF (0xDC)

This command has no parameter, and is used to retrieve the virtual odometer configuration. Device answer should be SBG_RET_VIRTUAL_ODO_CONF (0xDD) :

Field	CMD	LEN	DATA	
Value	SBG_RET_VIRTUAL_ODO_CONF 0xDD	8	distance (real32)	Reserved; Left to 0 (uint32)
Size (bytes)	1	2	4	4

2.1.8. Magnetometers and gyroscopes calibration

Gyroscopes and Magnetometers calibration commands are not linked directly to other settings. By this way a dedicated non volatile memory saving procedure is provided. This allows resetting default settings while keeping a particular magnetic calibration for example.

2.1.8.1. Magnetometers calibration general operations

This command concerns the calibration default setting reset and calibration saving to flash memory.

Commands availability:

IG-20	IG30A	IG-30G	IG-500A	IG-500N	IG-500E
no	yes	yes	yes	yes	yes

Possible values for argument are :

```
SBG_CALIB_LOAD_DEFAULT     0x00
SBG_CALIB_SAVE             0x05
```

SBG_CALIB_MAG (0x70)

Start, Stop, or save calibration data of the magnetometers.

Field	CMD	LEN	DATA
Value	SBG_CALIB_MAG 0x70	1	Argument (uint8)
Size (bytes)	1	2	1

Device's answer is an Ack.

2.1.8.2. Setting magnetic calibration parameters

The manual calibration procedure allows to manually send Hard/Soft Iron compensation parameters directly to the device. It may be useful if an external calibration procedure is used.

The calibrated magnetic field data is calculated with the following formula:

$$M_{Cal}=\begin{pmatrix} AGm_0 & AGm_3 & AGm_6 \\ AGm_1 & AGm_4 & AGm_7 \\ AGm_2 & AGm_5 & AGm_8 \end{pmatrix}\times\left[\begin{pmatrix} M_X \\ M_Y \\ M_Z \end{pmatrix}-\begin{pmatrix} Bias_X \\ Bias_Y \\ Bias_Z \end{pmatrix}\right]$$

With the Bias vector Bias and the Alignment and Gain Matrix AGm.

SBG_CALIB_MAG_SET_MANUAL (0x72)

This function allow to send Hard and Soft Iron transformations to the device.

Field	CMD	LEN	DATA	
Value	SBG_CALIB_MAG_SET_MANUAL (0x72)	48	Magnetic Bias Vector 3 * real32	Magnetic Alignment Matrix 9 * real32
Size (bytes)	1	2	12	36

Device's answer is an Ack. Possible error codes are :

Error	Possible Reason
`SBG_INVALID_PARAMETER`	User did not set a valid matrix or vector

SBG_CALIB_MAG_GET_TRANSFORMATIONS (0x73)

This command has no parameter and is used to retrieve the calibration parameters of the connected device.

The device should answer with the frame SBG_CALIB_MAG_RET_TRANSFORMATIONS (0x74):

Field	CMD	LEN	DATA	
Value	SBG_CALIB_MAG_RET_TRANSFORMATIONS (0x74)	48	Magnetic Bias Vector 3 * real32	Magnetic Alignment Matrix 9 * real32
Size (bytes)	1	2	12	36

2.1.8.3. Gyroscopes calibration

This command allows to save the current gyro value as a gyro bias. Three options are let for gyro calibration: Coarse, Medium and fine which will respectively measure gyro bias for 0.25s, 1s, and 3s. Best accuracy is achieved with the Fine method. Argument passed has five possible values :

```
SBG_CALIB_LOAD_DEFAULT    0x00
SBG_CALIB_MEASURE_COARSE  0x04
SBG_CALIB_MEASURE_MEDIUM  0x06
SBG_CALIB_MEASURE_FINE    0x07
SBG_CALIB_SAVE            0x05
```

Command availability:

IG-20	IG30A	IG-30G	IG-500A	IG-500N	IG-500E
no	no	no	yes	yes	yes

SBG_CALIB_GYRO_BIAS (0x71)

Calibrate the Gyroscopes bias.

Field	CMD	LEN	DATA
Value	SBG_CALIB_GYRO_BIAS 0x71	1	Argument (uint8)
Size (bytes)	1	2	1

Device's answer is an Ack. Possible error codes are :

Error	Possible Reason
`SBG_INVALID_PARAMETER`	User did not set a proper value for “argument”

2.1.9. External devices configuration (IG-500E)

2.1.9.1. External serial communication device

The IG-500E allows multiple external devices to be connected on it. A configuration frame is provided to select the external device type and associated common configurations, such as baudrate. In addition, a generic mechanism allows to send configuration that are interpreted by the external device module. This helps extending external devices capabilities with specific functions.

