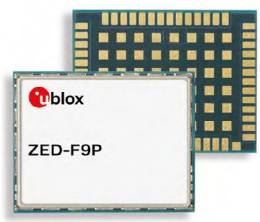
u-blox F9 HPG 1.30

u-blox F9 high precision GNSS receiver

Interface Description



Abstract
This document describes the interface (version 27.30) of the ZED-F9P, a multi-band GNSS module with integrated RTK offering centimeter level accuracy.

www.u-blox.com

UBX-21046737 - R01
C1-Public

Document information

Title	u-blox F9 HPG 1.30	
Subtitle	u-blox F9 high precision GNSS receiver	
Document type	Interface Description	
Document number	UBX-21046737	
Revision and date	R01	16-Dec-2021
Disclosure restriction	C1-Public	

Contents

1 General information

1.1 Document overview

This document describes the interface of the u-blox F9 high precision GNSS receiver . The interface consists of the following parts:

- NMEA protocol
- UBX protocol
- RTCM protocol
- SPARTN protocol
- Configuration interface

☞ Some of the features described here may not be available in the receiver, and some may require specific configurations to be enabled. See the data sheet for availability of the features and the integration manual for instructions for enabling them.

☞ Previous versions of u-blox receiver documentation combined general receiver description and interface specification. In the current documentation the receiver description is included in the integration manual.

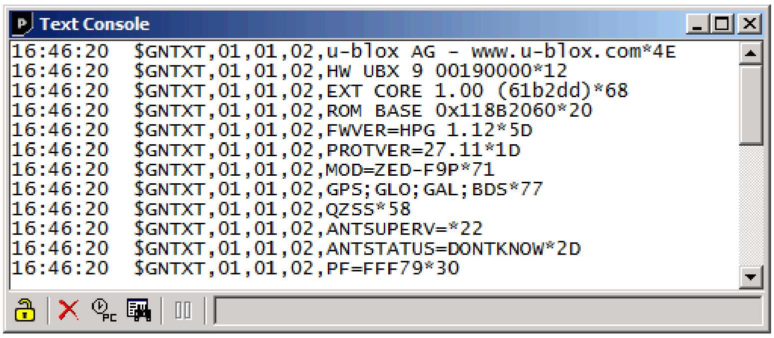
See also Related documents.

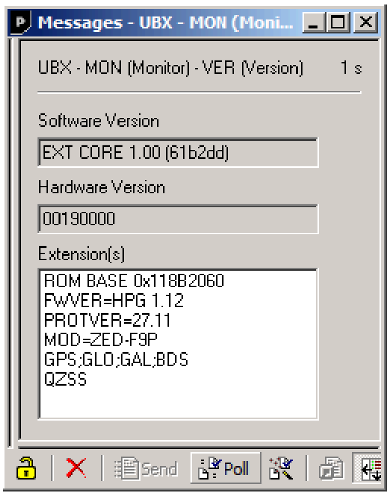
1.2 Firmware and protocol versions

u-blox generation 9 receivers execute firmware from internal ROM or from internal code-RAM. If the firmware image is stored in a flash it is loaded into the code-RAM before execution. It is also possible to store the firmware image in the host system. The firmware is then loaded into the code-RAM from the host processor. (Loading the firmware from the host processor is not supported in all products.) If there is no external firmware image, then the firmware is executed from the ROM.

The location and the version of the boot loader and the currently running firmware can be found in the boot screen and in the UBX-MON-VER message. If the firmware has been loaded from a connected flash or from the host processor, it is indicated by text "EXT". When the receiver is started, the boot screen is output automatically in UBX-INF-NOTICE or NMEA-Standard-TXT messages if configured using CFG-INFMSG. The UBX-MON-VER message can be polled using the UBX polling mechanism.

The following u-center screenshots show an example of a u-blox receiver running firmware loaded from flash:





The following information is available (✓) from the boot screen (**B**) and the UBX-MON-VER message (**M**):

B	M	Example	Information
✓		`u-blox AG - www.u-blox.com`	Start of the boot screen.
✓		`HW UBX 9 00190000`	Hardware version of the u-blox receiver.
	✓	`00190000`	
✓	✓	`EXT CORE 1.00 (61b2dd)`	Base (CORE) firmware version and revision number, loaded from external memory (EXT).
		`EXT LAP 1.00 (12a3bc)`	Product firmware version and revision number, loaded from external memory (EXT). Available only in some firmware versions. See below for a list of product acronyms.
✓	✓	`ROM BASE 0x118B2060`	Revision number of the underlying boot loader firmware in ROM.
✓	✓	`FWVER=HPG 1.12`	Product firmware version number, where: • `SPG` = Standard precision GNSS product • `HPG` = High precision GNSS product • `ADR` = Automotive dead reckoning product • `TIM` = Time sync product • `LAP` = Lane accurate positioning product • `HPS` = High precision sensor fusion product • `DBS` = Dual band standard precision • `MDR` = Multi-mode dead reckoning product • `PMP` = L-Band Inmarsat point-to-multipoint receiver • `QZS` = QZSS L6 centimeter level augmentation service (CLAS) message receiver • `DBD` = Dual band dead reckoning product
✓	✓	`PROTVER=34.00`	Supported protocol version.
✓	✓	`MOD=ZED-F9P`	Module name (if available).
✓	✓	`GPS;GLO;GAL;BDS`	List of supported major GNSS (see GNSS identifiers).
✓	✓	`SBAS;QZSS`	List of supported augmentation systems (see GNSS identifiers).
✓		`ANTSUPERV=AC SD PDoS SR`	Configuration of the antenna supervisor (if available), where: • `AC` = Active antenna control enabled • `SD` = Short circuit detection enabled • `OD` = Open circuit detection enabled • `PDoS` = Short circuit power down logic enabled • `SR` = Automatic recovery from short state enabled

B	M	Example	Information
✓		`PF=FFF79`	Product configuration.
✓		`BD=E01C`	GNSS band configuration.

☞ The "FWVER" product firmware version indicates which firmware is currently running. This is referred to as "firmware version" in this and other documents.

☞ The revision numbers should only be used to identify a known firmware version. They are not necessarily numeric nor are they guaranteed to increase with newer firmware versions.

☞ Similarly, firmware version numbers can have additional non-numeric information appended, such as in "5.00B03".

☞ Not every entry is output by all u-blox receivers. The availability of some of the information depends on the product, the firmware location and the firmware version.

The product firmware version and the base firmware version relate to the protocol version:

Product firmware version	Base firmware version	Protocol version
HPG 1.30	EXT CORE 1.00 (9acf11)	27.30

1.3 Receiver configuration

u-blox positioning receivers are fully configurable with UBX protocol messages. The configuration used by the receiver during normal operation is called the "current configuration". The current configuration can be changed during normal operation by sending UBX-CFG-VALSET messages over any I/O port (except UART2). The receiver will change its current configuration immediately after receiving a configuration message. The receiver will always use the current configuration only.

The current configuration is loaded from permanent configuration hard-coded in the receiver firmware (the defaults) and from non-volatile memory (user configuration) on startup of the receiver. Changes made to the current configuration at run-time will be lost when there is a power cycle, a hardware reset or a (complete) controlled software reset (see Configuration reset behavior).

See Configuration interface for a detailed description of the receiver configuration system, the explanation of the configuration concept and its principles and interfaces.

☞ The configuration interface has changed from earlier u-blox positioning receivers. There is some backwards compatibility provided in UBX-CFG configuration messages. Users are strongly advised to only use the Configuration interface. See also Legacy UBX message fields reference.

☞ See the integration manual for a basic receiver configuration most commonly used.

1.4 Naming

Message names are written in full with the parts of the name separated by hyphens ("-"). The full message name consists of the protocol name (e.g., *UBX)*, the class name (e.g. *NAV*) and the message name (e.g. *PVT*). For example the receiver software version information message is referred to as *UBX-MON-VER*. Similarly, the *NMEA-Standard-GGA* is the NMEA standard message (sentence) with the global positioning fix data.

References to fields of the message add the field name separated by a dot ("."), e.g. *UBX-MON-VER.swVersion*.

Some messages use a fourth level of naming, called the message version. One example is the *UBX-MGA-GPS* message for GPS assistance data, which exists in versions for ephemerides (*UBX-MGA-GPS-EPH*) and almanacs (*UBX-MGA-GPS-ALM*).

Names of configuration items are of the form *CFG-GROUP-ITEM*. For example, *CFG-NAVSPG-DYNMODEL* refers to the navigation dynamic platform model the receiver uses. Constants add a fourth level to the item name, such as *CFG-NAVSPG-DYNMODEL-AUTOMOT* for the automotive platform model. In the context of describing an item's value, only the last part of the constant name can be used (e.g. "set *CFG-NAVSPG-DYNMODEL* to *PORT* for portable applications").

1.5 GNSS, satellite and signal identifiers

1.5.1 Overview

The UBX protocol messages use two different numbering schemes. Some messages use a one-byte (type U1) field for the satellite identifier (normally named `svid`). This uses numbering similar to the "extended" NMEA scheme and is merely an extension of the scheme in use for previous generations of u-blox receivers.

With the ever increasing numbers of GNSS satellites, this scheme has been phased out in recent u-blox positioning receivers (as numbers greater than 255 would have become necessary). Consequently, newer messages use a more sophisticated, flexible and future-proof approach. This involves having a separate `gnssId` field to identify which GNSS the satellite is part of and a simple `svId` (SV for space vehicle) field that indicates which number the satellite is in that system. In nearly all cases, this means that the *svId* is the natural number associated with the satellite. For example the GLONASS SV4 is identified as *gnssId 6, svId 4*, while the GPS SV4 is *gnssId 0, svId 4*.

Signal identifiers are used where different signals from a GNSS satellite need to be distinguished (e.g. in the UBX-NAV-SIG message). A separate `sigId` field is used. These identifiers are only valid when combined with a GNSS identifier (`gnssId` field).

The NMEA protocol (version 4.10 and later) identifies GNSS satellites with a one-digit system ID and a two-digit satellite number. u-blox receivers support this method in their NMEA output when "strict" SV numbering is selected. In most cases this is the default setting, but it can be checked or changed using the Configuration interface (see also NMEA GNSS, satellite and signal numbering).

In order to support some GNSS (e.g. BeiDou, Galileo, QZSS), which are not supported by some or all NMEA protocol versions, an "extended" SV numbering scheme can be enabled. This uses the NMEA-defined numbers where possible but adds other number ranges to support other GNSS. Note however that these non-standard extensions require 3-digit numbers, which may not be supported by some NMEA parsing software. For example, QZSS satellites use numbers in the range 193 to 202.

The NMEA standard defines signal identifiers to distinguish different signals sent by a single GNSS satellite (e.g. L2 CL and CM). u-blox positioning receivers use those identifiers for signal identification, as far as the corresponding standard is supported in a particular product.

☞ Note that the following sections are a generic overview for different u-blox positioning receivers. A particular product may not support all of the described GNSS identifiers, satellite numbers, signal identifiers or combinations thereof.

1.5.2 GNSS identifiers

The following table lists each GNSS along with the GNSS identifier (UBX protocol) , the NMEA system identifiers (NMEA protocol), and abbreviations used in this document:

GNSS	Abbreviations		UBX gnssId	NMEA system ID		
				2.3 - 4.0	4.10	4.11
GPS	GPS	G	0	1	1	1
SBAS	SBAS	S	1	1	1	1
Galileo	GAL	E	2	n/a	3	3
BeiDou	BDS	B	3	n/a	(4)[1]	4
IMES	IMES	I	4	n/a	n/a	n/a
QZSS	QZSS	Q	5	n/a	(1)[1]	5
GLONASS	GLO	R	6	2	2	2
NavIC	NavIC	N	7	n/a	n/a	6

Other values will be added when support for other GNSS types will be enabled in u-blox receivers.

See also NMEA Talker ID.

1.5.3 Satellite identifiers

A summary of all the satellite numbering schemes used in the NMEA protocol and the UBX protocol is provided in the following table.

		UBX Protocol		NMEA Protocol 2.3 - 4.0		NMEA Protocol 4.10		NMEA Protocol 4.11	
GNSS	SV Range	gnssId:svId	single svid	(strict)	(extended)	(strict)	(extended)	(strict)	(extended)
GPS	G1-G32	0:1-32	1-32	1-32	1-32	1-32	1-32	1-32	1-32
SBAS	S120-S158	1:120-158	120-158	33-64	33-64, 152-158	33-64	33-64, 152-158	33-64	33-64, 152-158
Galileo	E1-E36	2:1-36	211-246	-	301-336	1-36	1-36	1-36	1-36
BeiDou	B1-B5	3:1-5	159-163	-	401-405	1-5	1-5	1-5	1-5
	B6-B37	3:6-37	33-64	-	406-437	6-37	6-37	6-37	6-37
	B38-B63	3:38-63	n/a	-	438-463	38-63	38-63	38-63	38-63
IMES	I1-I10	4:1-10	173-182	n/a	173-182	n/a	173-182	n/a	173-182
QZSS	Q1-Q10	5:1-10	193-202	n/a	193-202	n/a	193-202	1-10	1-10
GLONASS	R1-R32	6:1-32	65-96	65-96	65-96	65-96	65-96	65-96	65-96
	R?	6:255	255	null	null	null	null	null	null
NavIC	N1-N7	7:1-7	247-253	n/a	n/a	n/a	n/a	n/a	n/a

[1] While not defined by NMEA 4.10, u-blox receivers in this mode will use system ID 4 for BeiDou and, if extended satellite numbering is enabled, system ID 1 for QZSS.

GLONASS satellites can be tracked before they have been identified. In UBX messages such unknown satellites will be reported with svId 255. In NMEA messages they will be null (empty) fields. Product-related documentation and u-center will use R? to label unidentified GLONASS satellites.

1.5.4 Signal identifiers

A summary of all the signal identification schemes used in the NMEA protocol and the UBX protocol is provided in the following table. (Only a subset of the signals is supported by each product.) In NMEA the system and signal identifiers are in hexadecimal format.

	UBX Protocol		NMEA Protocol 4.10		NMEA Protocol 4.11	
Signal	gnssId	sigId	System ID	Signal ID	System ID	Signal ID
GPS L1C/A[2]	0	0	1	1	1	1
GPS L2 CL	0	3	1	6	1	6
GPS L2 CM	0	4	1	5	1	5
GPS L5 I	0	6	1	7	1	7
GPS L5 Q	0	7	1	8	1	8
SBAS L1C/A[2]	1	0	1	1	1	1
Galileo E1 C[2]	2	0	3	7	3	7
Galileo E1 B[2]	2	1	3	7	3	7
Galileo E5 aI	2	3	3	1	3	1
Galileo E5 aQ	2	4	3	1	3	1
Galileo E5 bI	2	5	3	2	3	2
Galileo E5 bQ	2	6	3	2	3	2
BeiDou B1I D1[2]	3	0	(4)[3]	(1)[4]	4	1
BeiDou B1I D2[2]	3	1	(4)[3]	(1)[4]	4	1
BeiDou B2I D1	3	2	(4)[3]	(3)[4]	4	B
BeiDou B2I D2	3	3	(4)[3]	(3)[4]	4	B
BeiDou B1C	3	5	(4)[3]	N/A	4	3
BeiDou B2a	3	7	(4)[3]	N/A	4	5
QZSS L1C/A[2]	5	0	(1)[3]	(1)[4]	5	1
QZSS L1S	5	1	(1)[3]	(4)[4]	5	4
QZSS L2 CM	5	4	(1)[3]	(5)[4]	5	5
QZSS L2 CL	5	5	(1)[3]	(6)[4]	5	6
QZSS L5 I	5	8	(1)[3]	N/A	5	7
QZSS L5 Q	5	9	(1)[3]	N/A	5	8
GLONASS L1 OF[2]	6	0	2	1	2	1
GLONASS L2 OF	6	2	2	3	2	3
NavIC L5 A	7	0	N/A	N/A	6	1

2 UBX messages that do not have an explicit `sigId` field contain information about the subset of signals marked.

3 While not defined by NMEA 4.10, u-blox receivers in this mode will use system ID 4 for BeiDou and, if extended satellite numbering is enabled, system ID 1 for QZSS.

4 BeiDou and QZSS signal ID are not defined in the NMEA protocol version 4.10. Values shown in the table are only valid for u-blox products and, for QZSS signal ID, if extended satellite numbering is enabled.

1.6 Message types

The following message types are defined:

Message type	Description
Input	Messages that are input to the receiver and never output. E.g. UBX-MGA-GPS-EPH.
Output	Messages that are output by the receiver in no particular interval and never input. E.g. UBX-ACK-ACK.
Input/output	Messages that can be output by or input to the receiver. E.g. UBX-MGA-DBD-DATA0.
Periodic	Messages that are output in regular intervals but cannot be polled. E.g. UBX-NAV-EOE.
Periodic/polled	Messages that are output in regular intervals and can be polled. E.g. UBX-NAV-PVT.
Command	Messages that are a command to the receiver. Similar to type *Input* these are input-only. E.g. UBX-CFG-RST.
Get	Output-only configuration or command messages. E.g. UBX-CFG-DAT.
Set	Input-only configuration or command messages. E.g. UBX-CFG-VALDEL.
Get/set	Input/output configuration or command messages. E.g. UBX-CFG-NAVX5.
Polled	Non-periodic messages that can only be polled. E.g. UBX-MON-VER.
Poll request	Poll request. E.g. UBX-MGA-DBD-POLL.

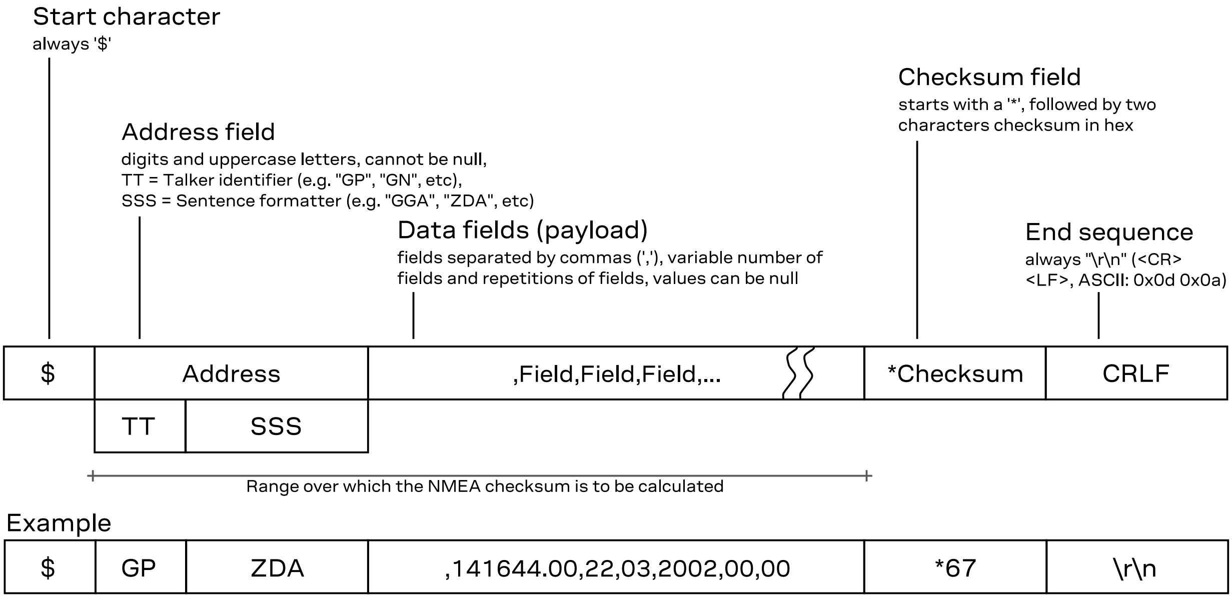
2 NMEA protocol

The following sections give an overview of the NMEA messages used by u-blox positioning receivers.

By default, the NMEA messages sent by u-blox positioning receivers are based on the NMEA 0183 version 4.11 standard. For further information on the NMEA standard, refer to the *NMEA 0183 Standard for Interfacing Marine Electronic Devices*, Version 4.11, November 2018, which is available on http://www.nmea.org/.

2.1 NMEA frame structure

The following figure shows the structure of a NMEA protocol message (called "sentences" in the standard).



2.2 NMEA protocol configuration

The NMEA protocol on u-blox receivers can be configured for customer applications by using the Configuration interface (CFG-NMEA-* items).

Several NMEA standard versions are supported. Version 4.11 (not in all products), 4.10, 4.00, 2.3, or 2.1 can be configured. See Configuration defaults for the default version. See CFG-NMEA-PROTVER to configure the version. See NMEA multi-GNSS operation and NMEA data fields for details on how this affects the output.

The following filtering flags can be used to configure the output of some NMEA message fields:

Filter	Configuration Item	Description
Position filtering	CFG-NMEA-OUT_INVFIX	Enable to permit positions from failed or invalid fixes to be reported (with the "V" status flag to indicate that the data is not valid).
Valid position filtering	CFG-NMEA-OUT_MSKFIX	Enable to permit positions from invalid fixes to be reported (with the "V" status flag to indicate that the data is not valid).
Time filtering	CFG-NMEA-OUT_INVTIME	Enable to permit the receiver's best knowledge of time to be output, even though it might be wrong.

Filter	Configuration Item	Description
Date filtering	CFG-NMEA-OUT_INVDATE	Enable to permit the receiver's best knowledge of date to be output, even though it might be wrong.
GPS-only filtering	CFG-NMEA-OUT_ONLYGPS	Enable to restrict output to only report GPS satellites.
Track filtering	CFG-NMEA-OUT_FROZENCOG	Enable to permit course over ground (COG) to be reported even when it would otherwise be frozen.

The following filtering flags can be used to configure the output of some NMEA message flags:

Mode	Configuration Item	Description
Compatibility mode	CFG-NMEA-COMPAT	Some older NMEA applications expect the NMEA output to be formatted in a specific way, for example, they will only work if the latitude and longitude have exactly four digits behind the decimal point. u-blox receivers offer a compatibility mode to support these legacy applications.
Consideration mode	CFG-NMEA-CONSIDER	u-blox receivers use a sophisticated signal quality detection scheme, in order to produce the best possible position output. This algorithm considers all SV measurements, and may eventually decide to only use a subset thereof, if it improves the overall position accuracy. If consideration mode is enabled, all satellites, which were considered for navigation, are communicated as being used for the position determination. If consideration mode is disabled, only those satellites which after the consideration step remained in the position output are marked as being used.
Limit length mode	CFG-NMEA-LIMIT82	Enabling this mode will limit the NMEA sentence length to a maximum of 82 characters.
High precision mode	CFG-NMEA-HIGHPREC	Enabling this mode increases precision of the position output. Latitude and longitude then have seven digits after the decimal point, and altitude has three digits after the decimal point. Note: The high precision mode cannot be set in conjunction with either compatibility mode or Limit82 mode.

The following extended configuration options are available:

Option	Configuration Item(s)	Description
GNSS to filter	CFG-NMEA-FILT_GPS etc.	Filters satellites based on the GNSS they belong to.
Satellite numbering	CFG-NMEA-SVNUMBERING	This field configures the display of satellites that do not have an NMEA-defined value. Note: this does not apply to satellites with an unknown ID. See also Satellite identifiers.
Main Talker ID	CFG-NMEA-MAINTALKERID	By default the main Talker ID (i.e. the Talker ID used for all messages other than GSV) is determined by the GNSS assignment of the receiver's channels (see configuration items CFG-SIGNAL-*). This field enables the main Talker ID to be overridden. See also NMEA Talker ID.
GSV Talker ID	CFG-NMEA-GSVTALKERID	By default the Talker ID for GSV messages is GNSS-specific (as defined by NMEA). This field enables the GSV Talker ID to be overridden.
BDS Talker ID	CFG-NMEA-BDSTALKERID	By default the Talker ID for BeiDou is "GB". This field enables the BeiDou Talker ID to be overridden.

2.3 NMEA-proprietary messages

The NMEA standard allows for proprietary, manufacturer-specific messages to be added. These shall be marked with a manufacturer mnemonic. The mnemonic assigned to u-blox is UBX and is used for all non-standard messages. These proprietary NMEA messages therefore have the address field set to PUBX. The first data field in a PUBX message identifies the message number with two digits.

2.4 NMEA multi-GNSS operation

Many applications that process NMEA messages assume that only a single GNSS is active. However, when multiple GNSS are configured, the NMEA specification requires the output to change in the following ways:

Main Talker ID The main NMEA Talker ID will be "GN" (e.g. instead of "GP" for a GPS-only receiver).

GSV Talker IDs The GSV message reports the signal strength of the visible satellites. However, the Talker ID it uses is specific to the GNSS it is reporting information for, so for a multi-GNSS receiver it will not be the same as the main Talker ID. While other messages use the "GN" Talker ID, the GSV message will use GNSS-specific Talker IDs. See also NMEA protocol configuration.

Multiple GSA and GRS messages Multiple GSA and GRS messages are output for each fix, one for each GNSS. This may confuse applications that assume they are output only once per position fix (as is the case for a single GNSS receiver).

GGA Talker IDs The NMEA specification indicates that the GGA message is GPS-specific. However, u-blox receivers support the output of a GGA message for each of the Talker IDs.

BeiDou and Galileo Only NMEA version 4.10 and later have support for these systems.

QZSS Only NMEA version 4.11 and later have support for this system.

Extended satellite numbering In order to support some GNSS (e.g. BeiDou, Galileo, QZSS) that are not supported by some or all NMEA protocol versions, an "extended" SV numbering scheme can be enabled. This uses the NMEA-defined numbers where possible, but adds other number ranges to support other GNSS. Note however that these non-standard extensions require 3-digit numbers, which may not be supported by some NMEA parsing software. For example, QZSS satellites use numbers in the range 193 to 202. See NMEA protocol configuration and Satellite identifiers.

2.5 NMEA data fields

Various data fields in NMEA messages depend on NMEA protocol configuration or require a definition for their interpretation.

2.5.1 NMEA Talker ID

One of the ways the NMEA standard differs depending on the GNSS is by using a two-letter message identifier, the "Talker ID". The specific Talker ID used by a u-blox receiver will depend on the product and its configuration. The table below shows the Talker ID that will be used for various GNSS configurations by default.

GNSS	Talker ID	Comments
GPS, SBAS	GP	NMEA 2.3+
GLONASS	GL	NMEA 2.3+
Galileo	GA	NMEA 4.10+
BeiDou	GB	NMEA 4.10+ (official NMEA only since 4.11)
NavIC	GI	NMEA 4.11+
QZSS	GQ	NMEA 4.11+ (GP for NMEA 2.3 - 4.10)
Any combination of GNSS	GN	

2.5.2 NMEA extra fields

The following extra fields are available in NMEA 4.10 and later.

Message	Extra fields
NMEA-Standard-GBS	`systemId` and `signalId`
NMEA-Standard-GNS	`navStatus`
NMEA-Standard-GRS	`systemId` and `signalId`
NMEA-Standard-GSA	`systemId`
NMEA-Standard-GSV	`signalId`
NMEA-Standard-RMC	`navStatus`

2.5.3 NMEA latitude and longitude format

According to the NMEA standard, latitude and longitude are output in the format *degrees, minutes and (decimal) fractions of minutes*. To convert to *degrees and fractions of degrees*, or *degrees, minutes, seconds and fractions of seconds*, the *minutes* and *fractional minutes* parts need to be converted. For example:

Format	Latitude	Longitude
Receiver output	`$GNRMC,014230.00,A,`**`4722.80340,N,00831.68218,E`**`,0.000,,120477,,,A,V*14`	
(d)ddmm.mmmmm	4722.80340 North	00831.68218 East
Degrees and minutes	47 degrees, 22.80340 minutes	8 degrees, 31.68218 minutes
Degrees	47.38005667 degrees	8.52803633 degrees
Degrees, minutes and seconds	47 degrees, 22 minutes, 48.2040 seconds	8 degrees, 31 minutes, 40.9308 seconds

2.5.4 NMEA GNSS, satellite and signal numbering

See GNSS, satellite and signal identifiers for details on how GNSS, satellites and signals are numbered in the NMEA protocol.

NMEA defines satellite numbering systems for some, but not all GNSS. The exact behavior depends on the configured NMEA protocol version and ("extended" or "strict") mode. See NMEA protocol configuration for details.

2.5.5 NMEA position fix flags

This section shows how u-blox positioning receivers implement the NMEA protocol and the conditions determining how flags are set.

The following flags are used in NMEA 4.10 and later.

NMEA Message	GLL, RMC	GGA	GLL, VTG	RMC, GNS
Field	**status**[5]	**quality**[6]	**posMode**[7]	**posMode**[7]
No position fix (at power-up, after losing satellite lock)	V	0	N	N
GNSS fix, but user limits exceeded	V	0	N	N
Dead reckoning fix, but user limits exceeded	V	6	E	E
Dead reckoning fix	A	6	E	E
RTK float	A	5	D	F
RTK fixed	A	4	D	R

5 Possible *status* values: V = data invalid, A = data valid

6 Possible values for *quality*: 0 = No fix, 1 = autonomous GNSS fix, 2 = differential GNSS fix, 4 = RTK fixed, 5 = RTK float, 6 = estimated/dead reckoning fix

7 Possible values for *posMode*: N = No fix, E = estimated/dead reckoning fix, A = autonomous GNSS fix, D = differential GNSS fix, F = RTK float, R = RTK fixed

NMEA Message	GLL, RMC	GGA	GLL, VTG	RMC, GNS
Field	**status**[5]	**quality**[6]	**posMode**[7]	**posMode**[7]
2D GNSS fix	A	1 / 2	A / D	A / D
3D GNSS fix	A	1 / 2	A / D	A / D
Combined GNSS/dead reckoning fix	A	1 / 2	A / D	A / D

In high precision GNSS (HPG) products it is recommended to select NMEA version 4.10 or above. Earlier versions do not support the float RTK (F) and real time kinematic (R) mode indicator flags in all messages.

The following flags are used in NMEA 2.3 - 4.0.

NMEA Message	GLL, RMC	GGA	GSA	GLL, VTG, RMC, GNS
Field	**status**[8]	**quality**[9]	**navMode**[10]	**posMode**[11]
No position fix (at power-up, after losing satellite lock)	V	0	1	N
GNSS fix, but user limits exceeded	V	0	1	N
Dead reckoning fix, but user limits exceeded	V	6	2	E
Dead reckoning fix	A	6	2	E
2D GNSS fix	A	1 / 2	2	A / D
3D GNSS fix	A	1 / 2	3	A / D
Combined GNSS/dead reckoning fix	A	1 / 2	3	A / D

The flags in NMEA 2.1 and earlier are the same as NMEA 2.3 but with the following differences:

- The *posMode* field is not output for GLL, RMC and VTG messages (each message has one field less).
- The GGA *quality* field is set to 1 (instead of 6) for both types of dead reckoning fix.

2.5.6 NMEA output of invalid or unknown data

By default the receiver will not output invalid data. In such cases, it will output empty fields. See NMEA protocol configuration for options to adjust this behavior.

A valid position fix is reported as follows:

```
$GPGLL,4717.11634,N,00833.91297,E,124923.00,A,A*6E
```

An invalid position fix (but valid time) is reported as follows:

```
$GPGLL,,,,,124924.00,V,N*42
```

If the time is unknown (e.g. during a cold start):

```
$GPGLL,,,,,,V,N*64
```

☞ Unlike the NMEA standard behavior to invalid data, dead reckoning products always report a position. It is marked as invalid (V) when the user limits are exceeded or valid (A) if the user limits are met.

8 Possible values for *status*: V = data invalid, A = data valid

9 Possible values for *quality*: 0 = no fix, 1 = autonomous GNSS fix, 2 = differential GNSS fix, 4 = RTK fixed, 5 = RTK float, 6 = estimated/dead reckoning fix

10 Possible values for *navMode*: 1 = No fix, 2 = 2D fix, 3 = 3D fix

11 Possible values for *posMode*: N = No fix, E = estimated/dead reckoning fix, A = autonomous GNSS fix, D = differential GNSS fix

2.6 NMEA messages overview

Message	*Class/ID*	*Description (Type)*
NMEA-Standard – Standard NMEA messages		
NMEA-Standard-DTM	0xf0 0x0a	• Datum reference (Output)
NMEA-Standard-GAQ	0xf0 0x45	• Poll a standard message (Talker ID GA) (Poll request)
NMEA-Standard-GBQ	0xf0 0x44	• Poll a standard message (Talker ID GB) (Poll request)
NMEA-Standard-GBS	0xf0 0x09	• GNSS satellite fault detection (Output)
NMEA-Standard-GGA	0xf0 0x00	• Global positioning system fix data (Output)
NMEA-Standard-GLL	0xf0 0x01	• Latitude and longitude, with time of position fix and status (Output)
NMEA-Standard-GLQ	0xf0 0x43	• Poll a standard message (Talker ID GL) (Poll request)
NMEA-Standard-GNQ	0xf0 0x42	• Poll a standard message (Talker ID GN) (Poll request)
NMEA-Standard-GNS	0xf0 0x0d	• GNSS fix data (Output)
NMEA-Standard-GPQ	0xf0 0x40	• Poll a standard message (Talker ID GP) (Poll request)
NMEA-Standard-GQQ	0xf0 0x47	• Poll a standard message (Talker ID GQ) (Poll request)
NMEA-Standard-GRS	0xf0 0x06	• GNSS range residuals (Output)
NMEA-Standard-GSA	0xf0 0x02	• GNSS DOP and active satellites (Output)
NMEA-Standard-GST	0xf0 0x07	• GNSS pseudorange error statistics (Output)
NMEA-Standard-GSV	0xf0 0x03	• GNSS satellites in view (Output)
NMEA-Standard-RLM	0xf0 0x0b	• Return link message (RLM) (Output)
NMEA-Standard-RMC	0xf0 0x04	• Recommended minimum data (Output)
NMEA-Standard-TXT	0xf0 0x41	• Text transmission (Output)
NMEA-Standard-VLW	0xf0 0x0f	• Dual ground/water distance (Output)
NMEA-Standard-VTG	0xf0 0x05	• Course over ground and ground speed (Output)
NMEA-Standard-ZDA	0xf0 0x08	• Time and date (Output)
NMEA-NAV2 – Secondary output NMEA messages		
NMEA-NAV2-GGA	0xf7 0x00	• Global positioning system fix data (Output)
NMEA-NAV2-GLL	0xf7 0x01	• Latitude and longitude, with time of position fix and status. (Output)
NMEA-NAV2-GNS	0xf7 0x0d	• GNSS fix data (Output)
NMEA-NAV2-GSA	0xf7 0x02	• GNSS DOP and active satellites (Output)
NMEA-NAV2-RMC	0xf7 0x04	• Recommended minimum data (Output)
NMEA-NAV2-VTG	0xf7 0x05	• Course over ground and ground speed (Output)
NMEA-NAV2-ZDA	0xf7 0x08	• Time and date (Output)
NMEA-PUBX – u-blox proprietary NMEA messages		
NMEA-PUBX-CONFIG	0xf1 0x41	• Set protocols and baud rate (Set)
NMEA-PUBX-POSITION	0xf1 0x00	• Poll a PUBX,00 message (Poll request) • Lat/Long position data (Output)
NMEA-PUBX-RATE	0xf1 0x40	• Set NMEA message output rate (Set)
NMEA-PUBX-SVSTATUS	0xf1 0x03	• Poll a PUBX,03 message (Poll request) • Satellite status (Output)
NMEA-PUBX-TIME	0xf1 0x04	• Poll a PUBX,04 message (Poll request) • Time of day and clock information (Output)

2.7 Standard messages

Standard NMEA messages as defined by the NMEA 0183 standard. See NMEA protocol for details.

2.7.1 DTM

2.7.1.1 Datum reference

Message	**NMEA-Standard-DTM** **Datum reference**
Type	Output
Comment	This message gives the difference between the current datum and the reference datum. The current datum is set to WGS84 by default. The reference datum cannot be changed and is always set to WGS84.
Information	*Class/ID:* 0xf0 0x0a *Number of fields:* 11
Structure	`$xxDTM,datum,subDatum,lat,NS,lon,EW,alt,refDatum*cs\r\n`
Examples	`$GPDTM,W84,,0.0,N,0.0,E,0.0,W84*6F\r\n` `$GPDTM,999,,0.08,N,0.07,E,-47.7,W84*1C\r\n`

Payload:

Field	*Name*	*Format*	*Unit*	*Example*	*Description*
0	`xxDTM`	string	-	$GPDTM	DTM Message ID (xx = current Talker ID, see NMEA Talker IDs table)
1	`datum`	string	-	W84	Local datum code: W84 = WGS84, P90 = PZ90, 999 = user-defined
2	`subDatum`	string	-	-	A null field (or a string describing the currently selected datum for protocol versions less than 14.00)
3	`lat`	numeric	min	0.08	Offset in Latitude
4	`NS`	character	-	S	North/South indicator
5	`lon`	numeric	min	0.07	Offset in Longitude
6	`EW`	character	-	E	East/West indicator
7	`alt`	numeric	m	-2.8	Offset in altitude
8	`refDatum`	string	-	W84	Reference datum code: W84 (WGS 84, fixed field)
9	`cs`	hexadecimal	-	*67	Checksum
10	`CRLF`	character	-	-	Carriage return and line feed

2.7.2 GAQ

2.7.2.1 Poll a standard message (Talker ID GA)

Message	**NMEA-Standard-GAQ** **Poll a standard message (Talker ID GA)**
Type	Poll request
Comment	Polls a standard NMEA message if the current Talker ID is GA.
Information	*Class/ID:* 0xf0 0x45 *Number of fields:* 4
Structure	`$xxGAQ,msgId*cs\r\n`
Example	`$EIGAQ,RMC*2B\r\n`

Payload:

Field	*Name*	*Format*	*Unit*	*Example*	*Description*

0	`xxGAQ`	string	-	$EIGAQ	GAQ Message ID (xx = Talker ID of the device requesting the poll)
1	`msgId`	string	-	RMC	Message ID of the message to be polled
2	`cs`	hexadecimal	-	*2B	Checksum
3	`CRLF`	character	-	-	Carriage return and line feed

2.7.3 GBQ

2.7.3.1 Poll a standard message (Talker ID GB)

Message	**NMEA-Standard-GBQ** **Poll a standard message (Talker ID GB)**
Type	Poll request
Comment	Polls a standard NMEA message if the current Talker ID is GB
Information	*Class/ID:* 0xf0 0x44 *Number of fields:* 4
Structure	`$xxGBQ,msgId*cs\r\n`
Example	`$EIGBQ,RMC*28\r\n`

Payload:

Field	*Name*	*Format*	*Unit*	*Example*	*Description*
0	`xxGBQ`	string	-	$EIGBQ	GBQ Message ID (xx = Talker ID of the device requesting the poll)
1	`msgId`	string	-	RMC	Message ID of the message to be polled
2	`cs`	hexadecimal	-	*28	Checksum
3	`CRLF`	character	-	-	Carriage return and line feed

2.7.4 GBS

2.7.4.1 GNSS satellite fault detection

Message	**NMEA-Standard-GBS** **GNSS satellite fault detection**
Type	Output
Comment	This message outputs the results of the Receiver Autonomous Integrity Monitoring Algorithm (RAIM). • The fields **errLat**, **errLon** and **errAlt** output the standard deviation of the position calculation, using all satellites that pass the RAIM test successfully. • The fields **errLat**, **errLon** and **errAlt** are only output if the RAIM process passed successfully (i.e. no or successful edits happened). These fields are never output if 4 or fewer satellites are used for the navigation calculation (because, in such cases, integrity cannot be determined by the receiver autonomously). • The fields **prob**, **bias** and **stdev** are only output if at least one satellite failed in the RAIM test. If more than one satellites fail the RAIM test, only the information for the worst satellite is output in this message.
Information	*Class/ID:* 0xf0 0x09 *Number of fields:* 13
Structure	`$xxGBS,time,errLat,errLon,errAlt,svid,prob,bias,stddev,systemId,signalId*cs\r\n`
Examples	`$GPGBS,235503.00,1.6,1.4,3.2,,,,,,*40\r\n` `$GPGBS,235458.00,1.4,1.3,3.1,03,,-21.4,3.8,1,0*5B\r\n`

Payload:

Field	*Name*	*Format*	*Unit*	*Example*	*Description*
0	`xxGBS`	string	-	$GPGBS	GBS Message ID (xx = current Talker ID, see NMEA Talker IDs table)

1	`time`	hhmmss.ss	-	235503.00	UTC time to which this RAIM sentence belongs. See section UTC representation in the integration manual for details.
2	`errLat`	numeric	m	1.6	Expected error in latitude
3	`errLon`	numeric	m	1.4	Expected error in longitude
4	`errAlt`	numeric	m	3.2	Expected error in altitude
5	`svid`	numeric	-	03	Satellite ID of most likely failed satellite
6	`prob`	numeric	-	-	Probability of missed detection: null (not supported, fixed field)
7	`bias`	numeric	m	-21.4	Estimated bias of most likely failed satellite (a priori residual)
8	`stddev`	numeric	m	3.8	Standard deviation of estimated bias
9	`systemId`	hexadecimal	-	1	NMEA-defined GNSS system ID, see Signal Identifiers table (only available in NMEA 4.10 and later)
10	`signalId`	hexadecimal	-	-	NMEA-defined GNSS signal ID, see Signal Identifiers table (only available in NMEA 4.10 and later)
11	`cs`	hexadecimal	-	*5B	Checksum
12	`CRLF`	character	-	-	Carriage return and line feed

2.7.5 GGA

2.7.5.1 Global positioning system fix data

Message	**NMEA-Standard-GGA** **Global positioning system fix data**
Type	Output
Comment	Time and position, together with GPS fixing-related data (number of satellites in use, and the resulting HDOP, age of differential data if in use, etc.). ☞ The output of this message is dependent on the currently selected datum (default: WGS84). The NMEA specification indicates that the GGA message is GPS-specific. However, when the receiver is configured for multi-GNSS, the GGA message contents will be generated from the multi-GNSS solution. For multi-GNSS use, it is recommended that the NMEA-GNS message is used instead.
Information	*Class/ID:* 0xf0 0x00 *Number of fields:* 17
Structure	`$xxGGA,time,lat,NS,lon,EW,quality,numSV,HDOP,alt,altUnit,sep,sepUnit,diffAge,diffSta ↲ tion*cs\r\n`
Example	`$GPGGA,092725.00,4717.11399,N,00833.91590,E,1,08,1.01,499.6,M,48.0,M,,*5B\r\n`

Payload:

Field	*Name*	*Format*	*Unit*	*Example*	*Description*
0	`xxGGA`	string	-	$GPGGA	GGA Message ID (xx = current Talker ID, see NMEA Talker IDs table)
1	`time`	hhmmss.ss	-	092725.00	UTC time. See section UTC representation in the integration manual for details.
2	`lat`	ddmm.mmmmm	-	4717.11399	Latitude (degrees and minutes), see format description
3	`NS`	character	-	N	North/South indicator
4	`lon`	dddmm.mmmmm	-	00833.91590	Longitude (degrees and minutes), see format description
5	`EW`	character	-	E	East/West indicator
6	`quality`	digit	-	1	Quality indicator for position fix, see position fix flags description