The following external devices types are supported:

```
SBG_EXT_NONE              0x00   No external device is attached
SBG_EXT_IG_DEVICE         0x03   SBG Systems IG device
SBG_EXT_NMEA              0x04   Other device which uses standard NMEA protocol
```

The baudrate can be one of the following:

[9600; 19200; 38400; 57600; 115200; 230400 460800; 921600]

In addition, some options are available to configure the UART interface (RS-232 / RS-422, or fast / slow mode). These options can be bitwize ORed.

```
SBG_EXT_PORT_RS232             0x0000   Set the external port in RS-232 mode (default)
SBG_EXT_PORT_RS422             0x0001   Set the external port in RS-422 mode
SBG_EXT_PORT_DIS_EMI_REDUCTION 0x0000   Fast mode of operation; baudrate is not limited
SBG_EXT_PORT_EN_EMI_REDUCTION  0x0002   Slow slew rate mode for EMI reduction.
                                        Baudrate is then limited to 230400bps
```

> ***Note:*** *A device reboot (actual power cycle) is necessary to take the UART port options into account.*

Commands availability:

IG-20	IG30A	IG-30G	IG-500A	IG-500N	IG-500E
no	no	no	no	no	yes

SBG_SET_EXTERNAL_DEVICE (0xBC)

This command set the external device configuration.

Field	CMD	LEN	DATA			
Value	SBG_SET_EXTERNAL_DEVICE 0xBC	8	Reserved Leave to 0	DeviceType (uint8)	Baudrate (uint32)	Uart Options
Size (bytes)	1	2	1	1	4	2

Device's answer is an Ack. Possible error codes are :

Error	Possible Reason
`SBG_INVALID_PARAMETER`	User did not set a proper value for “argument”

The IG-500E should return an acknowledge.

SBG_GET_EXTERNAL_DEVICE (0xBD)

This command has no parameter. The answer should be SBG_RET_EXTERNAL_DEVICE (0xBE):

Field	CMD	LEN	DATA		
Value	SBG_SET_EXTERNAL_DEVICE 0xBC	8	DeviceType (uint8)	Baudrate (uint32)	Uart Options (uint16)
Size (bytes)	1	2	1	4	2

SBG_SET_EXTERNAL_DEVICE_CONF (0xBF)

This command is used to send an external device specific configuration.

This command is interpreted as specific configuration, that is encapsulated into this frame. The protocol system will then transfer all the payload to the external device sub-system for reinterpretation.

> ***Note:*** *Check external devices integration manuals in order to see what payload to put depending on the external device.*

> ***Note 2:*** *Each payload is specific to the external device.*

Field	CMD	LEN	DATA
Value	SBG_SET_EXTERNAL_DEVICE_CONF 0xBF	xx	Specific configuration command with payload
Size (bytes)	1	2	Variable (>0)

Once this frame payload is interpreted by the external device module, it should send back to the main protocol and answer, that is encapsulated into the following frame: SBG_RET_EXTERNAL_DEVICE_CONF (0xC0).

Field	CMD	LEN	DATA
Value	SBG_RET_EXTERNAL_DEVICE_CONF 0xC0	xx	Specific configuration answer with payload
Size (bytes)	1	2	Variable (>0)

2.1.9.2. Odometer management

Three sets of commands are provided to configure the odometer management by the IG-500E.

- Odometer channel: may be 0 or 1.
- Odometer sensitive axis:

```
SBG_ODO_X                 0x00         Odometer sensitive axis is X
SBG_ODO_Y                 0x01         Odometer sensitive axis is Y
SBG_ODO_Z                 0x02         Odometer sensitive axis is Z
```

- Odometer direction:

```
SBG_ODO_DIR_POSITIVE      0x00         Odometer velocity is always positive
SBG_ODO_DIR_NEGATIVE      0x01         Odometer velocity is always negative
```

- Odometer pulses per meter. This parameter corresponds to the odometer gain.
- Odometer Lever Arm with respect to the sensor.
- Gain Options, as defined below:

Bit 7	Bit [6-0]
1 → Enable automatic gain correction by GPS 0 → Disable automatic gain correction by GPS	Gain error in %

Note: *In order to stop or inhibit the odometer operation, user should do the following operations:*

1) Disable the odometer from the Kalman filter velocity source

2) In case of long term odometer turn off, disable the Odometer logic input pin.

Note 2: *GPS automatic gain compensation is only available on a single odometer axis.*

Commands availability:

IG-20	IG30A	IG-30G	IG-500A	IG-500N	IG-500E
no	no	no	no	no	yes

SBG_SET_ODO_CONFIG (0xC7)

Configures the odometer general settings:

Field	CMD	LEN	DATA				
Value	SBG_SET_ODO_CONFIG 0xC7	8	Reserved Leave to 0	Channel (uint8)	Axis (uint8)	PulsesPerMeter (real32)	GainOpt (uint8)
Size (bytes)	1	2	1	1	1	4	1

Device's answer should be an Ack..