7	numSV	numeric	-	08	Number of satellites used (range: 0-12)
8	HDOP	numeric	-	1.01	Horizontal Dilution of Precision
9	alt	numeric	m	499.6	Altitude above mean sea level
10	altUnit	character	-	M	Altitude units: M (meters, fixed field)
11	sep	numeric	m	48.0	Geoid separation: difference between ellipsoid and mean sea level
12	sepUnit	character	-	M	Geoid separation units: M (meters, fixed field)
13	diffAge	numeric	s	-	Age of differential corrections (null when DGPS is not used)
14	diffStation	numeric	-	-	ID of station providing differential corrections (null when DGPS is not used)
15	cs	hexadecimal	-	*5B	Checksum
16	CRLF	character	-	-	Carriage return and line feed

2.7.6 GLL

2.7.6.1 Latitude and longitude, with time of position fix and status

Message	**NMEA-Standard-GLL** **Latitude and longitude, with time of position fix and status**
Type	Output
Comment	☞ The output of this message is dependent on the currently selected datum (default: WGS84)
Information	*Class/ID:* 0xf0 0x01 *Number of fields:* 10
Structure	`$xxGLL,lat,NS,lon,EW,time,status,posMode*cs\r\n`
Example	`$GPGLL,4717.11364,N,00833.91565,E,092321.00,A,A*60\r\n`

Payload:

Field	*Name*	*Format*	*Unit*	*Example*	*Description*
0	xxGLL	string	-	$GPGLL	GLL Message ID (xx = current Talker ID, see NMEA Talker IDs table)
1	lat	ddmm. mmmmm	-	4717.11364	Latitude (degrees and minutes), see format description
2	NS	character	-	N	North/South indicator
3	lon	dddmm. mmmmm	-	00833.91565	Longitude (degrees and minutes), see format description
4	EW	character	-	E	East/West indicator
5	time	hhmmss.ss	-	092321.00	UTC time. See section UTC representation in the integration manual for details.
6	status	character	-	A	Data validity status, see position fix flags description
7	posMode	character	-	A	Positioning mode, see position fix flags description (only available in NMEA 2.3 and later)
8	cs	hexadecimal	-	*60	Checksum
9	CRLF	character	-	-	Carriage return and line feed

2.7.7 GLQ

2.7.7.1 Poll a standard message (Talker ID GL)

Message	**NEMA-Standard-GLQ**
	Poll a standard message (Talker ID GL)
Type	Poll request
Comment	Polls a standard NMEA message if the current Talker ID is GL
Information	*Class/ID:* 0xf0 0x43 *Number of fields:* 4
Structure	`$xxGLQ,msgId*cs\r\n`
Example	`$EIGLQ,RMC*3A\r\n`

Payload:

Field	*Name*	*Format*	*Unit*	*Example*	*Description*
0	`xxGLQ`	string	-	$EIGLQ	GLQ Message ID (xx = Talker ID of the device requesting the poll)
1	`msgId`	string	-	RMC	Message ID of the message to be polled
2	`cs`	hexadecimal	-	*3A	Checksum
3	`CRLF`	character	-	-	Carriage return and line feed

2.7.8 GNQ

2.7.8.1 Poll a standard message (Talker ID GN)

Message	**NMEA-Standard-GNQ**
	Poll a standard message (Talker ID GN)
Type	Poll request
Comment	Polls a standard NMEA message if the current Talker ID is GN
Information	*Class/ID:* 0xf0 0x42 *Number of fields:* 4
Structure	`$xxGNQ,msgId*cs\r\n`
Example	`$EIGNQ,RMC*3A\r\n`

Payload:

Field	*Name*	*Format*	*Unit*	*Example*	*Description*
0	`xxGNQ`	string	-	$EIGNQ	GNQ Message ID (xx = Talker ID of the device requesting the poll)
1	`msgId`	string	-	RMC	Message ID of the message to be polled
2	`cs`	hexadecimal	-	*3A	Checksum
3	`CRLF`	character	-	-	Carriage return and line feed

2.7.9 GNS

2.7.9.1 GNSS fix data

Message	**NMEA-Standard-GNS**
	GNSS fix data
Type	Output
Comment	Time and position, together with GNSS fixing-related data (number of satellites in use, and the resulting HDOP, age of differential data if in use, etc.). ☞ The output of this message is dependent on the currently selected datum (default: WGS84)
Information	*Class/ID:* 0xf0 0x0d *Number of fields:* 16
Structure	`$xxGNS,time,lat,NS,lon,EW,posMode,numSV,HDOP,alt,sep,diffAge,diffStation,navStatus*cs\r\n`

Examples	`$GNGNS,103600.01,5114.51176,N,00012.29380,W,ANNN,07,1.18,111.5,45.6,,,V*00\r\n`
	`$GNGNS,122310.2,3722.425671,N,12258.856215,W,DAAA,14,0.9,1005.543,6.5,,,V*0E\r\n`
	`$GPGNS,122310.2,,,,,,07,,,,5.2,23,V*02\r\n`

Payload:

Field	*Name*	*Format*	*Unit*	*Example*	*Description*
0	`xxGNS`	string	-	$GPGNS	GNS Message ID (xx = current Talker ID, see NMEA Talker IDs table)
1	`time`	hhmmss.ss	-	091547.00	UTC time. See section UTC representation in the integration manual for details.
2	`lat`	ddmm. mmmmm	-	5114.50897	Latitude (degrees and minutes), see format description
3	`NS`	character	-	N	North/South indicator
4	`lon`	dddmm. mmmmm	-	00012.28663	Longitude (degrees and minutes), see format description
5	`EW`	character	-	E	East/West indicator
6	`posMode`	character	-	AAAA	Positioning mode, see position fix flags description. First character for GPS, second character for GLONASS, third character for Galileo, fourth character for BeiDou
7	`numSV`	numeric	-	10	Number of satellites used (range: 0-99)
8	`HDOP`	numeric	-	0.83	Horizontal Dilution of Precision
9	`alt`	numeric	m	111.1	Altitude above mean sea level
10	`sep`	numeric	m	45.6	Geoid separation: difference between ellipsoid and mean sea level
11	`diffAge`	numeric	s	-	Age of differential corrections (null when DGPS is not used)
12	`diffStation`	numeric	-	-	ID of station providing differential corrections (null when DGPS is not used)
13	`navStatus`	character	-	V	Navigational status indicator: V (Equipment is not providing navigational status information, fixed field, only available in NMEA 4.10 and later)
14	`cs`	hexadecimal	-	*71	Checksum
15	`CRLF`	character	-	-	Carriage return and line feed

2.7.10 GPQ

2.7.10.1 Poll a standard message (Talker ID GP)

Message	**NMEA-Standard-GPQ** **Poll a standard message (Talker ID GP)**
Type	Poll request
Comment	Polls a standard NMEA message if the current Talker ID is GP
Information	*Class/ID:* 0xf0 0x40 *Number of fields:* 4
Structure	`$xxGPQ,msgId*cs\r\n`
Example	`$EIGPQ,RMC*3A\r\n`

Payload:

Field	*Name*	*Format*	*Unit*	*Example*	*Description*
0	`xxGPQ`	string	-	$EIGPQ	GPQ Message ID (xx = Talker ID of the device requesting the poll)
1	`msgId`	string	-	RMC	Message ID of the message to be polled

2	`cs`	hexadecimal	-	*3A	Checksum
3	`CRLF`	character	-	-	Carriage return and line feed

2.7.11 GQQ

2.7.11.1 Poll a standard message (Talker ID GQ)

Message	**NMEA-Standard-GQQ** **Poll a standard message (Talker ID GQ)**
Type	Poll request
Comment	Polls a standard NMEA message if the current Talker ID is GQ
Information	*Class/ID:* 0xf0 0x47 *Number of fields:* 4
Structure	`$xxGQQ,msgId*cs\r\n`
Example	`$EIGQQ,RMC*3A\r\n`

Payload:

Field	*Name*	*Format*	*Unit*	*Example*	*Description*
0	`xxGQQ`	string	-	$EIGQQ	GQQ Message ID (xx = Talker ID of the device requesting the poll)
1	`msgId`	string	-	RMC	Message ID of the message to be polled
2	`cs`	hexadecimal	-	*3A	Checksum
3	`CRLF`	character	-	-	Carriage return and line feed

2.7.12 GRS

2.7.12.1 GNSS range residuals

Message	**NMEA-Standard-GRS** **GNSS range residuals**
Type	Output
Comment	If less than 12 SVs are available, the remaining fields are output empty. If more than 12 SVs are used, only the residuals of the first 12 SVs are output, in order to remain consistent with the NMEA standard. **In a multi-GNSS system this message will be output multiple times, once for each GNSS.** ☞ This message relates to associated GGA and GSA messages.
Information	*Class/ID:* 0xf0 0x06 *Number of fields:* 19
Structure	`$xxGRS,time,mode{,residual},systemId,signalId*cs\r\n`
Examples	`$GNGRS,104148.00,1,2.6,2.2,-1.6,-1.1,-1.7,-1.5,5.8,1.7,,,,,,1,1*52\r\n` `$GNGRS,104148.00,1,,0.0,2.5,0.0,,2.8,,,,,,,,1,5*52\r\n`

Payload:

Field	*Name*	*Format*	*Unit*	*Example*	*Description*
0	`xxGRS`	string	-	$GPGRS	GRS Message ID (xx = current Talker ID, see NMEA Talker IDs table)
1	`time`	hhmmss.ss	-	082632.00	UTC time of associated position fix. See section UTC representation in the integration manual for details.
2	`mode`	digit	-	1	Computation method used: • 1 = Residuals were recomputed after the GGA position was computed (fixed)
Start of repeated group (12 times)					
3 + n	`residual`	numeric	m	0.54	Range residuals for SVs used in navigation. The SV order matches the order from the GSA sentence

Field	Name	Format	Unit	Example	Description
End of repeated group (12 times)					
15	`systemId`	hexadecimal	-	1	NMEA-defined GNSS system ID, see Signal Identifiers table (only available in NMEA 4.10 and later)
16	`signalId`	hexadecimal	-	-	NMEA-defined GNSS signal ID, see Signal Identifiers table (only available in NMEA 4.10 and later)
17	`cs`	hexadecimal	-	*70	Checksum
18	`CRLF`	character	-	-	Carriage return and line feed

2.7.13 GSA

2.7.13.1 GNSS DOP and active satellites

Message	**NMEA-Standard-GSA** **GNSS DOP and active satellites**
Type	Output
Comment	The GNSS receiver operating mode, satellites used for navigation, and DOP values. • If less than 12 SVs are used for navigation, the remaining fields are left empty. If more than 12 SVs are used for navigation, only the IDs of the first 12 are output. • The SV numbers (fields 'svid') are in the range of 1 to 32 for GPS satellites, and 33 to 64 for SBAS satellites (33 = SBAS PRN 120, 34 = SBAS PRN 121, and so on) **In a multi-GNSS system this message will be output multiple times, once for each GNSS.**
Information	*Class/ID:* 0xf0 0x02 *Number of fields:* 21
Structure	`$xxGSA,opMode,navMode{,svid},PDOP,HDOP,VDOP,systemId*cs\r\n`
Example	`$GPGSA,A,3,23,29,07,08,09,18,26,28,,,,,1.94,1.18,1.54,1*0D\r\n`

Payload:

Field	*Name*	*Format*	*Unit*	*Example*	*Description*
0	`xxGSA`	string	-	$GPGSA	GSA Message ID (xx = current Talker ID, see NMEA Talker IDs table)
1	`opMode`	character	-	A	Operation mode: • M = Manually set to operate in 2D or 3D mode • A = Automatically switching between 2D or 3D mode
2	`navMode`	digit	-	3	Navigation mode, see position fix flags description
Start of repeated group (12 times)					
3 + n	`svid`	numeric	-	29	Satellite number
End of repeated group (12 times)					
15	`PDOP`	numeric	-	1.94	Position dilution of precision
16	`HDOP`	numeric	-	1.18	Horizontal dilution of precision
17	`VDOP`	numeric	-	1.54	Vertical dilution of precision
18	`systemId`	hexadecimal	-	1	NMEA-defined GNSS system ID, see Signal Identifiers table (only available in NMEA 4.10 and later)
19	`cs`	hexadecimal	-	*0D	Checksum
20	`CRLF`	character	-	-	Carriage return and line feed

2.7.14 GST

2.7.14.1 GNSS pseudorange error statistics

Message	**NEMA-Standard-GST** **GNSS pseudorange error statistics**
Type	Output
Comment	This message reports statistical information on the quality of the position solution.
Information	*Class/ID:* 0xf0 0x07 *Number of fields:* 11
Structure	`$xxGST,time,rangeRms,stdMajor,stdMinor,orient,stdLat,stdLong,stdAlt*cs\r\n`
Example	`$GPGST,082356.00,1.8,,,,1.7,1.3,2.2*7E\r\n`

Payload:

Field	*Name*	*Format*	*Unit*	*Example*	*Description*
0	`xxGST`	string	-	$GPGST	GST Message ID (xx = current Talker ID, see NMEA Talker IDs table)
1	`time`	hhmmss.ss	-	082356.00	UTC time of associated position fix. See section UTC representation in the integration manual for details.
2	`rangeRms`	numeric	m	1.8	RMS value of the standard deviation of the ranges
3	`stdMajor`	numeric	m	-	Standard deviation of semi-major axis
4	`stdMinor`	numeric	m	-	Standard deviation of semi-minor axis
5	`orient`	numeric	deg	-	Orientation of semi-major axis
6	`stdLat`	numeric	m	1.7	Standard deviation of latitude error
7	`stdLong`	numeric	m	1.3	Standard deviation of longitude error
8	`stdAlt`	numeric	m	2.2	Standard deviation of altitude error
9	`cs`	hexadecimal	-	*7E	Checksum
10	`CRLF`	character	-	-	Carriage return and line feed

2.7.15 GSV

2.7.15.1 GNSS satellites in view

Message	**NMEA-Standard-GSV** **GNSS satellites in view**
Type	Output
Comment	The number of satellites in view, together with each SV ID, elevation azimuth, and signal strength (C/No) value. Only four satellite details are transmitted in one message. **In a multi-GNSS system sets of GSV messages will be output multiple times, one set for each GNSS.**
Information	*Class/ID:* 0xf0 0x03 *Number of fields:* 7 + [1..4]·4
Structure	`$xxGSV,numMsg,msgNum,numSV{,svid,elv,az,cno},signalId*cs\r\n`

Examples	`$GPGSV,3,1,09,09,,,17,10,,,40,12,,,49,13,,,35,1*6F\r\n`
	`$GPGSV,3,2,09,15,,,44,17,,,45,19,,,44,24,,,50,1*64\r\n`
	`$GPGSV,3,3,09,25,,,40,1*6E\r\n`
	`$GPGSV,1,1,03,12,,,42,24,,,47,32,,,37,5*66\r\n`
	`$GAGSV,1,1,00,2*76\r\n`

Payload:

Field	*Name*	*Format*	*Unit*	*Example*	*Description*
0	`xxGSV`	string	-	$GPGSV	GSV Message ID (xx = GSV Talker ID, see NMEA Talker IDs table). Talker ID GN shall not be used.
1	`numMsg`	digit	-	3	Number of messages, total number of GSV messages being output (range: 1-9)
2	`msgNum`	digit	-	1	Number of this message (range: 1-numMsg)

3	numSV	numeric	-	10	Number of known satellites in view regarding both the talker ID and the signalId
Start of repeated group (1...4 times)					
4 + n·4	svid	numeric	-	23	Satellite ID
5 + n·4	elv	numeric	deg	38	Elevation (<= 90)
6 + n·4	az	numeric	deg	230	Azimuth (range: 0-359)
7 + n·4	cno	numeric	dBHz	44	Signal strength (C/N0, range: 0-99), null when not tracking
End of repeated group (1...4 times)					
4 + N·4	signalId	hexadecimal	-	-	NMEA-defined GNSS signal ID, see Signal Identifiers table (only available in NMEA 4.10 and later)
5 + N·4	cs	hexadecimal	-	*7F	Checksum
6 + N·4	CRLF	character	-	-	Carriage return and line feed

2.7.16 RLM

2.7.16.1 Return link message (RLM)

Message	**NMEA-Standard-RLM** **Return link message (RLM)**
Type	Output
Comment	The RLM sentence is used to transfer a Return link message from a Cospas-Sarsat recognized Return link service provider (RLSP). The RLM sentence supports communications to an emitting beacon once a distress alert has been detected, located and confirmed. The communications may include acknowledgement of the alert to the emitting beacon as well as optional text messages, and may also include remote beacon configuration and testing.
Information	*Class/ID:* 0xf0 0x0b *Number of fields:* 7
Structure	`$xxRLM,beacon,time,code,body*cs\r\n`
Examples	`$GARLM,00000078A9FBAD5,083559.00,3,C45B*57\r\n` `$GARLM,F7129D41BC6A78C,034433.02,3,B63CA732AFD419D2*57\r\n`

Payload:

Field	*Name*	*Format*	*Unit*	*Example*	*Description*
0	xxRLM	string	-	$GARLM	RLM message ID (xx = current Talker ID, see NMEA Talker IDs table)
1	beacon	hexadecimal	-	00000078A 9FBAD5	Beacon ID, identifies beacon intended to receive this message (fixed length 15 hexadecimal character field)
2	time	hhmmss.ss	-	083559.00	Time of reception field to indicate RLM timestamp in UTC. See section UTC representation in the integration manual for details.
3	code	character	-	3	Message code field to identify type of RLM Message Service: • 0 = Reserved for future RLM services • 1 = Acknowledgement service RLM • 2 = Command service RLM • 3 = Message service RLM • 4-E = Reserved for future RLM services • F = Test service RLM (currently used only by the Galileo program)
4	body	hexadecimal	-	C45B	Message body encapsulates the data parameters provided by the RLSP into hexadecimal format.
5	cs	hexadecimal	-	*57	Checksum

6	CRLF	character	-	-	Carriage return and line feed

2.7.17 RMC

2.7.17.1 Recommended minimum data

Message	**NMEA-Standard-RMC** **Recommended minimum data**
Type	Output
Comment	The recommended minimum sentence defined by NMEA for GNSS system data. ☞ The output of this message is dependent on the currently selected datum (default: WGS84)
Information	*Class/ID:* 0xf0 0x04 *Number of fields:* 16
Structure	`$xxRMC,time,status,lat,NS,lon,EW,spd,cog,date,mv,mvEW,posMode,navStatus*cs\r\n`
Example	`$GPRMC,083559.00,A,4717.11437,N,00833.91522,E,0.004,77.52,091202,,,A,V*57\r\n`

Payload:

Field	*Name*	*Format*	*Unit*	*Example*	*Description*
0	xxRMC	string	-	$GPRMC	RMC Message ID (xx = current Talker ID, see NMEA Talker IDs table)
1	time	hhmmss.ss	-	083559.00	UTC time. See section UTC representation in the integration manual for details.
2	status	character	-	A	Data validity status, see position fix flags description
3	lat	ddmm.mmmmm	-	4717.11437	Latitude (degrees and minutes), see format description
4	NS	character	-	N	North/South indicator
5	lon	dddmm.mmmmm	-	00833.91522	Longitude (degrees and minutes), see format description
6	EW	character	-	E	East/West indicator
7	spd	numeric	knots	0.004	Speed over ground
8	cog	numeric	deg	77.52	Course over ground
9	date	ddmmyy	-	091202	Date in day, month, year format. See section UTC representation in the integration manual for details.
10	mv	numeric	deg	-	Magnetic variation value
11	mvEW	character	-	-	Magnetic variation E/W indicator
12	posMode	character	-	A	Mode Indicator, see position fix flags description (only available in NMEA 2.3 and later)
13	navStatus	character	-	V	Navigational status indicator: V (Equipment is not providing navigational status information, fixed field, only available in NMEA 4.10 and later)
14	cs	hexadecimal	-	*57	Checksum
15	CRLF	character	-	-	Carriage return and line feed

2.7.18 TXT

2.7.18.1 Text transmission

Message	**NMEA-Standard-TXT** **Text transmission**
Type	Output

Comment	This message outputs various information on the receiver, such as power-up screen, software version etc. This message can be configured using the CFG-INFMSG configuration group.
Information	*Class/ID:* 0xf0 0x41 *Number of fields:* 7
Structure	`$xxTXT,numMsg,msgNum,msgType,text*cs\r\n`
Examples	`$GPTXT,01,01,02,u-blox ag - www.u-blox.com*50\r\n` `$GPTXT,01,01,02,ANTARIS ATR0620 HW 00000040*67\r\n`

Payload:

Field	*Name*	*Format*	*Unit*	*Example*	*Description*
0	`xxTXT`	string	-	$GPTXT	TXT Message ID (xx = current Talker ID, see NMEA Talker IDs table)
1	`numMsg`	numeric	-	01	Total number of messages in this transmission (range: 1-99)
2	`msgNum`	numeric	-	01	Message number in this transmission (range: 1-numMsg)
3	`msgType`	numeric	-	02	Text identifier (u-blox receivers specify the type of the message with this number): • 00 = Error • 01 = Warning • 02 = Notice • 07 = User
4	`text`	string	-	www.u-blox.com	Any ASCII text
5	`cs`	hexadecimal	-	*67	Checksum
6	`CRLF`	character	-	-	Carriage return and line feed

2.7.19 VLW

2.7.19.1 Dual ground/water distance

Message	**NMEA-Standard-VLW** **Dual ground/water distance**
Type	Output
Comment	The distance traveled, relative to the water and over the ground. This message relates to the odometer feature detailed in the integration manual.
Information	*Class/ID:* 0xf0 0x0f *Number of fields:* 11
Structure	`$xxVLW,twd,twdUnit,wd,wdUnit,tgd,tgdUnit,gd,gdUnit*cs\r\n`
Example	`$GPVLW,,N,,N,15.8,N,1.2,N*06\r\n`

Payload:

Field	*Name*	*Format*	*Unit*	*Example*	*Description*
0	`xxVLW`	string	-	$GPVLW	VLW Message ID (xx = current Talker ID, see NMEA Talker IDs table)
1	`twd`	numeric	nmi	-	Total cumulative water distance: null (fixed field)
2	`twdUnit`	character	-	N	Total cumulative water distance units: N (nautical miles, fixed field)
3	`wd`	numeric	nmi	-	Water distance since reset: null (fixed field)
4	`wdUnit`	character	-	N	Water distance since reset units: N (nautical miles, fixed field)
5	`tgd`	numeric	nmi	15.8	Total cumulative ground distance (only available in NMEA 4.00 and later)

6	tgdUnit	character	-	N	Total cumulative ground distance units: N (nautical miles, fixed field, only available in NMEA 4.00 and later)
7	gd	numeric	nmi	1.2	Ground distance since reset (only available in NMEA 4.00 and later)
8	gdUnit	character	-	N	Ground distance since reset units: N (nautical miles, fixed field, only available in NMEA 4.00 and later)
9	cs	hexadecimal	-	*06	Checksum
10	CRLF	character	-	-	Carriage return and line feed

2.7.20 VTG

2.7.20.1 Course over ground and ground speed

Message	**NMEA-Standard-VTG** **Course over ground and ground speed**
Type	Output
Comment	Velocity is given as course over ground (COG) and speed over ground (SOG).
Information	*Class/ID:* 0xf0 0x05 *Number of fields:* 12
Structure	`$xxVTG,cogt,cogtUnit,cogm,cogmUnit,sogn,sognUnit,sogk,sogkUnit,posMode*cs\r\n`
Example	`$GPVTG,77.52,T,,M,0.004,N,0.008,K,A*06\r\n`

Payload:

Field	*Name*	*Format*	*Unit*	*Example*	*Description*
0	xxVTG	string	-	$GPVTG	VTG Message ID (xx = current Talker ID, see NMEA Talker IDs table)
1	cogt	numeric	degrees	77.52	Course over ground (true)
2	cogtUnit	character	-	T	Course over ground units: T (degrees true, fixed field)
3	cogm	numeric	degrees	-	Course over ground (magnetic)
4	cogmUnit	character	-	M	Course over ground units: M (degrees magnetic, fixed field)
5	sogn	numeric	knots	0.004	Speed over ground
6	sognUnit	character	-	N	Speed over ground units: N (knots, fixed field)
7	sogk	numeric	km/h	0.008	Speed over ground
8	sogkUnit	character	-	K	Speed over ground units: K (kilometers per hour, fixed field)
9	posMode	character	-	A	Mode indicator, see position fix flags description (only available in NMEA 2.3 and later)
10	cs	hexadecimal	-	*06	Checksum
11	CRLF	character	-	-	Carriage return and line feed

2.7.21 ZDA

2.7.21.1 Time and date

Message	**NMEA-Standard-ZDA** **Time and date**
Type	Output
Comment	UTC, day, month, year and local time zone.
Information	*Class/ID:* 0xf0 0x08 *Number of fields:* 9

Structure	`$xxZDA,time,day,month,year,ltzh,ltzn*cs\r\n`
Example	`$GPZDA,082710.00,16,09,2002,00,00*64\r\n`

Payload:

Field	*Name*	*Format*	*Unit*	*Example*	*Description*
0	`xxZDA`	string	-	$GPZDA	ZDA Message ID (xx = current Talker ID, see NMEA Talker IDs table)
1	`time`	hhmmss.ss	-	082710.00	UTC Time. See section UTC representation in the integration manual for details.
2	`day`	dd	day	16	UTC day (range: 1-31)
3	`month`	mm	month	09	UTC month (range: 1-12)
4	`year`	yyyy	year	2002	UTC year
5	`ltzh`	xx	-	00	Local time zone hours (fixed field, always 00)
6	`ltzn`	zz	-	00	Local time zone minutes (fixed field, always 00)
7	`cs`	hexadecimal	-	*64	Checksum
8	`CRLF`	character	-	-	Carriage return and line feed

2.8 Secondary output messages

Secondary output NMEA messages. These are NMEA messages prepended with an NMEA TAG block as defined by the NMEA 0183 standard. See NMEA protocol for details.

2.8.1 GGA

2.8.1.1 Global positioning system fix data

Message	**NMEA-NAV2-GGA** **Global positioning system fix data**
Type	Output
Comment	Time and position, together with GPS fixing-related data (number of satellites in use, and the resulting HDOP, age of differential data if in use, etc.). To identify the navigation data source for NMEA Secondary filter output, the alphanumeric string source-identification (s:) parameter is used in a TAG Block, in respect to NMEA 0183 Standard. ☞ The output of this message is dependent on the currently selected datum (default: WGS84). The NMEA specification indicates that the GGA message is GPS-specific. However, when the receiver is configured for multi-GNSS, the GGA message contents will be generated from the multi-GNSS solution. For multi-GNSS use, it is recommended that the NMEA-GNS message is used instead.
Information	*Class/ID:* 0xf7 0x00 *Number of fields:* 21
Structure	`\s:1*78\$xxGGA,time,lat,NS,lon,EW,quality,numSV,HDOP,alt,altUnit,sep,sepUnit,diffAge ↲ ,diffStation*cs\r\n`
Example	`\s:1*78\$GPGGA,092725.00,4717.11399,N,00833.91590,E,1,08,1.01,499.6,M,48.0,M,,*5B\r\ ↲ n`

Payload:

Field	*Name*	*Format*	*Unit*	*Example*	*Description*
0	`tagStart`	string	-	\s:	NMEA TAG block start and parameter
1	`source`	numeric	-	1	NMEA TAG block source value (1 for secondary output messages)
2	`tagCs`	hexadecimal	-	*78	NMEA TAG checksum
3	`tagEnd`	string	-	\	NMEA TAG block end character
4	`xxGGA`	string	-	$GPGGA	GGA Message ID (xx = current Talker ID, see NMEA Talker IDs table)

5	time	hhmmss.ss	-	092725.00	UTC time. See section UTC representation in the integration manual for details.
6	lat	ddmm. mmmmm	-	4717.11399	Latitude (degrees and minutes), see format description
7	NS	character	-	N	North/South indicator
8	lon	dddmm. mmmmm	-	00833.91590	Longitude (degrees and minutes), see format description
9	EW	character	-	E	East/West indicator
10	quality	digit	-	1	Quality indicator for position fix, see position fix flags description
11	numSV	numeric	-	08	Number of satellites used (range: 0-12)
12	HDOP	numeric	-	1.01	Horizontal Dilution of Precision
13	alt	numeric	m	499.6	Altitude above mean sea level
14	altUnit	character	-	M	Altitude units: M (meters, fixed field)
15	sep	numeric	m	48.0	Geoid separation: difference between ellipsoid and mean sea level
16	sepUnit	character	-	M	Geoid separation units: M (meters, fixed field)
17	diffAge	numeric	s	-	Age of differential corrections (null when DGPS is not used)
18	diffStation	numeric	-	-	ID of station providing differential corrections (null when DGPS is not used)
19	cs	hexadecimal	-	*5B	Checksum
20	CRLF	character	-	-	Carriage return and line feed

2.8.2 GLL

2.8.2.1 Latitude and longitude, with time of position fix and status.

Message	**NMEA-NAV2-GLL** **Latitude and longitude, with time of position fix and status.**
Type	Output
Comment	Geographic Position - Latitude/Longitude. To identify the navigation data source for NMEA Secondary filter output, the alphanumeric string source-identification (s:) parameter is used in a TAG Block, in respect to NMEA 0183 Standard. ☞ The output of this message is dependent on the currently selected datum (default: WGS84)
Information	*Class/ID:* 0xf7 0x01 *Number of fields:* 14
Structure	`\s:1*78\$xxGLL,lat,NS,lon,EW,time,status,posMode*cs\r\n`
Example	`\s:1*78\$GPGLL,4717.11364,N,00833.91565,E,092321.00,A,A*60\r\n`

Payload:

Field	*Name*	*Format*	*Unit*	*Example*	*Description*
0	tagStart	string	-	\s:	NMEA TAG block start and parameter
1	source	numeric	-	1	NMEA TAG block source value (1 for secondary output messages)
2	tagCs	hexadecimal	-	*78	NMEA TAG checksum
3	tagEnd	string	-	\	NMEA TAG block end character
4	xxGLL	string	-	$GPGLL	GLL Message ID (xx = current Talker ID, see NMEA Talker IDs table)

5	lat	ddmm. mmmmm	-	4717.11364	Latitude (degrees and minutes), see format description
6	NS	character	-	N	North/South indicator
7	lon	dddmm. mmmmm	-	00833.91565	Longitude (degrees and minutes), see format description
8	EW	character	-	E	East/West indicator
9	time	hhmmss.ss	-	092321.00	UTC time. See section UTC representation in the integration manual for details.
10	status	character	-	A	Data validity status, see position fix flags description
11	posMode	character	-	A	Positioning mode, see position fix flags description (only available in NMEA 2.3 and later)
12	cs	hexadecimal	-	*60	Checksum
13	CRLF	character	-	-	Carriage return and line feed

2.8.3 GNS

2.8.3.1 GNSS fix data

Message	**NMEA-NAV2-GNS** **GNSS fix data**
Type	Output
Comment	Time and position, together with GNSS fixing-related data (number of satellites in use, and the resulting HDOP, age of differential data if in use, etc.). To identify the navigation data source for NMEA Secondary filter output, the alphanumeric string source-identification (s:) parameter is used in a TAG Block, in respect to NMEA 0183 Standard. ☞ The output of this message is dependent on the currently selected datum (default: WGS84)
Information	*Class/ID:* 0xf7 0x0d *Number of fields:* 20
Structure	`\s:1*78\$xxGNS,time,lat,NS,lon,EW,posMode,numSV,HDOP,alt,sep,diffAge,diffStation,navStatus*cs\r\n`

Examples	`\s:1*78\$GNGNS,103600.01,5114.51176,N,00012.29380,W,ANNN,07,1.18,111.5,45.6,,,V*00\r\n`
	`\s:1*78\$GNGNS,122310.2,3722.425671,N,12258.856215,W,DAAA,14,0.9,1005.543,6.5,,,V*0E\r\n`
	`\s:1*78\$GPGNS,122310.2,,,,,,07,,,,5.2,23,V*02\r\n`

Payload:

Field	*Name*	*Format*	*Unit*	*Example*	*Description*
0	tagStart	string	-	\s:	NMEA TAG block start and parameter
1	source	numeric	-	1	NMEA TAG block source value (1 for secondary output messages)
2	tagCs	hexadecimal	-	*78	NMEA TAG checksum
3	tagEnd	string	-	\	NMEA TAG block end character
4	xxGNS	string	-	$GPGNS	GNS Message ID (xx = current Talker ID, see NMEA Talker IDs table)
5	time	hhmmss.ss	-	091547.00	UTC time. See section UTC representation in the integration manual for details.
6	lat	ddmm. mmmmm	-	5114.50897	Latitude (degrees and minutes), see format description
7	NS	character	-	N	North/South indicator
8	lon	dddmm. mmmmm	-	00012.28663	Longitude (degrees and minutes), see format description
9	EW	character	-	E	East/West indicator

10	`posMode`	character	-	AAAA	Positioning mode, see position fix flags description. First character for GPS, second character for GLONASS, third character for Galileo, fourth character for BeiDou
11	`numSV`	numeric	-	10	Number of satellites used (range: 0-99)
12	`HDOP`	numeric	-	0.83	Horizontal Dilution of Precision
13	`alt`	numeric	m	111.1	Altitude above mean sea level
14	`sep`	numeric	m	45.6	Geoid separation: difference between ellipsoid and mean sea level
15	`diffAge`	numeric	s	-	Age of differential corrections (null when DGPS is not used)
16	`diffStation`	numeric	-	-	ID of station providing differential corrections (null when DGPS is not used)
17	`navStatus`	character	-	V	Navigational status indicator: V (Equipment is not providing navigational status information, fixed field, only available in NMEA 4.10 and later)
18	`cs`	hexadecimal	-	*71	Checksum
19	`CRLF`	character	-	-	Carriage return and line feed

2.8.4 GSA

2.8.4.1 GNSS DOP and active satellites

Message	**NMEA-NAV2-GSA** **GNSS DOP and active satellites**
Type	Output
Comment	The GNSS receiver operating mode, satellites used for navigation, and DOP values. • If less than 12 SVs are used for navigation, the remaining fields are left empty. If more than 12 SVs are used for navigation, only the IDs of the first 12 are output. • The SV numbers (fields 'svid') are in the range of 1 to 32 for GPS satellites, and 33 to 64 for SBAS satellites (33 = SBAS PRN 120, 34 = SBAS PRN 121, and so on) **In a multi-GNSS system this message will be output multiple times, once for each GNSS.** To identify the navigation data source for NMEA Secondary filter output, the alphanumeric string source-identification (s:) parameter is used in a TAG Block, in respect to NMEA 0183 Standard.
Information	*Class/ID:* 0xf7 0x02 *Number of fields:* 25
Structure	`\s:1*78\$xxGSA,opMode,navMode{,svid},PDOP,HDOP,VDOP,systemId*cs\r\n`
Example	`\s:1*78\$GPGSA,A,3,23,29,07,08,09,18,26,28,,,,,1.94,1.18,1.54,1*0D\r\n`

Payload:

Field	*Name*	*Format*	*Unit*	*Example*	*Description*
0	`tagStart`	string	-	\s:	NMEA TAG block start and parameter
1	`source`	numeric	-	1	NMEA TAG block source value (1 for secondary output messages)
2	`tagCs`	hexadecimal	-	*78	NMEA TAG checksum
3	`tagEnd`	string	-	\	NMEA TAG block end character
4	`xxGSA`	string	-	$GPGSA	GSA Message ID (xx = current Talker ID, see NMEA Talker IDs table)
5	`opMode`	character	-	A	Operation mode: • M = Manually set to operate in 2D or 3D mode • A = Automatically switching between 2D or 3D mode
6	`navMode`	digit	-	3	Navigation mode, see position fix flags description

Start of repeated group (12 times)					
7 + n	`svid`	numeric	-	29	Satellite number
End of repeated group (12 times)					
19	`PDOP`	numeric	-	1.94	Position dilution of precision
20	`HDOP`	numeric	-	1.18	Horizontal dilution of precision
21	`VDOP`	numeric	-	1.54	Vertical dilution of precision
22	`systemId`	hexadecimal	-	1	NMEA-defined GNSS system ID, see Signal Identifiers table (only available in NMEA 4.10 and later)
23	`cs`	hexadecimal	-	*0D	Checksum
24	`CRLF`	character	-	-	Carriage return and line feed

2.8.5 RMC

2.8.5.1 Recommended minimum data

Message	**NMEA-NAV2-RMC** **Recommended minimum data**
Type	Output
Comment	The recommended minimum sentence defined by NMEA for GNSS system data. To identify the navigation data source for NMEA Secondary filter output, the alphanumeric string source-identification (s:) parameter is used in a TAG Block, in respect to NMEA 0183 Standard. ☞ The output of this message is dependent on the currently selected datum (default: WGS84)
Information	*Class/ID:* 0xf7 0x04 *Number of fields:* 20
Structure	`\s:1*78\$xxRMC,time,status,lat,NS,lon,EW,spd,cog,date,mv,mvEW,posMode,navStatus*cs\r\n`
Example	`\s:1*78\$GPRMC,083559.00,A,4717.11437,N,00833.91522,E,0.004,77.52,091202,,,A,V*57\r\n`

Payload:

Field	*Name*	*Format*	*Unit*	*Example*	*Description*
0	`tagStart`	string	-	\s:	NMEA TAG block start and parameter
1	`source`	numeric	-	1	NMEA TAG block source value (1 for secondary output messages)
2	`tagCs`	hexadecimal	-	*78	NMEA TAG checksum
3	`tagEnd`	string	-	\	NMEA TAG block end character
4	`xxRMC`	string	-	$GPRMC	RMC Message ID (xx = current Talker ID, see NMEA Talker IDs table)
5	`time`	hhmmss.ss	-	083559.00	UTC time. See section UTC representation in the integration manual for details.
6	`status`	character	-	A	Data validity status, see position fix flags description
7	`lat`	ddmm.mmmmm	-	4717.11437	Latitude (degrees and minutes), see format description
8	`NS`	character	-	N	North/South indicator
9	`lon`	dddmm.mmmmm	-	00833.91522	Longitude (degrees and minutes), see format description
10	`EW`	character	-	E	East/West indicator
11	`spd`	numeric	knots	0.004	Speed over ground
12	`cog`	numeric	deg	77.52	Course over ground

13	date	ddmmyy	-	091202	Date in day, month, year format. See section UTC representation in the integration manual for details.
14	mv	numeric	deg	-	Magnetic variation value
15	mvEW	character	-	-	Magnetic variation E/W indicator
16	posMode	character	-	A	Mode Indicator, see position fix flags description (only available in NMEA 2.3 and later)
17	navStatus	character	-	V	Navigational status indicator: V (Equipment is not providing navigational status information, fixed field, only available in NMEA 4.10 and later)
18	cs	hexadecimal	-	*57	Checksum
19	CRLF	character	-	-	Carriage return and line feed

2.8.6 VTG

2.8.6.1 Course over ground and ground speed

Message	**NMEA-NAV2-VTG** **Course over ground and ground speed**
Type	Output
Comment	Velocity is given as course over ground (COG) and speed over ground (SOG). To identify the navigation data source for NMEA Secondary filter output, the alphanumeric string source-identification (s:) parameter is used in a TAG Block, in respect to NMEA 0183 Standard.
Information	*Class/ID:* 0xf7 0x05 *Number of fields:* 16
Structure	`\s:1*78\$xxVTG,cogt,cogtUnit,cogm,cogmUnit,sogn,sognUnit,sogk,sogkUnit,posMode*cs\r\n`
Example	`\s:1*78\$GPVTG,77.52,T,,M,0.004,N,0.008,K,A*06\r\n`

Payload:

Field	*Name*	*Format*	*Unit*	*Example*	*Description*
0	tagStart	string	-	\s:	NMEA TAG block start and parameter
1	source	numeric	-	1	NMEA TAG block source value (1 for secondary output messages)
2	tagCs	hexadecimal	-	*78	NMEA TAG checksum
3	tagEnd	string	-	\	NMEA TAG block end character
4	xxVTG	string	-	$GPVTG	VTG Message ID (xx = current Talker ID, see NMEA Talker IDs table)
5	cogt	numeric	degrees	77.52	Course over ground (true)
6	cogtUnit	character	-	T	Course over ground units: T (degrees true, fixed field)
7	cogm	numeric	degrees	-	Course over ground (magnetic)
8	cogmUnit	character	-	M	Course over ground units: M (degrees magnetic, fixed field)
9	sogn	numeric	knots	0.004	Speed over ground
10	sognUnit	character	-	N	Speed over ground units: N (knots, fixed field)
11	sogk	numeric	km/h	0.008	Speed over ground
12	sogkUnit	character	-	K	Speed over ground units: K (kilometers per hour, fixed field)
13	posMode	character	-	A	Mode indicator, see position fix flags description (only available in NMEA 2.3 and later)
14	cs	hexadecimal	-	*06	Checksum

15	CRLF	character	-	-	Carriage return and line feed

2.8.7 ZDA

2.8.7.1 Time and date

Message	**NMEA-NAV2-ZDA** **Time and date**
Type	Output
Comment	UTC, day, month, year and local time zone. To identify the navigation data source for NMEA Secondary filter output, the alphanumeric string source-identification (s:) parameter is used in a TAG Block, in respect to NMEA 0183 Standard.
Information	*Class/ID:* 0xf7 0x08 *Number of fields:* 13
Structure	`\s:1*78\$GPZDA,time,day,month,year,ltzh,ltzn*cs\r\n`
Example	`\s:1*78\$xxZDA,082710.00,16,09,2002,00,00*64\r\n`

Payload:

Field	*Name*	*Format*	*Unit*	*Example*	*Description*
0	tagStart	string	-	\s:	NMEA TAG block start and parameter
1	source	numeric	-	1	NMEA TAG block source value (1 for secondary output messages)
2	tagCs	hexadecimal	-	*78	NMEA TAG checksum
3	tagEnd	string	-	\	NMEA TAG block end character
4	xxZDA	string	-	$GPZDA	ZDA Message ID (xx = current Talker ID, see NMEA Talker IDs table)
5	time	hhmmss.ss	-	082710.00	UTC Time. See section UTC representation in the integration manual for details.
6	day	dd	day	16	UTC day (range: 1-31)
7	month	mm	month	09	UTC month (range: 1-12)
8	year	yyyy	year	2002	UTC year
9	ltzh	xx	-	00	Local time zone hours (fixed field, always 00)
10	ltzn	zz	-	00	Local time zone minutes (fixed field, always 00)
11	cs	hexadecimal	-	*64	Checksum
12	CRLF	character	-	-	Carriage return and line feed

2.9 PUBX messages

Proprietary NMEA messages for u-blox positioning receivers. See also NMEA-proprietary messages.