SBG_GET_ODO_CONFIG (0xC8)

This command retrieves the odometer channel configuration:

Field	CMD	LEN	DATA
Value	SBG_GET_ODO_CONFIG 0xC8	1	Channel (uint8)
Size (bytes)	1	2	1

The device's answer should be the following: SBG_RET_ODO_CONFIG (0xC9)

Field	CMD	LEN	DATA			
Value	SBG_RET_ODO_CONFIG 0xC9	7	Channel (uint8)	Axis (uint8)	PulsesPerMeter (real32)	GainOpt (uint8)
Size (bytes)	1	2	1	1	4	1

SBG_SET_ODO_DIRECTION (0xCA)

Configures the channel's sensing direction.

Field	CMD	LEN	DATA		
Value	SBG_SET_ODO_DIRECTION 0xCA	3	Reserved Leave to 0	Channel (uint8)	OdoDirection (uint8)
Size (bytes)	1	2	1	1	1

Device's answer should be an Ack..

SBG_GET_ODO_DIRECTION (0xCB)

Get the corresponding odometer channel's sensing direction.

Field	CMD	LEN	DATA
Value	SBG_GET_ODO_DIRECTION 0xCB	1	Channel (uint8)
Size (bytes)	1	2	1

The device's answer should be the following frame: SBG_RET_ODO_DIRECTION (0xCD).

Field	CMD	LEN	DATA	
Value	SBG_RET_ODO_DIRECTION 0xCD	2	Channel (uint8)	OdoDirection (uint8)
Size (bytes)	1	2	1	1

SBG_SET_ODO_LEVER_ARM (0xCE)

Configures the distance from the IG-500E to the odometer channel sensor, expressed in meters, in the sensor coordinate frame.

Field	CMD	LEN	DATA				
Value	SBG_SET_ODO_LEVER_ARM 0xCE	14	Reserved Leave to 0	Channel (uint8)	armX (real32)	armY (real32)	ArmZ (real32)
Size (bytes)	1	2	1	1	4	4	4

Device's answer should be an Ack..

SBG_GET_ODO_LEVER_ARM (0xCF)

Get the Odometer channel lever Arm setting

Field	CMD	LEN	DATA
Value	SBG_GET_ODO_LEVER_ARM 0xCF	1	Channel (uint8)
Size (bytes)	1	2	1

The device should answer with the following frame: SBG_RET_ODO_LEVER_ARM (0xD0).

Field	CMD	LEN	DATA			
Value	SBG_RET_ODO_LEVER_ARM 0xD0	13	Channel (uint8)	armX (real32)	armY (real32)	ArmZ (real32)
Size (bytes)	1	2	1	4	4	4

2.1.10. Advanced Options

The IG-500 now includes a set of advanced features that can be turned on or off with a single command.

Commands availability:

IG-20	IG30A	IG-30G	IG-500A	IG-500N	IG-500E
no	no	no	yes	yes	yes

All features are contained in a 32 bit bit-mask described as below:

```
SBG_SETTING_ENABLE_CONING          0x00000001   Use coning integration in the Kalman
                                                filter instead of gyroscopes values
SBG_SETTING_ALTITUDE_ABOVE_MSL     0x00000004   - 0 -> Altitude is above Ellipsoid
                                                - 1 -> Altitude is above Mean Sea Level
SBG_SETTING_DECLINATION_AUTO       0x00000008   Enable/Disable automatic magnetic
                                                declination computation
SBG_SETTING_GRAVITY_AUTO           0x00000010   Enable/Disable automatic local gravity
                                                computation
SBG_SETTING_OUTPUT_UNBIASED_GYRO   0x00000020   Enable/Disable Kalman unbiased gyroscope
                                                and delta angle outputs
SBG_SETTING_OUTPUT_UNBIASED_ACCEL  0x00000040   Enable/Disable Kalman unbiased
                                                accelerometer output
SBG_SETTING_MAG_HORIZONTAL_ONLY    0x00000080   Restrict magnetometers use to horizontal
                                                position (used in certain 2D calibrations)
SBG_SETTING_STATIC_INIT            0x00000100   Initialize the Kalman filter with he no
                                                motion assumption for 10 seconds for
                                                faster startup
SBG_SETTING_STATIC_INIT_UNTIL_MOTION_DETECTED 0x00000200 Force the Kalman filter to use
                                                the static assumption, until a motion is
                                                detected. This allows to:
                                                - Estimate gyro bias without heading
                                                - Get a faster startup
```

SBG_SET_ADVANCED_OPTIONS (0xD4,D5,D6)

This function gives access to several miscellaneous settings by the use of a bit-mask.

Field	CMD	LEN	DATA	
Value	SBG_SET_ADVANCED_OPTS (0xD4)	8	Options1 (uint32)	Reserved Leave to 0
Size (bytes)	1	2	4	4

Device's answer is an ACK.