2.9.1 CONFIG (PUBX,41)

2.9.1.1 Set protocols and baud rate

Message	**NMEA-PUBX-CONFIG** **Set protocols and baud rate**
Type	Set
Comment	
Information	*Class/ID:* 0xf1 0x41 *Number of fields:* 9
Structure	`$PUBX,41,portId,inProto,outProto,baudrate,autobauding*cs\r\n`

Example	`$PUBX,41,1,0007,0003,19200,0*25\r\n`

Payload:

Field	*Name*	*Format*	*Unit*	*Example*	*Description*
0	`PUBX`	string	-	$PUBX	Message ID, UBX protocol header, proprietary sentence
1	`msgId`	numeric	-	41	Proprietary message identifier
2	`portId`	numeric	-	1	ID of communication port. See section Communication ports in the integration manual for details.
3	`inProto`	hexadecimal	-	0007	Input protocol mask. Bitmask, specifying which protocols(s) are allowed for input. See section Communication ports in the integration manual for details.
4	`outProto`	hexadecimal	-	0003	Output protocol mask. Bitmask, specifying which protocols(s) are allowed for input. See section Communication ports in the integration manual for details.
5	`baudrate`	numeric	bits/s	19200	Baud rate
6	`autobauding`	numeric	-	-	Autobauding: 1=enable, 0=disable (not supported on u-blox 5, set to 0)
7	`cs`	hexadecimal	-	*25	Checksum
8	`CRLF`	character	-	-	Carriage return and line feed

2.9.2 POSITION (PUBX,00)

2.9.2.1 Poll a PUBX,00 message

Message	**NMEA-PUBX-POSITION** **Poll a PUBX,00 message**
Type	Poll request
Comment	A PUBX,00 message is polled by sending the PUBX,00 message without any data fields.
Information	*Class/ID:* 0xf1 0x00 *Number of fields:* 4
Structure	`$PUBX,00*33\r\n`
Example	`$PUBX,00*33\r\n`

Payload:

Field	*Name*	*Format*	*Unit*	*Example*	*Description*
0	`PUBX`	string	-	$PUBX	Message ID, UBX protocol header, proprietary sentence
1	`msgId`	numeric	-	00	Set to 00 to poll a PUBX,00 message
2	`cs`	hexadecimal	-	*33	Checksum
3	`CRLF`	character	-	-	Carriage return and line feed

2.9.2.2 Lat/Long position data

Message	**NMEA-PUBX-POSITION** **Lat/Long position data**
Type	Output
Comment	This message contains position solution data. The datum selection may be changed using the message UBX-CFG-DAT. ☞ The output of this message is dependent on the currently selected datum (default: WGS84).
Information	*Class/ID:* 0xf1 0x00 *Number of fields:* 23
Structure	`$PUBX,00,time,lat,NS,long,EW,altRef,navStat,hAcc,vAcc,SOG,COG,vVel,diffAge,HDOP,VDOP ↲ ,TDOP,numSvs,reserved,DR,*cs\r\n`

Example	`$PUBX,00,081350.00,4717.113210,N,00833.915187,E,546.589,G3,2.1,2.0,0.007,77.52,0.007 ,,0.92,1.19,0.77,9,0,0*5F\r\n`

Payload:

Field	*Name*	*Format*	*Unit*	*Example*	*Description*
0	`PUBX`	string	-	$PUBX	Message ID, UBX protocol header, proprietary sentence
1	`msgId`	numeric	-	00	Proprietary message identifier: 00
2	`time`	hhmmss.ss	-	081350.00	UTC time. See section UTC representation in the integration manual for details.
3	`lat`	ddmm. mmmmm	-	4717.113210	Latitude (degrees and minutes), see format description
4	`NS`	character	-	N	North/South Indicator
5	`long`	dddmm. mmmmm	-	00833.915187	Longitude (degrees and minutes), see format description
6	`EW`	character	-	E	East/West indicator
7	`altRef`	numeric	m	546.589	Altitude above user datum ellipsoid
8	`navStat`	string	-	G3	Navigation Status: • NF = No Fix • DR = Dead reckoning only solution • G2 = Stand alone 2D solution • G3 = Stand alone 3D solution • D2 = Differential 2D solution • D3 = Differential 3D solution • RK = Combined GPS + dead reckoning solution • TT = Time only solution
9	`hAcc`	numeric	m	2.1	Horizontal accuracy estimate
10	`vAcc`	numeric	m	2.0	Vertical accuracy estimate
11	`SOG`	numeric	km/h	0.007	Speed over ground
12	`COG`	numeric	deg	77.52	Course over ground
13	`vVel`	numeric	m/s	0.007	Vertical velocity (positive downwards)
14	`diffAge`	numeric	s	-	Age of differential corrections (blank when DGPS is not used)
15	`HDOP`	numeric	-	0.92	HDOP, Horizontal Dilution of Precision
16	`VDOP`	numeric	-	1.19	VDOP, Vertical Dilution of Precision
17	`TDOP`	numeric	-	0.77	TDOP, Time Dilution of Precision
18	`numSvs`	numeric	-	9	Number of satellites used in the navigation solution
19	`reserved`	numeric	-	-	Reserved, always set to 0
20	`DR`	numeric	-	-	DR used
21	`cs`	hexadecimal	-	*5B	Checksum
22	`CRLF`	character	-	-	Carriage return and line feed

2.9.3 RATE (PUBX,40)

2.9.3.1 Set NMEA message output rate

Message	**NMEA-PUBX-RATE** **Set NMEA message output rate**
Type	Set
Comment	Set/Get message rate configuration (s) to/from the receiver.

- Send rate is relative to the event a message is registered on. For example, if the rate of a navigation message is set to 2, the message is sent every second navigation solution.

Information	*Class/ID:* 0xf1 0x40 *Number of fields:* 11
Structure	`$PUBX,40,msgId,rddc,rus1,rus2,rusb,rspi,reserved*cs\r\n`
Example	`$PUBX,40,GLL,1,0,0,0,0,0*5D\r\n`

Payload:

Field	*Name*	*Format*	*Unit*	*Example*	*Description*
0	`PUBX`	string	-	$PUBX	Message ID, UBX protocol header, proprietary sentence
1	`ID`	numeric	-	40	Proprietary message identifier
2	`msgId`	string	-	GLL	NMEA message identifier
3	`rddc`	numeric	cycles	1	output rate on DDC • 0 disables that message from being output on this port • 1 means that this message is output every epoch
4	`rus1`	numeric	cycles	1	output rate on USART 1 • 0 disables that message from being output on this port • 1 means that this message is output every epoch
5	`rus2`	numeric	cycles	1	output rate on USART 2 • 0 disables that message from being output on this port • 1 means that this message is output every epoch
6	`rusb`	numeric	cycles	1	output rate on USB • 0 disables that message from being output on this port • 1 means that this message is output every epoch
7	`rspi`	numeric	cycles	1	output rate on SPI • 0 disables that message from being output on this port • 1 means that this message is output every epoch
8	`reserved`	numeric	-	-	Reserved: always fill with 0
9	`cs`	hexadecimal	-	*5D	Checksum
10	`CRLF`	character	-	-	Carriage return and line feed

2.9.4 SVSTATUS (PUBX,03)

2.9.4.1 Poll a PUBX,03 message

Message	**NMEA-PUBX-SVSTATUS** **Poll a PUBX,03 message**
Type	Poll request
Comment	A PUBX,03 message is polled by sending the PUBX,03 message without any data fields.
Information	*Class/ID:* 0xf1 0x03 *Number of fields:* 4
Structure	`$PUBX,03*30\r\n`
Example	`$PUBX,03*30\r\n`

Payload:

Field	*Name*	*Format*	*Unit*	*Example*	*Description*
0	`PUBX`	string	-	$PUBX	Message ID, UBX protocol header, proprietary sentence
1	`msgId`	numeric	-	03	Set to 03 to poll a PUBX,03 message

2	cs	hexadecimal	-	*30	Checksum
3	CRLF	character	-	-	Carriage return and line feed

2.9.4.2 Satellite status

Message	**NMEA-PUBX-SVSTATUS** **Satellite status**
Type	Output
Comment	The PUBX,03 message contains satellite status information.
Information	*Class/ID:* 0xf1 0x03 *Number of fields:* 5 + n·6
Structure	`$PUBX,03,GT{,sv,s,az,el,cno,lck},*cs\r\n`
Example	`$PUBX,03,11,23,-,,,45,010,29,-,,,46,013,07,-,,,42,015,08,U,067,31,42,025,10,U,195,33 ,46,026,18,U,326,08,39,026,17,-,,,32,015,26,U,306,66,48,025,27,U,073,10,36,026,28,U, 089,61,46,024,15,-,,,39,014*0D\r\n`

Payload:

Field	*Name*	*Format*	*Unit*	*Example*	*Description*
0	PUBX	string	-	$PUBX	Message ID, UBX protocol header, proprietary sentence
1	msgId	numeric	-	03	Proprietary message identifier: 03
2	n	numeric	-	11	Number of GNSS satellites tracked
Start of repeated group (n times)					
3 + n·6	sv	numeric	-	23	Satellite ID according to UBX svId mapping (see Satellite Numbering)
4 + n·6	s	character	-	-	Satellite status: • - = Not used • U = Used in solution • e = Ephemeris available, but not used for navigation
5 + n·6	az	numeric	deg	-	Satellite azimuth (range: 0-359)
6 + n·6	el	numeric	deg	-	Satellite elevation (<= 90)
7 + n·6	cno	numeric	dBHz	45	Signal strength (C/N0, range 0-99), blank when not tracking
8 + n·6	lck	numeric	s	010	Satellite carrier lock time (range: 0-64) • 0 = code lock only • 64 = lock for 64 seconds or more
End of repeated group (n times)					
3 + n·6	cs	hexadecimal	-	*0D	Checksum
4 + n·6	CRLF	character	-	-	Carriage return and line feed

2.9.5 TIME (PUBX,04)

2.9.5.1 Poll a PUBX,04 message

Message	**NMEA-PUBX-TIME** **Poll a PUBX,04 message**
Type	Poll request
Comment	A PUBX,04 message is polled by sending the PUBX,04 message without any data fields.
Information	*Class/ID:* 0xf1 0x04 *Number of fields:* 4
Structure	`$PUBX,04*37\r\n`
Example	`$PUBX,04*37\r\n`

Payload:

Field	Name	Format	Unit	Example	Description
0	PUBX	string	-	$PUBX	Message ID, UBX protocol header, proprietary sentence
1	msgId	numeric	-	04	Set to 04 to poll a PUBX,04 message
2	cs	hexadecimal	-	*37	Checksum
3	CRLF	character	-	-	Carriage return and line feed

2.9.5.2 Time of day and clock information

Message	**NMEA-PUBX-TIME** **Time of day and clock information**
Type	Output
Comment	
Information	*Class/ID:* 0xf1 0x04 *Number of fields:* 12
Structure	`$PUBX,04,time,date,utcTow,utcWk,leapSec,clkBias,clkDrift,tpGran,*cs\r\n`
Example	`$PUBX,04,073731.00,091202,113851.00,1196,15D,1930035,-2660.664,43,*3C\r\n`

Payload:

Field	Name	Format	Unit	Example	Description
0	PUBX	string	-	$PUBX	Message ID, UBX protocol header, proprietary sentence
1	msgId	numeric	-	04	Proprietary message identifier: 04
2	time	hhmmss.ss	-	073731.00	UTC time. See section UTC representation in the integration manual for details.
3	date	ddmmyy	-	091202	UTC date, day, month, year. See section UTC representation in the integration manual for details.
4	utcTow	numeric	s	113851.00	UTC time of week
5	utcWk	numeric	-	1196	UTC week number, continues beyond 1023
6	leapSec	numeric/ text	s	15D	Leap seconds (not supported for protocol versions less than 13.01) The number is marked with a *D* if the value is the firmware default value. If the value is not marked it has been received from a satellite.
7	clkBias	numeric	ns	1930035	Receiver clock bias
8	clkDrift	numeric	ns/s	-2660.664	Receiver clock drift
9	tpGran	numeric	ns	43	Time pulse granularity, the quantization error of the TIMEPULSE pin
10	cs	hexadecimal	-	*3C	Checksum
11	CRLF	character	-	-	Carriage return and line feed

3 UBX protocol

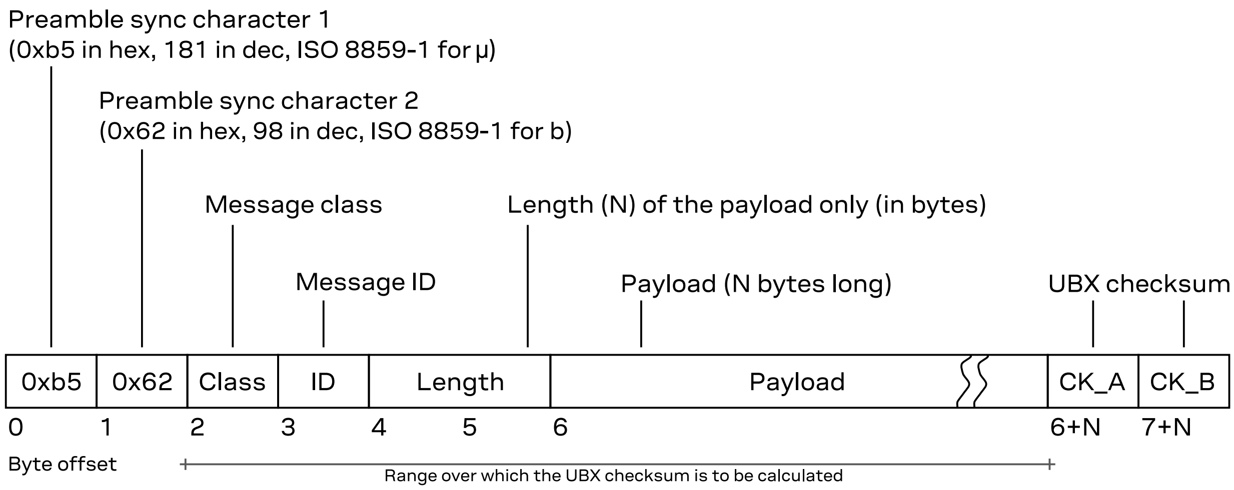
3.1 UBX protocol key features

u-blox receivers support a u-blox-proprietary protocol to communicate with a host computer. This protocol has the following key features:

- Compact – uses 8-bit binary data
- Checksum protected – uses a low-overhead checksum algorithm
- Modular – uses a two-stage message identifier (Class and Message ID)

3.2 UBX frame structure

The structure of a basic UBX frame is shown in the following diagram.



- Every *frame* starts with a 2-byte *preamble* consisting of two synchronization characters: `0xb5` and `0x62`.
- A 1-byte *message class* field follows. A class is a group of messages that are related to each other.
- A 1-byte *message ID* field defines the message that is to follow.
- A 2-byte *length* field follows. The length is defined as being that of the payload only. It does not include the preamble, message class, message ID, length, or UBX checksum fields. The number format of the length field is an unsigned little-endian 16-bit integer (a "U2" in UBX data types).
- The *payload* field contains a variable number (= *length*) of bytes.
- The two 1-byte *CK_A* and *CK_B* fields hold a 16-bit checksum whose calculation is defined in UBX checksum section. This concludes the frame.

3.3 UBX payload definition rules

This section contains the rules and guidelines for UBX message payloads. See also UBX message example.

3.3.1 UBX structure packing

Values are placed in such an order that structure packing is not a problem. This means that two-byte values shall start on offsets that are a multiple of two; four-byte values shall start at a multiple of four; and so on.

3.3.2 UBX reserved elements

Some messages contain reserved fields or bits to allow for future expansion. The contents of these elements should be ignored in output messages and must be set to zero in input messages. Where a message is output and subsequently returned to the receiver as an input message, reserved elements can either be explicitly set to zero or left with whatever value they were output with.

For fields in a bitfield the same rules apply. Note that bits not described are automatically reserved and are not explicitly stated (see UBX message example).

3.3.3 UBX undefined values

The description of some fields provide specific meanings for specific values. For example, the field `gnssId` appears in many UBX messages and uses 0 to indicate GPS, 1 for SBAS and so on (see GNSS identifiers for details); however it is usually stored in a byte with far more possible values than the handful currently defined. All such undefined values are reserved for future expansion and therefore should not be used.

3.3.4 UBX conditional values

Some UBX messages use validity flag fields to indicate whether the values of some value fields are valid. For example the UBX-NAV-PVT message has the `validDate` and `validTime` fields that indicate whether the date (`year`, `month` and `day` fields), and, respectively, the time (`hour`, `min` and `sec` fields) are valid. This means that these value fields will only contain meaningful data if the corresponding flag field is set (has the value 1).

3.3.5 UBX data types

The following data types (number formats) are defined.

Name	Type	Size (Bytes)	Range	Resolution
U1	unsigned 8-bit integer	1	$0...2^{8}-1$	1
I1	signed 8-bit integer, two's complement	1	$-2^{7}...2^{7}-1$	1
X1	8-bit bitfield	1	n/a	n/a
U2	unsigned little-endian 16-bit integer	2	$0...2^{16}-1$	1
I2	signed little-endian 16-bit integer, two's complement	2	$-2^{15}...2^{15}-1$	1
X2	16-bit little-endian bitfield	2	n/a	n/a
U4	unsigned little-endian 32-bit integer	4	$0...2^{32}-1$	1
I4	signed little-endian 32-bit integer, two's complement	4	$-2^{31}...2^{31}-1$	1
X4	32-bit little-endian bitfield	4	n/a	n/a

Name	Type	Size (Bytes)	Range	Resolution
R4	IEEE 754 single (32-bit) precision	4	$-2^{127}...2^{127}$	~ value$\cdot 2^{-24}$
R8	IEEE 754 double (64-bit) precision	8	$-2^{1023}...2^{1023}$	~ value$\cdot 2^{-53}$
CH	ASCII / ISO 8859-1 char (8-bit)	1	n/a	n/a
U:*n*	unsigned bitfield value of *n* bits width	var.	variable	variable
I:*n*	signed (two's complement) bitfield value of *n* bits width	var.	variable	variable
S:*n*	signed bitfield value of *n* bits width, in sign (most significant bit) and magnitude (remaining bits) notation	var.	variable	variable

3.3.6 UBX fields scale and unit

Fields in UBX messages can have a unit defined. Whenever possible, SI units and symbols are used (e.g. "m" for meters, "s" for seconds). For civil (UTC) time representation units of years (y), months (month), days (d), hours (h), minutes (min) and seconds (s) are used.

Fields in UBX messages can have a scale factor defined. Unity (factor 1) is assumed if no scale is specified. For integer type fields this is often combined with a unit. When a scale is combined with a unit, the scale represents the smallest storage unit. For example, if meters (m) are expressed (stored) in centimeters the scale would be 0.01 (or 1e-2). This is equivalent of specifying a unit of centimeters (cm) and no scale.

The description of some integer values (e.g. U2, I4 or I8) indicates a fixed-point format (e.g. [UU.FF], [IIIII.FFF] or [IIIIIII.FFFFFFFF]). The fixed-point value can be retrieved from the integer value by first casting it to appropriate type (e.g. as a floating-point number) and then scaling it with the indicated scaling factor.

3.3.7 UBX repeated fields

There are two types of repetitions in UBX messages. The first type specifies that a single field is repeated a constant number of times. This repetition is defined in the type of the field. For example, the UBX message example can specify a field `data` of type *U1[5]*. In this case the `data` field should be interpreted as an array of five U1 values.

The second type of repetition in messages is referred to as *repeated groups*, which groups one or more fields into a block of payload data. There are several types of repetition:

- The number of repetitions of *variable-by-field group* is indicated by another, earlier field in the same message. The number of repetitions can be zero or more, depending on the value of the referenced field.
- A *constant group* has a constant number of repetitions.
- An *optional group* is repeated zero or one times, depending on the available payload data. That is, the fields are present in the message only if the payload of the message is large enough to cover the whole group of fields.
- The number of repetitions of a *variable-by-size* group is given by the available payload size. The group will repeat until there is not enough payload data left to cover the whole group of fields another time.

Note that only some combinations of repeated groups of fields are possible in a single message. See also UBX payload decoding.

3.3.8 UBX payload decoding

UBX message payloads are designed so that the data (fields) can be extracted by a single pass through the payload from start to end. Fixed-size messages are the trivial case where the offset of all fields is unambiguously defined. Variable-size messages have variable number of repetitions of one or multiple groups of fields. For groups where the number of repetitions is given by the value of another field, that field can always be found at a fixed offset in the message payload before the respective group of fields. Groups whose number of repetitions depend on the payload size can only be the last group of fields in a message and only one such group may exist in a message. See also UBX repeated fields.

3.4 UBX checksum

The checksum is calculated over the message, starting and including the class field up until, but excluding, the checksum fields (see the figure UBX frame structure).

The checksum algorithm used is the 8-bit Fletcher algorithm, which is used in the TCP standard RFC 1145). This algorithm works as follows:

- `Buffer[N]` is an array of bytes that contains the data over which the checksum is to be calculated.
- The two `CK_A` and `CK_A` values are 8-bit unsigned integers, only! If implementing with larger-sized integer values, make sure to mask both `CK_A` and `CK_B` with the value `0xff` after both operations in the loop.
- After the loop, the two *U1* values contain the checksum, transmitted after the message payload, which concludes the frame.

```
CK_A = 0, CK_B = 0
For (I = 0; I < N; I++)
{
    CK_A = CK_A + Buffer[I]
    CK_B = CK_B + CK_A
}
```

3.5 UBX message flow

There are certain features associated with the messages being sent back and forth:

3.5.1 UBX acknowledgement

When messages from the class CFG are sent to the receiver, the receiver will send an "acknowledge" (UBX-ACK-ACK) or a "not acknowledge" (UBX-ACK-NAK) message back to the sender, depending on whether or not the message was processed correctly.

Some messages from other classes also use the same acknowledgement mechanism.