SBG_GET_ADVANCED_OPTIONS (0xD5)

This command has no parameter and is used to retrieve the IG-500 advanced options. Device answer should be SBG_RET_ADVANCED_OPTS (0xD6):

Field	CMD	LEN	DATA	
Value	SBG_RET_ADVANCED_OPTS (0xD6)	8	Options1 (uint32)	Reserved Leave to 0
Size (bytes)	1	2	4	4

2.2. Retrieving IG devices outputs

2.2.1. Normal mode

All these commands use the standard output buffer described in section 1.6.

Commands availability:

IG-20	IG30A	IG-30G	IG-500A	IG-500N	IG-500E
yes	yes	yes	yes	yes	yes

SBG_GET_DEFAULT_OUTPUT (0x56)

This command retrieves the default output. (configured with SBG_SET_DEFAULT_OUTPUT_MASK).

It has no argument. Device's answer is:

Field	CMD	LEN	DATA
Value	SBG_RET_DEFAULT_OUTPUT 0x57	Variable	Default output buffer
Size (bytes)	1	2	Variable

SBG_GET_SPECIFIC_OUTPUT (0x58)

This command asks for a particular output buffer, specified in argument to the device.

Field	CMD	LEN	DATA
Value	SBG_GET_SPECIFIC_OUTPUT 0x58	4	Output mask (uint32)
Size (bytes)	1	2	4

The device answers with SBG_RET_SPECIFIC_OUTPUT (0x59). There is no specific error code for that command.

Field	CMD	LEN	DATA
Value	SBG_RET_SPECIFIC_OUTPUT 0x59	Variable	Specific output buffer
Size (bytes)	1	2	Variable

2.2.2. Continuous mode

All these commands use the standard output buffer described in section 1.6.

Commands availability:

IG-20	IG30A	IG-30G	IG-500A	IG-500N	IG-500E
yes	yes	yes	yes	yes	yes

SBG_CONTINUOUS_DEFAULT_OUTPUT (0x90)

This command is not sent by user. When the continuous mode is enabled, the device continuously sends that command, at a regular frequency. It expects no answer from the user.

Field	CMD	LEN	DATA
Value	SBG_ CONTINUOUS_DEFAULT_OUTPUT 0x90	Variable	Default output buffer
Size (bytes)	1	2	Variable

2.2.3. Triggered output

A triggered output frame is generate each time a trigger channel becomes valid. It contains the output buffer described in section 1.6 , plus additional data as explained below.

Commands availability:

IG-20	IG30A	IG-30G	IG-500A	IG-500N	IG-500E
yes	yes	yes	yes	yes	yes

SBG_TRIGGERED_OUTPUT (0x91)

This command is not sent by user. When one or several trigger channels are configured and activated by an event, the frame is sent to the user.

This frame contains three fields, a trigger mask used to indicate which event(s) have activated the output, an output mask that defines which data are contained in the output buffer and the triggered output buffer as described in section 1.6.

Field	CMD	LEN	DATA		
Value	SBG_ TRIGGERED_OUTPUT 0x91	8 + Output buffer size	Trigger Mask (uint32)	Output Mask (uint32)	Triggered Output buffer
Size (bytes)	1	2	4	4	Variable

2.2.4. Advanced outputs

2.2.4.1. Space Vehicle Information

SBG_GET_GPS_SVINFO (0xA3)

This command has no argument and is used to retrieve the advanced information about satellites tracked by the GPS receiver.

The device may answer with SBG_RET_GPS_SVINFO (0xA4). This answer has got a variable length, depending on the channel number of the GPS receiver (which is 16 for Antaris 4 GPS receivers, and up to 50 with the Ublox 5 GPS receiver). The frame contains the satellite information for each channel of the GPS receiver. The number of fields contained in the frame is defined by the first parameter N. N can be up to 50.

Field	CMD	LEN	DATA							
Value	SBG_ RET_GPS_SVINFO 0A4	1 + N*5	N (uint8)	GpsSvInfo					GpsSvInfo	...
				SatelliteID (uint8)	FlagsQuality (uint8)	SignalStrength (uint8)	Azimuth (int8)	Elevation (int8)		
Size (bytes)	1	2	1	1	1	1	1	1	5	...
				5						

Note : *Check IG-30G and IG-500N User Manual to get the units of the different fields.*

2.3. External aiding source

The IG-500A, IG-500N and IG-500E, devices can use external aiding source for heading, velocity and position information. This advanced feature open a wide range of applications were, for example, a GPS information is unavailable.

To understand how to use these commands, let's take a simple example, a car navigation system. When the car has a good GPS reception, the IG-500N device can use the integrated GPS receiver as an accurate source of heading, velocity and position information. As soon as the car go through a tunnel, the GPS will be unavailable and the device will start to drift regarding heading, velocity and especially position.

To avoid or limit such a drift, the user can provide an external velocity and heading information based on a measure made on the car's wheels.

Heading, as well as velocity and position aiding frames are considered as unidirectional. The device will not send any ACK to those commands in order to simplify real time operation of each target and host systems.

2.3.1. Attitude related aiding source

2.3.1.1. Heading measurements

When the device has been configured to use an external source for heading reference, this command is used to input a new heading information.

See the command `SBG_SET_FILTER_HEADING_SOURCE` for aiding source configuration.

As soon as this command is received by the device, the Kalman filter will use this new heading information to compensate for the heading drift. Each heading information is associated with an accuracy field used by the Kalman filter to determine the amount of corrections to apply.

If your measurement is very accurate (accuracy field near 0.0), the filter will rely mostly on the provided information otherwise, it will correct only a small error on the estimated heading.

The heading and accuracy fields are expressed in radians. Accuracy is considered as the measurement standard deviation.

The valid range for heading is between $[-\pi,+\pi]$ and for accuracy it is between $[0;+2\pi]$

Commands availability:

IG-20	IG30A	IG-30G	IG-500A	IG-500N	IG-500E
no	no	no	yes	yes	yes

SBG_SEND_FILTER_HEADING (0x43)

Sends a new heading information to the device.

Field	CMD	LEN	DATA	
Value	SBG_SEND_FILTER_HEADING 0x43	8	heading (real32)	accuracy (real32)
Size (bytes)	1	2	4	4

2.3.2. Navigation related aiding source

When the device has been configured to use an external aiding source for velocity, these commands are used to send a new velocity and position information.

As soon as those commands are received by the device, the Kalman filter will use this new velocity/position information to compensate for the drift in position or velocity. Each information is associated with an accuracy field used by the Kalman filter to determine the amount of corrections to apply.

If your measurement is very accurate (accuracy field near 0.0), the filter will rely mostly on the provided information otherwise, it will correct only a small error on the estimated velocity.

Commands availability:

IG-20	IG30A	IG-30G	IG-500A	IG-500N	IG-500E
no	no	no	no	yes	yes

SBG_SEND_NAV_VELOCITY (0xB4)

Sends an external velocity data. Velocity is expressed in the device coordinate system.

Accuracy is considered as the measurement standard deviation.

All fields are expressed in $m \cdot s^{-1}$

Field	CMD	LEN	DATA			
Value	SBG_SEND_NAV_VELOCITY 0xB4	16	vX (real32)	vY (real32)	vZ (real32)	accuracy (real32)
Size (bytes)	1	2	4	4	4	4

Note: *Check the command* `SBG_SET_NAV_VELOCITY_SRC` *for aiding source configuration.*

SBG_SEND_NAV_POSITION (0xB5)

The position is in North, East, Height, and in [deg, deg, meters].

The horizontal and vertical accuracies are both expressed in meters (standard deviation).

Field	CMD	LEN	DATA				
Value	SBG_SEND_NAV_POSITION 0xB5	32	north (real64)	east (real64)	height (real64)	hAcc (real32)	vAcc (real32)
Size (bytes)	1	2	8	8	8	4	4

3. IG-20 and IG-30 Specific commands

3.1. Filters frequencies

These commands set or get the different frequencies involved in the device: Accelerometers, gyroscopes and magnetometers cut-off frequencies, and Main loop frequency.

Command availability:

IG-20	IG30A	IG-30G	IG-500A	IG-500N	IG-500E
yes	yes	yes	no	no	no

SBG_SET_FILTER_FREQUENCIES (0x30)

This command allows to set all configurable frequencies,

Field	CMD	LEN	DATA					
Value	SBG_SET_FILTER_FREQUENCIES 0x30	21	Reserved Set to 0 (uint8)	Reserved Set to 0.0 (real32)	Cut-off gyro. (real32)	Cut-off accel. (real32)	Cut-off mag. (real32)	Main freq. (real32)
Size (bytes)	1	2	1	4	4	4	4	4

***Note 1 :** Passing 0 in arguments can be used to make the system ignore this parameter without any error. It is then possible to change only one of the five parameters at a time.*

***Note 2 :** The main loop filter cannot exactly be set to any frequency. Actually, the device counts a fixed number of milliseconds for the Kalman filter loop. If user sets a frequency that does not fit exactly with the system, the device will choose the nearest lower frequency. Please read the user manual for more information about that.*

Possible Error Codes	Possible Reason
`SBG_INVALID_PARAMETER`	User chose a non valid frequency for one or many parameters → *Choose only valid frequencies, or set argument to 0 to ignore it.*

SBG_GET_FILTER_FREQUENCIES (0x31)

This command retrieves the different frequencies of the system. It has no parameter.