3.5.2 UBX polling mechanism

The UBX protocol is designed so that messages can be polled by sending the message required to the receiver but without a payload (or with just a single parameter that identifies the poll request). The receiver then responds with the same message with the payload populated.

3.6 GNSS, satellite and signal numbering

See GNSS, satellite and signal identifiers for details on how GNSS, satellites and signals are numbered in the UBX protocol.

3.7 UBX message example

This is an example of the definition of UBX messages as shown in the following sections.

Message ❶	**UBX-DEMO-EXAMPLE** **Example demo message**				
Type ❷	Periodic/polled				
Comment ❸	This is a comment that describes the use of the demo example message. There can be references to other sections in the documentation (such as: UBX protocol). ☞ Note that there can be important remarks here.				
Message ❹ *Structure*	*Header*	*Class ID*	*Length (bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x01 0x07	16 + numRepeat*4	*see below*	CK_A CK_B

Payload description: ❺

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	U4	`aField`	-	-	a field that contains an unsigned integer with no particular scale or unit
4	I4	`anotherField`	1e-2	m	a field that contains a length in meters (m) with a scale of 1e-2 (= 0.01), i.e. a length in centimeters
8	X2	`bitfield` ❻	-	-	this field contains flags or values smaller than one byte, whose definition follows below (bits not described are reserved)
bit 0	U:1	`aFieldValid`	-	-	the first bit in `bitfield` indicates whether the `aField` is valid or not (see UBX conditional values)
bit 1	U:1	`someFlag`	-	-	the second bit is a flag (1 = true, 0 = false)
bits 5…2	U:4	`aBitFieldValue`	-	-	a 4-bits value (range: 0…15)
10	U1[5] ❼	`reserved0`	-	-	a reserved field, whose value shall be ignored (in output messages) or set to 0 (in input messages)
15	U1	`numRepeat`	-	-	number of repetitions in the group of fields below
Start of repeated group (`numRepeat` times) ❽					
16 + n*4	I2	`someValue`	-	-	a signed value in a repeated group of fields
18 + n*4	U2	`anotherValue`	-	-	another value in a repeated group of fields
End of repeated group (`numRepeat` times)					

❶ The first line shows the message name (see Naming). The second line shows a short description of the message.

❷ The message type (see Message types).

❸ This section contains comments that describe the message. Often links to other related sections in the documentation or other related messages are found here.

❹ The message structure gives the parameters for the UBX frame structure, notably the message class and message ID values and the payload length. For many messages the payload length is a fixed number (of bytes). Messages that contain repeated blocks of information (fields) have a variable payload (see UBX repeated fields).

❺ The message payload definition is given as a list of fields and their parameters. Each field starts at a specified offset (in bytes) in the payload (see also UBX structure packing), is of a specific type (see UBX data types), has a unique name (within the message), and a description. Optionally, fields can have a scale and/or a unit (see UBX fields scale and unit).

❻ Bitfields ("X" types) are broken down into smaller parts. Each part can be one or more bits wide. Values that are two or more bits wide can be unsigned or one of two signed value representation (see UBX data types). Note that the ten unused bits 15…6 are not explicitly stated as UBX reserved elements.

❼ Fields can be arrays of values of the same type (see UBX repeated fields).

❽ Groups of fields can be repeated in the payload. The number of repetitions can be given by another field in the message (this example), a constant number, zero or one times (known as "optional group"), or derived from the remaining payload size (labeled as "repeated N times"). See also UBX repeated fields and UBX payload decoding.

3.8 UBX messages overview

Message	***Class/ID***	***Description (Type)***
UBX-ACK – Acknowledgement and negative acknowledgement messages		
UBX-ACK-ACK	0x05 0x01	• Message acknowledged (Output)
UBX-ACK-NAK	0x05 0x00	• Message not acknowledged (Output)
UBX-CFG – Configuration and command messages		
UBX-CFG-ANT	0x06 0x13	• Antenna control settings (Get/set)
UBX-CFG-CFG	0x06 0x09	• Clear, save and load configurations (Command)
UBX-CFG-DAT	0x06 0x06	• Set user-defined datum (Set) • Get currently defined datum (Get)
UBX-CFG-DGNSS	0x06 0x70	• DGNSS configuration (Get/set)
UBX-CFG-GEOFENCE	0x06 0x69	• Geofencing configuration (Get/set)
UBX-CFG-GNSS	0x06 0x3e	• GNSS system configuration (Get/set)
UBX-CFG-INF	0x06 0x02	• Poll configuration for one protocol (Poll request) • Information message configuration (Get/set)
UBX-CFG-ITFM	0x06 0x39	• Jamming/interference monitor configuration (Get/set)
UBX-CFG-LOGFILTER	0x06 0x47	• Data logger configuration (Get/set)
UBX-CFG-MSG	0x06 0x01	• Poll a message configuration (Poll request) • Set message rate(s) (Get/set) • Set message rate (Get/set)
UBX-CFG-NAV5	0x06 0x24	• Navigation engine settings (Get/set)
UBX-CFG-NAVX5	0x06 0x23	• Navigation engine expert settings (Get/set)
UBX-CFG-NMEA	0x06 0x17	• Extended NMEA protocol configuration V1 (Get/set)
UBX-CFG-ODO	0x06 0x1e	• Odometer, low-speed COG engine settings (Get/set)
UBX-CFG-PRT	0x06 0x00	• Polls the configuration for one I/O port (Poll request) • Port configuration for UART ports (Get/set) • Port configuration for USB port (Get/set) • Port configuration for SPI port (Get/set)

Message	Class/ID	Description (Type)
		• Port configuration for I2C (DDC) port (Get/set)
UBX-CFG-PWR	0x06 0x57	• Put receiver in a defined power state (Set)
UBX-CFG-RATE	0x06 0x08	• Navigation/measurement rate settings (Get/set)
UBX-CFG-RINV	0x06 0x34	• Contents of remote inventory (Get/set)
UBX-CFG-RST	0x06 0x04	• Reset receiver / Clear backup data structures (Command)
UBX-CFG-SBAS	0x06 0x16	• SBAS configuration (Get/set)
UBX-CFG-TMODE3	0x06 0x71	• Time mode settings 3 (Get/set)
UBX-CFG-TP5	0x06 0x31	• Time pulse parameters (Get/set)
UBX-CFG-USB	0x06 0x1b	• USB configuration (Get/set)
UBX-CFG-VALDEL	0x06 0x8c	• Delete configuration item values (Set) • Delete configuration item values (with transaction) (Set)
UBX-CFG-VALGET	0x06 0x8b	• Get configuration items (Poll request) • Configuration items (Polled)
UBX-CFG-VALSET	0x06 0x8a	• Set configuration item values (Set) • Set configuration item values (with transaction) (Set)
UBX-INF – Information messages		
UBX-INF-DEBUG	0x04 0x04	• ASCII output with debug contents (Output)
UBX-INF-ERROR	0x04 0x00	• ASCII output with error contents (Output)
UBX-INF-NOTICE	0x04 0x02	• ASCII output with informational contents (Output)
UBX-INF-TEST	0x04 0x03	• ASCII output with test contents (Output)
UBX-INF-WARNING	0x04 0x01	• ASCII output with warning contents (Output)
UBX-LOG – Logging messages		
UBX-LOG-CREATE	0x21 0x07	• Create log file (Command)
UBX-LOG-ERASE	0x21 0x03	• Erase logged data (Command)
UBX-LOG-FINDTIME	0x21 0x0e	• Find index of a log entry based on a given time (Input) • Response to FINDTIME request (Output)
UBX-LOG-INFO	0x21 0x08	• Poll for log information (Poll request) • Log information (Output)
UBX-LOG-RETRIEVE	0x21 0x09	• Request log data (Command)
UBX-LOG-RETRIEVEPOS	0x21 0x0b	• Position fix log entry (Output)
UBX-LOG-RETRIEVEPOSEXTRA	0x21 0x0f	• Odometer log entry (Output)
UBX-LOG-RETRIEVESTRING	0x21 0x0d	• Byte string log entry (Output)
UBX-LOG-STRING	0x21 0x04	• Store arbitrary string in on-board flash (Command)
UBX-MGA – GNSS assistance (A-GNSS) messages		
UBX-MGA-ACK	0x13 0x60	• Multiple GNSS acknowledge message (Output)
UBX-MGA-BDS	0x13 0x03	• BeiDou ephemeris assistance (Input) • BeiDou almanac assistance (Input) • BeiDou health assistance (Input) • BeiDou UTC assistance (Input) • BeiDou ionosphere assistance (Input)
UBX-MGA-DBD	0x13 0x80	• Poll the navigation database (Poll request) • Navigation database dump entry (Input/output)
UBX-MGA-GAL	0x13 0x02	• Galileo ephemeris assistance (Input) • Galileo almanac assistance (Input) • Galileo GPS time offset assistance (Input) • Galileo UTC assistance (Input)

Message	Class/ID	Description (Type)
UBX-MGA-GLO	0x13 0x06	• GLONASS ephemeris assistance (Input) • GLONASS almanac assistance (Input) • GLONASS auxiliary time offset assistance (Input)
UBX-MGA-GPS	0x13 0x00	• GPS ephemeris assistance (Input) • GPS almanac assistance (Input) • GPS health assistance (Input) • GPS UTC assistance (Input) • GPS ionosphere assistance (Input)
UBX-MGA-INI	0x13 0x40	• Initial position assistance (Input) • Initial time assistance (Input) • Initial clock drift assistance (Input) • Initial frequency assistance (Input)
UBX-MGA-QZSS	0x13 0x05	• QZSS ephemeris assistance (Input) • QZSS almanac assistance (Input) • QZSS health assistance (Input)
UBX-MON – Monitoring messages		
UBX-MON-COMMS	0x0a 0x36	• Communication port information (Periodic/polled)
UBX-MON-GNSS	0x0a 0x28	• Information message major GNSS selection (Polled)
UBX-MON-HW	0x0a 0x09	• Hardware status (Periodic/polled)
UBX-MON-HW2	0x0a 0x0b	• Extended hardware status (Periodic/polled)
UBX-MON-HW3	0x0a 0x37	• I/O pin status (Periodic/polled)
UBX-MON-IO	0x0a 0x02	• I/O system status (Periodic/polled)
UBX-MON-MSGPP	0x0a 0x06	• Message parse and process status (Periodic/polled)
UBX-MON-PATCH	0x0a 0x27	• Installed patches (Polled)
UBX-MON-RF	0x0a 0x38	• RF information (Periodic/polled)
UBX-MON-RXBUF	0x0a 0x07	• Receiver buffer status (Periodic/polled)
UBX-MON-RXR	0x0a 0x21	• Receiver status information (Output)
UBX-MON-SPAN	0x0a 0x31	• Signal characteristics (Periodic/polled)
UBX-MON-SYS	0x0a 0x39	• Current system performance information (Periodic/polled)
UBX-MON-TXBUF	0x0a 0x08	• Transmitter buffer status (Periodic/polled)
UBX-MON-VER	0x0a 0x04	• Receiver and software version (Polled)
UBX-NAV – Navigation solution messages		
UBX-NAV-CLOCK	0x01 0x22	• Clock solution (Periodic/polled)
UBX-NAV-COV	0x01 0x36	• Covariance matrices (Periodic/polled)
UBX-NAV-DOP	0x01 0x04	• Dilution of precision (Periodic/polled)
UBX-NAV-EOE	0x01 0x61	• End of epoch (Periodic)
UBX-NAV-GEOFENCE	0x01 0x39	• Geofencing status (Periodic/polled)
UBX-NAV-HPPOSECEF	0x01 0x13	• High precision position solution in ECEF (Periodic/polled)
UBX-NAV-HPPOSLLH	0x01 0x14	• High precision geodetic position solution (Periodic/polled)
UBX-NAV-ODO	0x01 0x09	• Odometer solution (Periodic/polled)
UBX-NAV-ORB	0x01 0x34	• GNSS orbit database info (Periodic/polled)
UBX-NAV-PL	0x01 0x62	• Protection level information (Periodic)
UBX-NAV-POSECEF	0x01 0x01	• Position solution in ECEF (Periodic/polled)
UBX-NAV-POSLLH	0x01 0x02	• Geodetic position solution (Periodic/polled)
UBX-NAV-PVT	0x01 0x07	• Navigation position velocity time solution (Periodic/polled)
UBX-NAV-RELPOSNED	0x01 0x3c	• Relative positioning information in NED frame (Periodic/polled)

Message	Class/ID	Description (Type)
UBX-NAV-RESETODO	0x01 0x10	• Reset odometer (Command)
UBX-NAV-SAT	0x01 0x35	• Satellite information (Periodic/polled)
UBX-NAV-SBAS	0x01 0x32	• SBAS status data (Periodic/polled)
UBX-NAV-SIG	0x01 0x43	• Signal information (Periodic/polled)
UBX-NAV-SLAS	0x01 0x42	• QZSS L1S SLAS status data (Periodic/polled)
UBX-NAV-STATUS	0x01 0x03	• Receiver navigation status (Periodic/polled)
UBX-NAV-SVIN	0x01 0x3b	• Survey-in data (Periodic/polled)
UBX-NAV-TIMEBDS	0x01 0x24	• BeiDou time solution (Periodic/polled)
UBX-NAV-TIMEGAL	0x01 0x25	• Galileo time solution (Periodic/polled)
UBX-NAV-TIMEGLO	0x01 0x23	• GLONASS time solution (Periodic/polled)
UBX-NAV-TIMEGPS	0x01 0x20	• GPS time solution (Periodic/polled)
UBX-NAV-TIMELS	0x01 0x26	• Leap second event information (Periodic/polled)
UBX-NAV-TIMEQZSS	0x01 0x27	• QZSS time solution (Periodic/polled)
UBX-NAV-TIMEUTC	0x01 0x21	• UTC time solution (Periodic/polled)
UBX-NAV-VELECEF	0x01 0x11	• Velocity solution in ECEF (Periodic/polled)
UBX-NAV-VELNED	0x01 0x12	• Velocity solution in NED frame (Periodic/polled)
UBX-NAV2 – Navigation solution messages (Secondary output)		
UBX-NAV2-CLOCK	0x29 0x22	• Clock solution (Periodic/polled)
UBX-NAV2-COV	0x29 0x36	• Covariance matrices (Periodic/polled)
UBX-NAV2-DOP	0x29 0x04	• Dilution of precision (Periodic/polled)
UBX-NAV2-EOE	0x29 0x61	• End of epoch (Periodic)
UBX-NAV2-ODO	0x29 0x09	• Odometer solution (Periodic/polled)
UBX-NAV2-POSECEF	0x29 0x01	• Position solution in ECEF (Periodic/polled)
UBX-NAV2-POSLLH	0x29 0x02	• Geodetic position solution (Periodic/polled)
UBX-NAV2-PVT	0x29 0x07	• Navigation position velocity time solution (Periodic/polled)
UBX-NAV2-SAT	0x29 0x35	• Satellite information (Periodic/polled)
UBX-NAV2-SBAS	0x29 0x32	• SBAS status data (Periodic/polled)
UBX-NAV2-SIG	0x29 0x43	• Signal information (Periodic/polled)
UBX-NAV2-SLAS	0x29 0x42	• QZSS L1S SLAS status data (Periodic/polled)
UBX-NAV2-STATUS	0x29 0x03	• Receiver navigation status (Periodic/polled)
UBX-NAV2-SVIN	0x29 0x3b	• Survey-in data (Periodic/polled)
UBX-NAV2-TIMEBDS	0x29 0x24	• BeiDou time solution (Periodic/polled)
UBX-NAV2-TIMEGAL	0x29 0x25	• Galileo time solution (Periodic/polled)
UBX-NAV2-TIMEGLO	0x29 0x23	• GLONASS time solution (Periodic/polled)
UBX-NAV2-TIMEGPS	0x29 0x20	• GPS time solution (Periodic/polled)
UBX-NAV2-TIMELS	0x29 0x26	• Leap second event information (Periodic/polled)
UBX-NAV2-TIMEQZSS	0x29 0x27	• QZSS time solution (Periodic/polled)
UBX-NAV2-TIMEUTC	0x29 0x21	• UTC time solution (Periodic/polled)
UBX-NAV2-VELECEF	0x29 0x11	• Velocity solution in ECEF (Periodic/polled)
UBX-NAV2-VELNED	0x29 0x12	• Velocity solution in NED frame (Periodic/polled)
UBX-RXM – Receiver manager messages		
UBX-RXM-COR	0x02 0x34	• Differential correction input status (Output)
UBX-RXM-MEASX	0x02 0x14	• Satellite measurements for RRLP (Periodic/polled)
UBX-RXM-PMP	0x02 0x72	• PMP (LBAND) message (Input)

Message	Class/ID	Description (Type)
UBX-RXM-PMREQ	0x02 0x41	• Power management request (Command)
UBX-RXM-QZSSL6	0x02 0x73	• QZSS L6 message (Input)
UBX-RXM-RAWX	0x02 0x15	• Multi-GNSS raw measurements (Periodic/polled)
UBX-RXM-RLM	0x02 0x59	• Galileo SAR short-RLM report (Output) • Galileo SAR long-RLM report (Output)
UBX-RXM-RTCM	0x02 0x32	• RTCM input status (Output)
UBX-RXM-SFRBX	0x02 0x13	• Broadcast navigation data subframe (Output)
UBX-RXM-SPARTN	0x02 0x33	• SPARTN input status (Output)
UBX-RXM-SPARTNKEY	0x02 0x36	• Poll installed keys (Poll request) • Transfer dynamic SPARTN keys (Input/output)
UBX-SEC – Security messages		
UBX-SEC-UNIQID	0x27 0x03	• Unique chip ID (Output)
UBX-TIM – Timing messages		
UBX-TIM-TM2	0x0d 0x03	• Time mark data (Periodic/polled)
UBX-TIM-TP	0x0d 0x01	• Time pulse time data (Periodic/polled)
UBX-TIM-VRFY	0x0d 0x06	• Sourced time verification (Periodic/polled)
UBX-UPD – Firmware update messages		
UBX-UPD-SOS	0x09 0x14	• Poll backup restore status (Poll request) • Create backup in flash (Command) • Clear backup in flash (Command) • Backup creation acknowledge (Output) • System restored from backup (Output)

3.9 UBX-ACK (0x05)

The messages in the UBX-ACK class are used to indicate acknowledgement or rejection (i.e. negative acknowledgement) of input messages, such as UBX-CFG messages.

3.9.1 UBX-ACK-ACK (0x05 0x01)

3.9.1.1 Message acknowledged

Message	**UBX-ACK-ACK** **Message acknowledged**				
Type	Output				
Comment	Output upon processing of an input message. A UBX-ACK-ACK is sent as soon as possible but at least within one second.				
Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x05	0x01	2	see below	CK_A CK_B

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	U1	`clsID`	-	-	Class ID of the Acknowledged Message
1	U1	`msgID`	-	-	Message ID of the Acknowledged Message

3.9.2 UBX-ACK-NAK (0x05 0x00)

3.9.2.1 Message not acknowledged

Message	**UBX-ACK-NAK** **Message not acknowledged**				
Type	Output				
Comment	Output upon processing of an input message. A UBX-ACK-NAK is sent as soon as possible but at least within one second.				
Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x05	0x00	2	see below	CK_A CK_B

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	U1	`clsID`	-	-	Class ID of the Not-Acknowledged Message
1	U1	`msgID`	-	-	Message ID of the Not-Acknowledged Message

3.10 UBX-CFG (0x06)

The messages in the UBX-CFG class are used to configure the receiver and poll current configuration values as well as for sending commands to the receiver. Unless stated otherwise, any message in this class sent to the receiver is either acknowledged (by a UBX-ACK-ACK message) if processed successfully or rejected (with a UBX-ACK-NAK message) if processed unsuccessfully.

3.10.1 UBX-CFG-ANT (0x06 0x13)

3.10.1.1 Antenna control settings

Message	**UBX-CFG-ANT** **Antenna control settings**					
Type	Get/set					
Comment	**This message is deprecated in protocol versions greater than 23.01. Use UBX-CFG-VALSET, UBX-CFG-VALGET, UBX-CFG-VALDEL instead.** See the Legacy UBX Message Fields Reference for the corresponding configuration item. This message allows the user to configure the antenna supervisor. The antenna supervisor can be used to detect the status of an active antenna and control it. It can be used to turn off the supply to the antenna in the event of a short cirquit (for example) or to manage power consumption in power save mode. Refer to antenna supervisor configuration in the integration manual for more information regarding the behavior of the antenna supervisor. Note that not all pins can be used for antenna supervisor operation, the default pins are recommended. Consult the integration manual if you need to use the other pins.					
Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x06	0x13	4	see below	CK_A CK_B

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	X2	`flags`	-	-	Antenna flag mask
bit 0	U:1	`svcs`	-	-	Enable antenna supply voltage control signal
bit 1	U:1	`scd`	-	-	Enable short circuit detection
bit 2	U:1	`ocd`	-	-	Enable open circuit detection
bit 3	U:1	`pdwnOnSCD`	-	-	Power down antenna supply if short circuit is detected. (only in combination with bit 1)
bit 4	U:1	`recovery`	-	-	Enable automatic recovery from short state

2	X2	`pins`	-	-	Antenna pin configuration
bits 4…0	U:5	`pinSwitch`	-	-	PIO-pin used for switching antenna supply
bits 9…5	U:5	`pinSCD`	-	-	PIO-pin used for detecting a short in the antenna supply
bits 14…10	U:5	`pinOCD`	-	-	PIO-pin used for detecting open/not connected antenna
bit 15	U:1	`reconfig`	-	-	if set to one, and this command is sent to the receiver, the receiver will reconfigure the pins as specified.