The device should return SBG_RET_FILTER_FREQUENCIES (0x32)

Field	CMD	LEN	DATA				
Value	SBG_RET_FILTER_FREQUENCIES 0x32	20	Reserved Set to 0.0 (real32)	Cut-off gyro. (real32)	Cut-off accel. (real32)	Cut-off mag. (real32)	Main freq. (real32)
Size (bytes)	1	2	4	4	4	4	4

3.2. Orientation filter Options

Commands used to enable or disable orientation computation by an IG-20 / IG-30

Possible options are:

```
SBG_FILTER_OPTION_ENABLE_ATTITUDE                    0x10 (default)
```

By default, all options are enabled.

Note : *If user do not set any of these masks, All options will be disabled*

Command availability:

IG-20	IG30A	IG-30G	IG-500A	IG-500N	IG-500E
yes	yes	yes	no	no	no

SBG_SET_FILTER_ATTITUDE_OPTIONS (0x36)

Configure Kalman filter options.

Field	CMD	LEN	DATA	
Value	SBG_SET_FILTER_ATTITUDE_OPTIONS 0x36	5	Reserved Leave to 0 (uint8)	Filter Attitude mode (uint32)
Size (bytes)	1	2	1	4

SBG_GET_FILTER_ATTITUDE_OPTIONS (0x37)

This command is used to retrieve the different options of the Kalman filter.

The device may answer SBG_RET_FILTER_ATTITUDE_OPTIONS (0x38)

Field	CMD	LEN	DATA
Value	SBG_RET_FILTER_ATTITUDE_OPTIONS 0x38	4	Filter Attitude Options (uint32)
Size (bytes)	1	2	4

3.3. IG-30G GPS receiver Options

Set of commands used to configures or check some advanced GPS options:

- Dynamic model used for position calculation.
- SBAS Configuration
- Altitude reference for GPS position: Altitude above ellipsoid or altitude above Mean Sea Level.

Possible values for model are:

SBG_GPS_MODEL_STATIONARY	0x01	
SBG_GPS_MODEL_PEDESTRIAN	0x02	
SBG_GPS_MODEL_AUTOMOTIVE	0x03	
SBG_GPS_MODEL_SEA	0x04	
SBG_GPS_MODEL_AIRBONE_1G	0x05	
SBG_GPS_MODEL_AIRBONE_2G	0x06	(Default)
SBG_GPS_MODEL_AIRBONE_4G	0x07	

Possible masks for options are :

SBG_GPS_DISABLE_SBAS	0x00	
SBG_GPS_ENABLE_SBAS_DIFF_CORRECTIONS	0x01	(Default)
SBG_GPS_ENABLE_SBAS_RANGING	0x02	(Default)
SBG_GPS_ALTITUDE_ABOVE_MSL	0x04	

Command availability:

IG-20	IG30A	IG-30G	IG-500A	IG-500N	IG-500E
no	no	yes	no	yes	no

SBG_SET_GPS_OPTIONS (0xA5)

Configure GPS options.

Field	CMD	LEN	DATA		
Value	SBG_SET_GPS_OPTIONS 0xA5	3	Reserved Leave to 0 (uint8)	model (uint8)	options (uint8)
Size (bytes)	1	2	1	1	1

Device's answer is an ACK.

SBG_GET_GPS_OPTIONS (0xA6)

This command has no argument and is used to retrieve the GPS options. See SBG_SET_GPS_OPTIONS for arguments description.

The device answer `SBG_RET_GPS_OPTIONS (0xA7)`

Field	CMD	LEN	DATA	
Value	SBG_RET_GPS_OPTIONS 0xA7	2	model (uint8)	options (uint8)
Size (bytes)	1	2	1	1

4. ASCII and NMEA protocol output

4.1. Protocol overview

This ASCII protocol output has been implemented to facilitate IG-500 integration into existing systems.

Only a limited implementation is performed here: There is no NMEA/ASCII input allowed, and the main IG-Devices serial protocol should be used to configure the IG-Device, including ASCII output configuration.

4.1.1. NMEA sentences format

The NMEA sentences implemented in the IG-500 are based on NMEA 0183 Version 2.3.

All NMEA frames comply with the following general format:

Field	**Start of frame**	**Talker ID**	**Sentence Formater**	**{,value}**	**Check-sum**	**End of frame**
Value	$	<XX>	<XXX>		*<checksum>	<CR><LF>
Description	All frames start with $	**GP** stands for GPS. Other Ids exist for other devices	This 3 characters field define the message content	Data fields are separated by a ','. Data field length can vary, even for a certain field.	Starts with a '*' and consist of a 2 characters representing a hex 8 bit value. The checksum is the XOR of all previous characters excluding $ Checksum range corresponds to the blue cells below:	All frames end with a carriage return and line feed.
Example	`$`	`GP`	`ZDA`	`,201530.00,04,07,2002,00,00`	`*60`	`<CR><LF>`

Example frame:

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

Note: *Hence there are several talker Ids and some might refer to inertial navigation systems, the GP talker ID is the most common ID. In addition, some NMEA decoders would not accept anything else than GP.*

That's why the IG-500 will only output frames with a talker ID set to “GP”'.

4.1.2. Proprietary ASCII sentences format

These SBG Systems proprietary sentences follow a very similar protocol that should be compatible with most NMEA decoders.

The only difference is that the talker ID and Sentence formatter are replaced by the three letters SBG and a two digits number corresponding to the sentence ID.

Field	Start of frame	Sentence ID	{,value}	Check-sum	End of frame
Value	$	<SBGXX>		*<checksum>	<CR><LF>
Description	All frames start with $	Sentence ID encoded in with SBG followed by two digits.	Data fields are separated by a ','. Data field length can vary, even for a certain field.	Starts with a '*' and consist of a 2 characters representing a hex 8 bit value. The checksum is the XOR of all previous characters excluding $	All frames end with a carriage return and line feed.
Example	`$`	`SBG01`	`,010605.18,-000.34,-06.67,007.36,001.49`	`*60`	`<CR><LF>`

Example frame:

```
$SBG01,010605.18,-000.34,-06.67,007.36,001.49*60<CR><LF>
```

4.1.3. Fields format

All fields are composed of ASCII characters.

Some fields defined as “decimal” can have a variable length, whereas fields defined as “xx.yyy” have a fixed length.

4.1.3.1. Time format

Time is provided as “hhmmss.ss”. It means that if a time parameter is set to “105621.02”, the UTC time is 10:56:21.02 UTC

4.1.3.2. Latitude and Longitude formats

These angles are expressed as degrees and minutes (“ddmm.mmmmm” for latitude).

A latitude field set to “4543.123450” means:

- 45°43.1234'
- 45° 43' 7.416”
- 45.718723333°

Note: *Longitudes contain 3 digits for degrees whereas latitudes only contain 2 digits.*

4.2. Sentences description

4.2.1. NMEA GPGGA

This frame contain positioning data as well as accuracy data and altitude.

Frame format is the following:

```
$GPGGA,hhmmss.ss,Latitude,N,Longitude,E,FS,NoSV,HDOP,msl,m,Altref,m,DiffAge,DiffStation*cs<CR><LF>
```

Field	Name	Format	Description
0	$GPGGA	string	Message ID – GGA frame
1	Time	hhmmss.ss	UTC time, current time
2	Latitude	ddmm.mmmmm	Latitude: Degrees + Minutes
3	N	char	North / South indicator. N = North ; S = South
4	Longitude	dddmm.mmmmm	Latitude: Degrees + Minutes
5	E	char	East / West indicator. E = East ; W = West
6	FS	digit	Fix status. See definition below.
7	NoSV	integer	Number of satellites used in solution
8	HDOP	Decimal number	Dilution of precision
9	msl	Decimal number	Altitude abode MSL. May be negative.
10	M	M	Altitude unit (Meter) fixed field.
11	AltRef	Decimal number	Geoid separation
12	M	M	Altitude unit (Meter) fixed field.
13	DiffAge	-	Age of differential corrections. Not being output by this device.
14	Diff station	-	Diff reference station. Not being output by this device.
15	Check sum	*cs	Checksum: Xor of all previous bytes excluding $
16	End of frame	<CR><LF>	End of frame : Carriage return and line feed.

Fix Status definition

Value	Meaning
0	No Fix
1	GPS fix
2	GPS with DGPS
4	RTK fix
5	Floating RTK fix
6	Dead reckoning fix

Example frame:

```
$GPGGA,010843.28,4852.13785,N,00209.48994,E,1,07,2.4,30.51,M,-47.27,M,,*75<CR><LF>
```

4.2.2. NMEA GPRMC

This is the “minimum recommended data” frame. It contains 2D position data as well as velocity and course over ground.

Frame format is the following:

```
$GPRMC,hhmmss,status,latitude,N,longitude,E,spd,cog,ddmmyy,mv,mvE,mode*cs<CR><LF>
```

Field	Name	Format	Description
0	$GPRMC	string	Message ID – RMC frame
1	Time	hhmmss.ss	UTC time, current time
2	Status	char	Status field: A = Data valid. V = Warning See description below
3	Latitude	ddmm.mmmmm	Latitude: Degrees + Minutes
4	N	char	North / South indicator. N = North ; S = South
5	Longitude	dddmm.mmmmm	Latitude: Degrees + Minutes
6	E	char	East / West indicator. E = East ; W = West
7	Spd	Decimal number	Speed over ground in Knots
8	Cog	Decimal number	Course over ground in Degrees. [0; 360°]
9	date	ddmmyy	Date , month, year
10	mv	-	Magnetic variation value (not being output by device).
11	mvE	-	Magnetic variation E/W indication. Not being output by device.
12	mode	char	Mode indicator. See description below
13	Check sum	*cs	Checksum: Xor of all previous bytes excluding $
14	End of frame	<CR><LF>	End of frame : carriage return and line feed.