3.10.2 UBX-CFG-CFG (0x06 0x09)

3.10.2.1 Clear, save and load configurations

Message	**UBX-CFG-CFG** **Clear, save and load configurations**
Type	Command
Comment	See Receiver configuration for a detailed description on how receiver configuration should be used. The behavior of this message has changed for protocol versions greater than 23.01. Use UBX-CFG-VALSET and UBX-CFG-VALDEL with the appropriate layers instead. These new messages support selective saving and clearing to retain the behavior removed from this message. The three masks which were used to clear, save and load a subsection of configuration have lost their meaning. It is no longer possible to save or clear a subsection of the configuration using this message. The behavior of the masks is now: • if any bit is set in the clearMask: all configuration in the selected non-volatile memory is deleted • if any bit is set in the saveMask: all current configuration is stored (copied) to the selected layers • if any bit is set in the loadMask: The current configuration is discarded and rebuilt from all the lower layers Note that commands can be combined. The sequence of execution is clear, save, then load. ☞ Old functionality of this message is not available in protocol versions greater than 23.01. Use UBX-CFG-VALSET, UBX-CFG-VALGET, UBX-CFG-VALDEL instead.

Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x06	0x09	12 + [0,1]	*see below*	CK_A CK_B

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	X4	`clearMask`	-	-	Mask for configuration to clear
bits 31…0	U:32	`clearAll`	-	-	Clear all saved configuration from the selected non-volatile memory if any bit is set
4	X4	`saveMask`	-	-	Mask for configuration to save
bits 31…0	U:32	`saveAll`	-	-	Save all current configuration to the selected non-volatile memory if any bit is set
8	X4	`loadMask`	-	-	Mask for configuration to load
bits 31…0	U:32	`loadAll`	-	-	Discard current configuration and rebuilt it from lower non-volatile memory layers if any bit is set
Start of optional group					
12	X1	`deviceMask`	-	-	Mask which selects the memory devices for saving and/or clearing operation Note that if a deviceMask is not provided, the receiver defaults the operation requested to battery-backed RAM (BBR) and Flash (if available)
bit 0	U:1	`devBBR`	-	-	Battery-backed RAM
bit 1	U:1	`devFlash`	-	-	Flash

bit 2	U:1	devEEPROM	-	-	EEPROM (only supported for protocol versions less than 14.00)
bit 4	U:1	devSpiFlash	-	-	SPI Flash (only supported for protocol versions less than 14.00)
End of optional group					

3.10.3 UBX-CFG-DAT (0x06 0x06)

3.10.3.1 Set user-defined datum

Message	**UBX-CFG-DAT** **Set user-defined datum**					
Type	Set					
Comment	**This message is deprecated in protocol versions greater than 23.01. Use** UBX-CFG-VALSET, UBX-CFG-VALGET, UBX-CFG-VALDEL **instead.** See the Legacy UBX Message Fields Reference for the corresponding configuration item.					
Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x06	0x06	44	*see below*	CK_A CK_B

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	R8	majA	-	m	Semi-major axis (accepted range = 6,300,000.0 to 6,500,000.0 meters).
8	R8	flat	-	-	1.0 / flattening (accepted range is 0.0 to 500.0).
16	R4	dX	-	m	X axis shift at the origin (accepted range is +/- 5000.0 meters).
20	R4	dY	-	m	Y axis shift at the origin (accepted range is +/- 5000.0 meters).
24	R4	dZ	-	m	Z axis shift at the origin (accepted range is +/- 5000.0 meters).
28	R4	rotX	-	s	Rotation about the X axis (accepted range is +/- 20.0 milli-arc seconds).
32	R4	rotY	-	s	Rotation about the Y axis (accepted range is +/- 20.0 milli-arc seconds).
36	R4	rotZ	-	s	Rotation about the Z axis (accepted range is +/- 20.0 milli-arc seconds).
40	R4	scale	-	ppm	Scale change (accepted range is 0.0 to 50.0 parts per million).

3.10.3.2 Get currently defined datum

Message	**UBX-CFG-DAT** **Get currently defined datum**					
Type	Get					
Comment	**This message is deprecated in protocol versions greater than 23.01. Use** UBX-CFG-VALSET, UBX-CFG-VALGET, UBX-CFG-VALDEL **instead.** See the Legacy UBX Message Fields Reference for the corresponding configuration item. Returns the parameters of the currently defined datum. If no user-defined datum has been set, this will default to WGS84.					
Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x06	0x06	52	*see below*	CK_A CK_B

Payload description:

Byte offset	Type	Name	Scale	Unit	Description
0	U2	datumNum	-	-	Datum number: 0 = WGS84, 0xFFFF = user-defined (extra values are defined for protocol versions less than 13.00)
2	CH[6]	datumName	-	-	ASCII string: WGS84 or USER (extra values are defined for protocol versions less than 13.00)
8	R8	majA	-	m	Semi-major axis (accepted range = 6,300,000.0 to 6,500,000.0 meters).
16	R8	flat	-	-	1.0 / flattening (accepted range is 0.0 to 500.0).
24	R4	dX	-	m	X axis shift at the origin (accepted range is +/- 5000.0 meters).
28	R4	dY	-	m	Y axis shift at the origin (accepted range is +/- 5000.0 meters).
32	R4	dZ	-	m	Z axis shift at the origin (accepted range is +/- 5000.0 meters).
36	R4	rotX	-	s	Rotation about the X axis (accepted range is +/- 20.0 milli-arc seconds).
40	R4	rotY	-	s	Rotation about the Y axis (accepted range is +/- 20.0 milli-arc seconds).
44	R4	rotZ	-	s	Rotation about the Z axis (accepted range is +/- 20.0 milli-arc seconds).
48	R4	scale	-	ppm	Scale change (accepted range is 0.0 to 50.0 parts per million).

3.10.4 UBX-CFG-DGNSS (0x06 0x70)

3.10.4.1 DGNSS configuration

Message	**UBX-CFG-DGNSS** **DGNSS configuration**
Type	Get/set
Comment	**This message is deprecated in protocol versions greater than 23.01. Use UBX-CFG-VALSET, UBX-CFG-VALGET, UBX-CFG-VALDEL instead.** See the Legacy UBX Message Fields Reference for the corresponding configuration item. This message allows the user to configure the DGNSS configuration of the receiver.

Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x06	0x70	4	*see below*	CK_A CK_B

Payload description:

Byte offset	Type	Name	Scale	Unit	Description
0	U1	dgnssMode	-	-	Specifies differential mode: • 2 = RTK float: No attempts are made to fix ambiguities. • 3 = RTK fixed: Ambiguities are fixed whenever possible.
1	U1[3]	reserved0	-	-	Reserved

3.10.5 UBX-CFG-GEOFENCE (0x06 0x69)

3.10.5.1 Geofencing configuration

Message	**UBX-CFG-GEOFENCE** **Geofencing configuration**
Type	Get/set
Comment	**This message is deprecated in protocol versions greater than 23.01. Use UBX-CFG-VALSET, UBX-CFG-VALGET, UBX-CFG-VALDEL instead.** See the Legacy UBX Message Fields Reference for the corresponding configuration item. Gets or sets the geofencing configuration. If the receiver is sent a valid new configuration, it will respond with a UBX-ACK-ACK message and immediately change to the new configuration. Otherwise the receiver will reject the request, by issuing a UBX-ACK-NAK and continuing operation with the previous configuration. Note that the acknowledge message does not indicate whether the PIO configuration has been successfully applied (pin assigned), it only indicates the successful configuration of the feature. The configured PIO must be previously unoccupied for successful assignment.

Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x06	0x69	8 + numFences·12	*see below*	CK_A CK_B

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	U1	`version`	-	-	Message version (0x00 for this version)
1	U1	`numFences`	-	-	Number of geofences contained in this message. Note that the receiver can only store a limited number of geofences (currently 4).
2	U1	`confLvl`	-	-	Required confidence level for state evaluation. This value times the position's standard deviation (sigma) defines the confidence band. • 0 = no confidence required • 1 = 68% • 2 = 95% • 3 = 99.7% • 4 = 99.99%
3	U1	`reserved0`	-	-	Reserved
4	U1	`pioEnabled`	-	-	1 = Enable PIO combined fence state output, 0 = disable
5	U1	`pinPolarity`	-	-	PIO pin polarity. 0 = Low means inside, 1 = Low means outside. Unknown state is always high.
6	U1	`pin`	-	-	PIO pin number
7	U1	`reserved1`	-	-	Reserved
Start of repeated group (`numFences` times)					
8 + n·12	I4	`lat`	1e-7	deg	Latitude of the geofence circle center
12 + n·12	I4	`lon`	1e-7	deg	Longitude of the geofence circle center
16 + n·12	U4	`radius`	1e-2	m	Radius of the geofence circle
End of repeated group (`numFences` times)					

3.10.6 UBX-CFG-GNSS (0x06 0x3e)

3.10.6.1 GNSS system configuration

Message	**UBX-CFG-GNSS** **GNSS system configuration**
Type	Get/set

Comment	**This message is deprecated in protocol versions greater than 23.01. Use** UBX-CFG-VALSET, UBX-CFG-VALGET, UBX-CFG-VALDEL **instead.** See the Legacy UBX Message Fields Reference for the corresponding configuration item. Gets or sets the GNSS system channel sharing configuration. If the receiver is sent a valid new configuration, it will respond with a UBX-ACK-ACK message and immediately change to the new configuration. Otherwise the receiver will reject the request, by issuing a UBX-ACK-NAK and continuing operation with the previous configuration. Configuration requirements: • It is necessary for at least one major GNSS to be enabled, after applying the new configuration to the current one. • It is also required that at least 4 tracking channels are available to each enabled major GNSS, i.e. `maxTrkCh` must have a minimum value of 4 for each enabled major GNSS. • The number of tracking channels in use must not exceed the number of tracking channels available in hardware, and the sum of all reserved tracking channels needs to be less than or equal to the number of tracking channels in use. Notes: • To avoid cross-correlation issues, it is recommended that GPS and QZSS are always both enabled or both disabled. • Polling this message returns the configuration of all supported GNSS, whether enabled or not; it may also include GNSS unsupported by the particular product, but in such cases the enable flag will always be unset. • See section Satellite Numbering for a description of the GNSS IDs available. • Configuration specific to the GNSS system can be done via other messages (e.g. **UBX-CFG-SBAS**).

Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x06	0x3e	4 + numConfigBlocks·8	*see below*	CK_A CK_B

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	U1	`msgVer`	-	-	Message version (0x00 for this version)
1	U1	`numTrkChHw`	-	-	Number of tracking channels available in hardware (read only)
2	U1	`numTrkChUse`	-	-	(Read only for protocol versions greater than 23.00) Number of tracking channels to use. Must be > 0, <= numTrkChHw. If 0xFF, then number of tracking channels to use will be set to numTrkChHw.
3	U1	`numConfigBlocks`	-	-	Number of configuration blocks following
Start of repeated group (`numConfigBlocks` times)					
4 + n·8	U1	`gnssId`	-	-	System identifier (see Satellite Numbering)
5 + n·8	U1	`resTrkCh`	-	-	(Read only for protocol versions greater than 23.00) Number of reserved (minimum) tracking channels for this system.
6 + n·8	U1	`maxTrkCh`	-	-	(Read only for protocol versions greater than 23.00) Maximum number of tracking channels used for this system. Must be > 0, >= resTrkChn, <= numTrkChUse and <= maximum number of tracking channels supported for this system.
7 + n·8	U1	`reserved0`	-	-	Reserved
8 + n·8	X4	`flags`	-	-	Bitfield of flags. At least one signal must be configured in every enabled system.
bit 0	U:1	`enable`	-	-	Enable this system
bits 23…16	U:8	`sigCfgMask`	-	-	Signal configuration mask When gnssId is 0 (GPS) • 0x01 = GPS L1C/A

Byte offset	Type	Name	Scale	Unit	Description
					• 0x10 = GPS L2C • 0x20 = GPS L5 When gnssId is 1 (SBAS) • 0x01 = SBAS L1C/A When gnssId is 2 (Galileo) • 0x01 = Galileo E1 (not supported for protocol versions less than 18.00) • 0x10 = Galileo E5a • 0x20 = Galileo E5b When gnssId is 3 (BeiDou) • 0x01 = BeiDou B1I • 0x10 = BeiDou B2I • 0x80 = BeiDou B2A When gnssId is 4 (IMES) • 0x01 = IMES L1 When gnssId is 5 (QZSS) • 0x01 = QZSS L1C/A • 0x04 = QZSS L1S • 0x10 = QZSS L2C • 0x20 = QZSS L5 When gnssId is 6 (GLONASS) • 0x01 = GLONASS L1 • 0x10 = GLONASS L2
End of repeated group (`numConfigBlocks` times)					

3.10.7 UBX-CFG-INF (0x06 0x02)

3.10.7.1 Poll configuration for one protocol

Message	**UBX-CFG-INF** **Poll configuration for one protocol**
Type	Poll request
Comment	**This message is deprecated in protocol versions greater than 23.01. Use** UBX-CFG-VALSET, UBX-CFG-VALGET, UBX-CFG-VALDEL **instead.** See the Legacy UBX Message Fields Reference for the corresponding configuration item.

Message structure	Header	Class	ID	Length (Bytes)	Payload	Checksum
	0xb5 0x62	0x06	0x02	1	*see below*	CK_A CK_B

Payload description:

Byte offset	Type	Name	Scale	Unit	Description
0	U1	`protocolID`	-	-	Protocol identifier, identifying the output protocol for this poll request. The following are valid protocol identifiers: • 0: UBX protocol • 1: NMEA protocol • 2-255: Reserved

3.10.7.2 Information message configuration

Message	**UBX-CFG-INF** **Information message configuration**
Type	Get/set
Comment	**This message is deprecated in protocol versions greater than 23.01. Use** UBX-CFG-VALSET, UBX-CFG-VALGET, UBX-CFG-VALDEL **instead.**

The value of infMsgMask[x] below is formed so that each bit represents one of the INF class messages (bit 0 for ERROR, bit 1 for WARNING and so on). For a complete list, see the Message class INF. Several configurations can be concatenated to one input message. In this case the payload length can be a multiple of the normal length. Output messages from the module contain only one configuration unit.

Note that:

- I/O ports 1 and 2 correspond to serial ports 1 and 2.
- I/O port 0 is I2C (DDC).
- I/O port 3 is USB.
- I/O port 4 is SPI.
- I/O port 5 is reserved for future use.

Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x06	0x02	[0..n]·10	*see below*	CK_A CK_B

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
Start of repeated group (N times)					
0 + n·10	U1	`protocolID`	-	-	Protocol identifier, identifying for which protocol the configuration is set/get. The following are valid protocol identifiers: • 0: UBX protocol • 1: NMEA protocol • 2-255: Reserved
1 + n·10	U1[3]	`reserved0`	-	-	Reserved
4 + n·10	X1[6]	`infMsgMask`	-	-	A bit mask, saying which information messages are enabled on each I/O port
bit 0	U:1	`ERROR`	-	-	enable ERROR
bit 1	U:1	`WARNING`	-	-	enable WARNING
bit 2	U:1	`NOTICE`	-	-	enable NOTICE
bit 3	U:1	`TEST`	-	-	enable TEST
bit 4	U:1	`DEBUG`	-	-	enable DEBUG
End of repeated group (N times)					

3.10.8 UBX-CFG-ITFM (0x06 0x39)

3.10.8.1 Jamming/interference monitor configuration

Message	**UBX-CFG-ITFM** **Jamming/interference monitor configuration**
Type	Get/set
Comment	**This message is deprecated in protocol versions greater than 23.01. Use UBX-CFG-VALSET, UBX-CFG-VALGET, UBX-CFG-VALDEL instead.** See the Legacy UBX Message Fields Reference for the corresponding configuration item.

Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x06	0x39	8	*see below*	CK_A CK_B

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	X4	`config`	-	-	Interference config word
bits 3…0	U:4	`bbThreshold`	-	-	Broadband jamming detection threshold
bits 8…4	U:5	`cwThreshold`	-	-	CW jamming detection threshold
bits 30…9	U:22	`algorithmBits`	-	-	Reserved algorithm settings - should be set to 0x16B156 in hex for correct settings

bit 31	U:1	enable	-	-	Enable interference detection
4	X4	config2	-	-	Extra settings for jamming/interference monitor
bits 11…0	U:12	generalBits	-	-	General settings - should be set to 0x31E in hex for correct setting
bits 13…12	U:2	antSetting	-	-	Antenna setting, 0=unknown, 1=passive, 2=active
bit 14	U:1	enable2	-	-	Set to 1 to scan auxiliary bands (u-blox 8 / u-blox M8 only, otherwise ignored)

3.10.9 UBX-CFG-LOGFILTER (0x06 0x47)

3.10.9.1 Data logger configuration

Message	**UBX-CFG-LOGFILTER** **Data logger configuration**
Type	Get/set
Comment	**This message is deprecated in protocol versions greater than 23.01. Use UBX-CFG-VALSET, UBX-CFG-VALGET, UBX-CFG-VALDEL instead.** See the Legacy UBX Message Fields Reference for the corresponding configuration item. This message can be used to configure the data logger, i.e. to enable/disable the log recording and to get/set the position entry filter settings. Position entries can be filtered based on time difference, position difference or current speed thresholds. Position and speed filtering also have a minimum time interval. A position is logged if any of the thresholds are exceeded. If a threshold is set to zero it is ignored. The maximum rate of position logging is 1 Hz. The filter settings will be configured to the provided values only if the 'applyAllFilterSettings' flag is set. This allows the recording to be enabled/disabled independently of configuring the filter settings. Configuring the data logger in the absence of a logging file is supported. By doing so, once the logging file is created, the data logger configuration will take effect immediately and logging recording and filtering will activate according to the configuration.

Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x06	0x47	12	*see below*	CK_A CK_B

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	U1	version	-	-	Message version (0x01 for this version)
1	X1	flags	-	-	Flags
bit 0	U:1	recordEnabled	-	-	1 = enable recording, 0 = disable recording
bit 1	U:1	psmOncePerWakupEnabled	-	-	1 = enable recording only one single position per PSM on/off mode wake-up period, 0 = disable once per wake-up
bit 2	U:1	applyAllFilterSettings	-	-	1 = apply all filter settings, 0 = only apply recordEnabled
2	U2	minInterval	-	s	Minimum time interval between logged positions (0 = not set). **This is only applied in combination with the speed and/or position thresholds.** If both minInterval and timeThreshold are set, minInterval must be less than or equal to timeThreshold.
4	U2	timeThreshold	-	s	If the time difference is greater than the threshold, then the position is logged (0 = not set).
6	U2	speedThreshold	-	m/s	If the current speed is greater than the threshold, then the position is logged (0 = not set). minInterval also applies.

8	U4	`positionThreshold`	-	m	If the 3D position difference is greater than the threshold, then the position is logged (0 = not set). minInterval also applies.

3.10.10 UBX-CFG-MSG (0x06 0x01)

3.10.10.1 Poll a message configuration

Message	**UBX-CFG-MSG** **Poll a message configuration**
Type	Poll request
Comment	**This message is deprecated in protocol versions greater than 23.01. Use UBX-CFG-VALSET, UBX-CFG-VALGET, UBX-CFG-VALDEL instead.** See the Legacy UBX Message Fields Reference for the corresponding configuration item.

Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x06	0x01	2	*see below*	CK_A CK_B

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	U1	`msgClass`	-	-	Message class
1	U1	`msgID`	-	-	Message identifier

3.10.10.2 Set message rate(s)

Message	**UBX-CFG-MSG** **Set message rate(s)**
Type	Get/set
Comment	**This message is deprecated in protocol versions greater than 23.01. Use UBX-CFG-VALSET, UBX-CFG-VALGET, UBX-CFG-VALDEL instead.** See the Legacy UBX Message Fields Reference for the corresponding configuration item. Get/set message rate configuration (s) to/from the receiver. • Send rate is relative to the event a message is registered on. For example, if the rate of a navigation message is set to 2, the message is sent every second navigation solution. For configuring NMEA messages, the section NMEA Messages Overview describes class and identifier numbers used.

Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x06	0x01	8	*see below*	CK_A CK_B

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	U1	`msgClass`	-	-	Message class
1	U1	`msgID`	-	-	Message identifier
2	U1[6]	`rate`	-	-	Send rate on I/O port (6 ports)

3.10.10.3 Set message rate

Message	**UBX-CFG-MSG** **Set message rate**
Type	Get/set
Comment	**This message is deprecated in protocol versions greater than 23.01. Use UBX-CFG-VALSET, UBX-CFG-VALGET, UBX-CFG-VALDEL instead.** See the Legacy UBX Message Fields Reference for the corresponding configuration item. Set message rate configuration for the current port.

Message structure	Header	Class	ID	Length (Bytes)	Payload	Checksum
	0xb5 0x62	0x06	0x01	3	see below	CK_A CK_B

Payload description:

Byte offset	Type	Name	Scale	Unit	Description
0	U1	msgClass	-	-	Message class
1	U1	msgID	-	-	Message identifier
2	U1	rate	-	-	Send rate on current port

3.10.11 UBX-CFG-NAV5 (0x06 0x24)

3.10.11.1 Navigation engine settings

Message	**UBX-CFG-NAV5** **Navigation engine settings**
Type	Get/set
Comment	**This message is deprecated in protocol versions greater than 23.01. Use** UBX-CFG-VALSET, UBX-CFG-VALGET, UBX-CFG-VALDEL **instead.** See the Legacy UBX Message Fields Reference for the corresponding configuration item.