Mode indicator field and status field values

Field	No position fix (not initialized or Pos. error > 500m)	Dead reckoning error > 50m	Dead reckoning error < 50m	GPS Fix available	GPS Fix available with diff. corrections
Status	V	V	A	A	A
Mode	N	E	E	A	D

Example frame:

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

4.2.3. NMEA GPZDA

Time and date output.

```
$GPZDA,hhmmss.ss,day,month,year,ltzh,ltzn*cs<CR><LF>
```

Field	Name	Format	Description
0	$GPZDA	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 to 00)
6	ltzn	0	Local zone minutes (not supported, fixed to 00)
7	Check sum	*cs	Checksum: Xor of all previous bytes excluding $
8	End of frame	<CR><LF>	End of frame : carriage return and line feed.

Example frame:

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

4.2.4. ASCII SBG01

Time and Euler angles are provided in this frame.

```
$SBG01,hhmmss.ss,roll,pitch,yaw,acc*cs<CR><LF>
```

Field	Name	Format	Description
0	$GPZDA	string	Message ID – SBG01 message
1	Time	hhmmss.ss	UTC time, current time
2	roll	[-]ddd.dd	Roll angle in degrees [-180; 180]
3	pitch	[-]dd.dd	Pitch angle in degrees [-90;90]
4	yaw	[-]ddd.dd	Yaw angle in degrees [-180; 180]
5	acc	ddd.dd	Estimated Accuracy
7	Check sum	*cs	Checksum: Xor of all previous bytes excluding $
8	End of frame	<CR><LF>	End of frame : carriage return and line feed.

Example frame:

```
$SBG01,010605.18,-000.34,-06.67,007.36,001.49*60<CR><LF>
```

4.2.5. NMEA HEHDT

Heading, True.

```
$HEHDT,heading,T*cs<CR><LF>
```

Field	Name	Format	Description
0	$HEHDT	string	Message ID – HEHDT message
1	heading	ddd.dd	Heading angle, [0;360]
2	T	c	Heading true
7	Check sum	*cs	Checksum: Xor of all previous bytes excluding $
8	End of frame	<CR><LF>	End of frame : carriage return and line feed.

Example frame:

```
$HEHDT,172.01,T*1A<CR><LF>
```

4.2.6. KVH Extended

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

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

Field	Name	Format	Description
0	%	string	Message ID – HEHDT message
1	pitch	ddd	Pitch angle – in tenths of degrees. Divide by 10 to get pitch in °.
2	roll	ddd	Roll angle – in tenths of degrees. Divide by 10 to get roll in °.
7	heading	ddd	Heading in tenths of degrees. From 0 to 3600.
8	Heading rate	ddd	Heading rate, in hundredths of degrees per seconds.
9	End of frame	<CR><LF>	End of frame : carriage return and line feed.

Example frame:

```
%10,-5,3489,11<CR><LF>
```

4.2.7. NMEA HEHDM

Heading, Magnetic.

```
$HEHDM,heading,M*cs<CR><LF>
```

Field	Name	Format	Description
0	$HEHDM	string	Message ID – HEHDT message
1	heading	ddd.dd	Heading angle, [0;360]
2	M	c	Heading , magnetic
7	Check sum	*cs	Checksum: Xor of all previous bytes excluding $
8	End of frame	<CR><LF>	End of frame : carriage return and line feed.

Example frame:

```
$HEHDM,167.76,M*1E<CR><LF>
```

4.2.8. PSXN,23

Euler angles, and heave information.

```
$PSXN,23,roll,pitch,heading,heave*cs<CR><LF>
```

Field	Name	Format	Description
0	$GPZDA	string	Message ID – SBG01 message
1	Time	hhmmss.ss	UTC time, current time
2	roll	[-]ddd.dd	Roll angle in degrees [-180; 180]
3	pitch	[-]dd.dd	Pitch angle in degrees [-90;90]
4	heading	ddd.dd	Yaw angle in degrees [0; 360]
5	heave	ddd.dd	Heave position in meters (positive down)
7	Check sum	*cs	Checksum: Xor of all previous bytes excluding $
8	End of frame	<CR><LF>	End of frame : carriage return and line feed.

Example frame:

```
$PSXN,23,0.25,0.55,163.47,-0.00*15<CR><LF>
```

5. Support

If you have any question, comment or problem with the use of this low level protocol, we would be glad to help you, so please feel free to contact us. Please do not forget to mention your Device ID of your IG devices (written on your device's label).

You can contact us by:

- Email : support@sbg-systems.com
- Phone : +33 (0)1 80 88 45 00