Message structure	Header	Class	ID	Length (Bytes)	Payload	Checksum
	0xb5 0x62	0x06	0x24	36	see below	CK_A CK_B

Payload description:

Byte offset	Type	Name	Scale	Unit	Description
0	X2	mask	-	-	Parameters bitmask. Only the masked parameters will be applied.
bit 0	U:1	dyn	-	-	Apply dynamic model settings
bit 1	U:1	minEl	-	-	Apply minimum elevation settings
bit 2	U:1	posFixMode	-	-	Apply fix mode settings
bit 3	U:1	drLim	-	-	Reserved (apply DR limit settings, only applicable for protocol versions less than 14.00)
bit 4	U:1	posMask	-	-	Apply position mask settings
bit 5	U:1	timeMask	-	-	Apply time mask settings
bit 6	U:1	staticHoldMask	-	-	Apply static hold settings
bit 7	U:1	dgpsMask	-	-	Apply DGPS settings (not supported for protocol versions less than 13.00)
bit 8	U:1	cnoThreshold	-	-	Apply CNO threshold settings (cnoThresh, cnoThreshNumSVs) (not supported for protocol versions less than 14.00)
bit 10	U:1	utc	-	-	Apply UTC settings (not supported for protocol versions less than 16.00)

2	U1	dynModel	-	-	Dynamic platform model: • 0 = portable • 2 = stationary • 3 = pedestrian • 4 = automotive • 5 = sea • 6 = airborne with <1g acceleration • 7 = airborne with <2g acceleration • 8 = airborne with <4g acceleration • 9 = wrist-worn watch (not supported for protocol versions less than 18.00) • 10 = motorbike (supported for protocol versions 19.20, and 35.10) • 11 = robotic lawn mower (supported for protocol versions 33.21) • 12 = electric kick scooter (supported for protocol versions 33.21, and 35.10)
3	U1	fixMode	-	-	Position fixing mode: • 1 = 2D only • 2 = 3D only • 3 = auto 2D/3D
4	I4	fixedAlt	0.01	m	Fixed altitude (mean sea level) for 2D fix mode
8	U4	fixedAltVar	0.0001	m^2	Fixed altitude variance for 2D mode
12	I1	minElev	-	deg	Minimum elevation for a GNSS satellite to be used in NAV
13	U1	drLimit	-	s	Reserved (maximum time to perform dead reckoning (linear extrapolation) in case of GPS signal loss, only applicable for protocol versions less than 14.00)
14	U2	pDop	0.1	-	Position DOP mask to use
16	U2	tDop	0.1	-	Time DOP mask to use
18	U2	pAcc	-	m	Position accuracy mask
20	U2	tAcc	-	m	Time accuracy mask
22	U1	staticHoldThresh	-	cm/s	Static hold threshold
23	U1	dgnssTimeout	-	s	DGNSS timeout (not supported for protocol versions less than 13.00)
24	U1	cnoThreshNumSVs	-	-	Number of satellites required to have C/N0 above cnoThresh for a fix to be attempted (not supported for protocol versions less than 14.00)
25	U1	cnoThresh	-	dBHz	C/N0 threshold for deciding whether to attempt a fix (not supported for protocol versions less than 14.00)
26	U1[2]	reserved0	-	-	Reserved
28	U2	staticHoldMaxDist	-	m	Static hold distance threshold (before quitting static hold) (not supported for protocol versions less than 15.00)

30	U1	utcStandard	-	-	UTC standard to be used (see GNSS time bases section in the integration manual): • 0 = Automatic; receiver selects based on GNSS configuration • 3 = UTC as operated by the U.S. Naval Observatory (USNO); derived from GPS time • 5 = UTC as combined from multiple European laboratories; derived from Galileo time • 6 = UTC as operated by the former Soviet Union (SU); derived from GLONASS time • 7 = UTC as operated by the National Time Service Center (NTSC), China; derived from BeiDou time • 8 = UTC as operated by the National Physics Laboratory, India (NPLI); derived from NavIC time (not supported for protocol versions less than 16.00)
31	U1[5]	reserved1	-	-	Reserved

3.10.12 UBX-CFG-NAVX5 (0x06 0x23)

3.10.12.1 Navigation engine expert settings

Message	**UBX-CFG-NAVX5** **Navigation engine expert settings**					
Type	Get/set					
Comment	**This message is deprecated in protocol versions greater than 23.01. Use** UBX-CFG-VALSET, UBX-CFG-VALGET, UBX-CFG-VALDEL **instead.** See the Legacy UBX Message Fields Reference for the corresponding configuration item.					
Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x06	0x23	40	*see below*	CK_A CK_B

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	U2	version	-	-	Message version (0x0002 for this version)
2	X2	mask1	-	-	First parameters bitmask. Only the flagged parameters will be applied, unused bits must be set to 0.
bit 2	U:1	minMax	-	-	1 = apply min/max SVs settings
bit 3	U:1	minCno	-	-	1 = apply minimum C/N0 setting
bit 6	U:1	initial3dfix	-	-	1 = apply initial 3D fix settings
bit 9	U:1	wknRoll	-	-	1 = apply GPS weeknumber rollover settings
bit 10	U:1	ackAid	-	-	1 = apply assistance acknowledgement settings
bit 13	U:1	ppp	-	-	1 = apply usePPP flag
bit 14	U:1	aop	-	-	1 = apply aopCfg (useAOP flag) and aopOrbMaxErr settings (AssistNow Autonomous)
4	X4	mask2	-	-	Second parameters bitmask. Only the flagged parameters will be applied, unused bits must be set to 0.
bit 6	U:1	adr	-	-	Apply ADR/UDR sensor fusion on/off setting (useAdr flag)
bit 7	U:1	sigAttenComp	-	-	Only supported on certain products
8	U1[2]	reserved0	-	-	Reserved
10	U1	minSVs	-	#SVs	Minimum number of satellites for navigation

11	U1	maxSVs	-	#SVs	Maximum number of satellites for navigation
12	U1	minCNO	-	dBHz	Minimum satellite signal level for navigation
13	U1	reserved1	-	-	Reserved
14	U1	iniFix3D	-	-	1 = initial fix must be 3D
15	U1[2]	reserved2	-	-	Reserved
17	U1	ackAiding	-	-	1 = issue acknowledgements for assistance message input
18	U2	wknRollover	-	-	GPS week rollover number; GPS week numbers will be set correctly from this week up to 1024 weeks after this week. Setting this to 0 reverts to firmware default.
20	U1	sigAttenCompMode	-	dBHz	Only supported on certain products
21	U1	reserved3	-	-	Reserved
22	U1[2]	reserved4	-	-	Reserved
24	U1[2]	reserved5	-	-	Reserved
26	U1	usePPP	-	-	1 = use Precise Point Positioning (only available with the PPP product variant)
27	U1	aopCfg	-	-	AssistNow Autonomous configuration
bit 0	U:1	useAOP	-	-	1 = enable AssistNow Autonomous
28	U1[2]	reserved6	-	-	Reserved
30	U2	aopOrbMaxErr	-	m	Maximum acceptable (modeled) AssistNow Autonomous orbit error (valid range = 5..1000, or 0 = reset to firmware default)
32	U1[4]	reserved7	-	-	Reserved
36	U1[3]	reserved8	-	-	Reserved
39	U1	useAdr	-	-	Only supported on certain products

3.10.13 UBX-CFG-NMEA (0x06 0x17)

3.10.13.1 Extended NMEA protocol configuration V1

Message	**UBX-CFG-NMEA** **Extended NMEA protocol configuration V1**
Type	Get/set
Comment	**This message is deprecated in protocol versions greater than 23.01. Use UBX-CFG-VALSET, UBX-CFG-VALGET, UBX-CFG-VALDEL instead.** Get/set the NMEA protocol configuration. See section NMEA Protocol Configuration for a detailed description of the configuration effects on NMEA output. See the Legacy UBX Message Fields Reference for the corresponding configuration item.

	Header	Class	ID	Length (Bytes)	Payload	Checksum
Message structure	0xb5 0x62	0x06	0x17	20	*see below*	CK_A CK_B

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	X1	filter	-	-	filter flags
bit 0	U:1	posFilt	-	-	Enable position output for failed or invalid fixes
bit 1	U:1	mskPosFilt	-	-	Enable position output for invalid fixes
bit 2	U:1	timeFilt	-	-	Enable time output for invalid times

	bit 3	U:1	dateFilt	-	-	Enable date output for invalid dates
	bit 4	U:1	gpsOnlyFilter	-	-	Restrict output to GPS satellites only
	bit 5	U:1	trackFilt	-	-	Enable COG output even if COG is frozen
1		U1	nmeaVersion	-	-	• 0x4b = NMEA version 4.11 (not available in all products) • 0x41 = NMEA version 4.10 (not available in all products) • 0x40 = NMEA version 4.0 (not available in all products) • 0x23 = NMEA version 2.3 • 0x21 = NMEA version 2.1
2		U1	numSV	-	-	Maximum number of SVs to report per TalkerId. • 0 = unlimited • 8 = 8 SVs • 12 = 12 SVs • 16 = 16 SVs
3		X1	flags	-	-	flags
	bit 0	U:1	compat	-	-	enable compatibility mode. This might be needed for certain applications when customer's NMEA parser expects a fixed number of digits in position coordinates.
	bit 1	U:1	consider	-	-	enable considering mode.
	bit 2	U:1	limit82	-	-	enable strict limit to 82 characters maximum.
	bit 3	U:1	highPrec	-	-	enable high precision mode. This flag cannot be set in conjunction with either compatibility mode or Limit82 mode (not supported for protocol versions less than 20.01).
4		X4	gnssToFilter	-	-	Filters out satellites based on their GNSS. If a bitfield is enabled, the corresponding satellites will be not output.
	bit 0	U:1	gps	-	-	Disable reporting of GPS satellites
	bit 1	U:1	sbas	-	-	Disable reporting of SBAS satellites
	bit 2	U:1	galileo	-	-	Disable reporting of Galileo satellites
	bit 4	U:1	qzss	-	-	Disable reporting of QZSS satellites
	bit 5	U:1	glonass	-	-	Disable reporting of GLONASS satellites
	bit 6	U:1	beidou	-	-	Disable reporting of BeiDou satellites
8		U1	svNumbering	-	-	Configures the display of satellites that do not have an NMEA-defined value. Note: this does not apply to satellites with an unknown ID. • 0 = Strict - Satellites are not output • 1 = Extended - Use proprietary numbering (see Satellite Numbering)

9	U1	mainTalkerId	-	-	By default the main Talker ID (i.e. the Talker ID used for all messages other than GSV) is determined by the GNSS assignment of the receiver's channels (see UBX-CFG-GNSS). This field enables the main Talker ID to be overridden. • 0 = Main Talker ID is not overridden • 1 = Set main Talker ID to 'GP' • 2 = Set main Talker ID to 'GL' • 3 = Set main Talker ID to 'GN' • 4 = Set main Talker ID to 'GA' (not supported for protocol versions less than 15.00) • 5 = Set main Talker ID to 'GB' (not supported for protocol versions less than 15.00) • 6 = Set main Talker ID to 'GQ' (available in NMEA 4.11 and later)
10	U1	gsvTalkerId	-	-	By default the Talker ID for GSV messages is GNSS-specific (as defined by NMEA). This field enables the GSV Talker ID to be overridden. • 0 = Use GNSS-specific Talker ID (as defined by NMEA) • 1 = Use the main Talker ID
11	U1	version	-	-	Message version (0x01 for this version)
12	CH[2]	bdsTalkerId	-	-	Sets the two characters that should be used for the BeiDou Talker ID. If these are set to zero, then the default BeiDou Talker ID will be used.
14	U1[6]	reserved0	-	-	Reserved

3.10.14 UBX-CFG-ODO (0x06 0x1e)

3.10.14.1 Odometer, low-speed COG engine settings

Message	**UBX-CFG-ODO** **Odometer, low-speed COG engine settings**
Type	Get/set
Comment	**This message is deprecated in protocol versions greater than 23.01. Use UBX-CFG-VALSET, UBX-CFG-VALGET, UBX-CFG-VALDEL instead.** See the Legacy UBX Message Fields Reference for the corresponding configuration item. ☞ This feature is not supported for the FTS product variant.

Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x06	0x1e	20	*see below*	CK_A CK_B

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	U1	version	-	-	Message version (0x00 for this version)
1	U1[3]	reserved0	-	-	Reserved
4	U1	flags	-	-	Odometer/Low-speed COG filter flags
bit 0	U:1	useODO	-	-	Odometer-enabled flag
bit 1	U:1	useCOG	-	-	Low-speed COG filter enabled flag
bit 2	U:1	outLPVel	-	-	Output low-pass filtered velocity flag
bit 3	U:1	outLPCog	-	-	Output low-pass filtered heading (COG) flag
5	X1	odoCfg	-	-	Odometer filter settings

bits 2...0	U:3	`profile`	-	-	Profile type (0=running, 1=cycling, 2=swimming, 3=car, 4=custom)
6	U1[6]	`reserved1`	-	-	Reserved
12	U1	`cogMaxSpeed`	1e-1	m/s	Speed below which course-over-ground (COG) is computed with the low-speed COG filter
13	U1	`cogMaxPosAcc`	-	m	Maximum acceptable position accuracy for computing COG with the low-speed COG filter
14	U1[2]	`reserved2`	-	-	Reserved
16	U1	`velLpGain`	-	-	Velocity low-pass filter level, range 0..255
17	U1	`cogLpGain`	-	-	COG low-pass filter level (at speed < 8 m/s), range 0..255
18	U1[2]	`reserved3`	-	-	Reserved

3.10.15 UBX-CFG-PRT (0x06 0x00)

3.10.15.1 Polls the configuration for one I/O port

Message	**UBX-CFG-PRT** **Polls the configuration for one I/O port**
Type	Poll request
Comment	**This message is deprecated in protocol versions greater than 23.01. Use UBX-CFG-VALSET, UBX-CFG-VALGET, UBX-CFG-VALDEL instead.** See the Legacy UBX Message Fields Reference for the corresponding configuration item. Sending this message with a port ID as payload results in having the receiver return the configuration for the specified port.

Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x06	0x00	1	*see below*	CK_A CK_B

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	U1	`PortID`	-	-	Port identifier number (see the other versions of CFG-PRT for valid values)

3.10.15.2 Port configuration for UART ports

Message	**UBX-CFG-PRT** **Port configuration for UART ports**
Type	Get/set
Comment	**This message is deprecated in protocol versions greater than 23.01. Use UBX-CFG-VALSET, UBX-CFG-VALGET, UBX-CFG-VALDEL instead.** See the Legacy UBX Message Fields Reference for the corresponding configuration item. Several configurations can be concatenated to one input message. In this case the payload length can be a multiple of the normal length (see the other versions of CFG-PRT). Output messages from the module contain only one configuration unit. Note that this message can affect baud rate and other transmission parameters. Because there may be messages queued for transmission there may be uncertainty about which protocol applies to such messages. In addition a message currently in transmission may be corrupted by a protocol change. Host data reception parameters may have to be changed to be able to receive future messages, including the acknowledge message resulting from the CFG-PRT message.

Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x06	0x00	20	*see below*	CK_A CK_B

Payload description:

Byte offset	Type	Name	Scale	Unit	Description
0	U1	portID	-	-	Port identifier number (see the integration manual for valid UART port IDs)
1	U1	reserved0	-	-	Reserved
2	X2	txReady	-	-	TX ready PIN configuration (not supported for protocol versions less than 13.01)
bit 0	U:1	en	-	-	Enable TX ready feature for this port
bit 1	U:1	pol	-	-	Polarity • 0 High-active • 1 Low-active
bits 6...2	U:5	pin	-	-	PIO to be used (must not be in use by another function)
bits 15...7	U:9	thres	-	-	Threshold The given threshold is multiplied by 8 bytes. The TX ready PIN goes active after >= thres*8 bytes are pending for the port and going inactive after the last pending bytes have been written to hardware (0-4 bytes before end of stream). • 0x000 no threshold • 0x001 8byte • 0x002 16byte • ... • 0x1FE 4080byte • 0x1FF 4088byte
4	X4	mode	-	-	A bit mask describing the UART mode
bits 7...6	U:2	charLen	-	-	Character length • 00 5bit (not supported) • 01 6bit (not supported) • 10 7bit (supported only with parity) • 11 8bit
bits 11...9	U:3	parity	-	-	• 000 Even parity • 001 Odd parity • 10X No parity • X1X Reserved
bits 13...12	U:2	nStopBits	-	-	Number of Stop bits • 00 1 Stop bit • 01 1.5 Stop bit • 10 2 Stop bit • 11 0.5 Stop bit
8	U4	baudRate	-	Bits/s	Baud rate in bits/second
12	X2	inProtoMask	-	-	A mask describing which input protocols are active. Each bit of this mask is used for a protocol. Through that, multiple protocols can be defined on a single port.
bit 0	U:1	inUbx	-	-	UBX protocol
bit 1	U:1	inNmea	-	-	NMEA protocol
bit 2	U:1	inRtcm	-	-	RTCM2 protocol
bit 5	U:1	inRtcm3	-	-	RTCM3 protocol (not supported for protocol versions less than 20.00)
14	X2	outProtoMask	-	-	A mask describing which output protocols are active. Each bit of this mask is used for a protocol. Through that, multiple protocols can be defined on a single port.
bit 0	U:1	outUbx	-	-	UBX protocol

bit 1	U:1	outNmea	-	-	NMEA protocol
bit 5	U:1	outRtcm3	-	-	RTCM3 protocol (not supported for protocol versions less than 20.00)
16	X2	flags	-	-	Flags bit mask
bit 1	U:1	extendedTxTimeout	-	-	Extended TX timeout: if set, the port will time out if allocated TX memory >=4 kB and no activity for 1.5 s. If not set the port will time out if no activity for 1.5 s regardless on the amount of allocated TX memory (not supported for protocol versions less than 13.01).
18	U1[2]	reserved1	-	-	Reserved

3.10.15.3 Port configuration for USB port

Message	**UBX-CFG-PRT** **Port configuration for USB port**
Type	Get/set
Comment	**This message is deprecated in protocol versions greater than 23.01. Use UBX-CFG-VALSET, UBX-CFG-VALGET, UBX-CFG-VALDEL instead.** See the Legacy UBX Message Fields Reference for the corresponding configuration item. Several configurations can be concatenated to one input message. In this case the payload length can be a multiple of the normal length (see the other versions of CFG-PRT). Output messages from the module contain only one configuration unit.

Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x06	0x00	20	*see below*	CK_A CK_B

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	U1	portID	-	-	Port identifier number (= 3 for USB port)
1	U1	reserved0	-	-	Reserved
2	X2	txReady	-	-	TX ready PIN configuration (not supported for protocol versions less than 13.01)
bit 0	U:1	en	-	-	Enable TX ready feature for this port
bit 1	U:1	pol	-	-	Polarity • 0 High-active • 1 Low-active
bits 6…2	U:5	pin	-	-	PIO to be used (must not be in use by another function)
bits 15…7	U:9	thres	-	-	Threshold The given threshold is multiplied by 8 bytes. The TX ready PIN goes active after >= thres*8 bytes are pending for the port and going inactive after the last pending bytes have been written to hardware (0-4 bytes before end of stream). • 0x000 no threshold • 0x001 8byte • 0x002 16byte • ... • 0x1FE 4080byte • 0x1FF 4088byte
4	U1[8]	reserved1	-	-	Reserved
12	X2	inProtoMask	-	-	A mask describing which input protocols are active. Each bit of this mask is used for a protocol. Through that, multiple protocols can be defined on a single port.

bit 0	U:1	inUbx	-	-	UBX protocol
bit 1	U:1	inNmea	-	-	NMEA protocol
bit 2	U:1	inRtcm	-	-	RTCM2 protocol
bit 5	U:1	inRtcm3	-	-	RTCM3 protocol (not supported for protocol versions less than 20.00)
14	X2	outProtoMask	-	-	A mask describing which output protocols are active. Each bit of this mask is used for a protocol. Through that, multiple protocols can be defined on a single port.
bit 0	U:1	outUbx	-	-	UBX protocol
bit 1	U:1	outNmea	-	-	NMEA protocol
bit 5	U:1	outRtcm3	-	-	RTCM3 protocol (not supported for protocol versions less than 20.00)
16	U1[2]	reserved2	-	-	Reserved
18	U1[2]	reserved3	-	-	Reserved

3.10.15.4 Port configuration for SPI port

Message	**UBX-CFG-PRT** **Port configuration for SPI port**
Type	Get/set
Comment	**This message is deprecated in protocol versions greater than 23.01. Use UBX-CFG-VALSET, UBX-CFG-VALGET, UBX-CFG-VALDEL instead.** See the Legacy UBX Message Fields Reference for the corresponding configuration item. Several configurations can be concatenated to one input message. In this case the payload length can be a multiple of the normal length. Output messages from the module contain only one configuration unit.

Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x06	0x00	20	*see below*	CK_A CK_B

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	U1	portID	-	-	Port identifier number (= 4 for SPI port)
1	U1	reserved0	-	-	Reserved
2	X2	txReady	-	-	TX ready PIN configuration (not supported for protocol versions less than 13.01)
bit 0	U:1	en	-	-	Enable TX ready feature for this port
bit 1	U:1	pol	-	-	Polarity • 0 High-active • 1 Low-active
bits 6…2	U:5	pin	-	-	PIO to be used (must not be in use by another function)
bits 15…7	U:9	thres	-	-	Threshold The given threshold is multiplied by 8 bytes. The TX ready PIN goes active after >= thres*8 bytes are pending for the port and going inactive after the last pending bytes have been written to hardware (0-4 bytes before end of stream). • 0x000 no threshold • 0x001 8byte • 0x002 16byte • ... • 0x1FE 4080byte • 0x1FF 4088byte

4	X4	`mode`	-	-	SPI Mode Flags
bits 2...1	U:2	`spiMode`	-	-	• 00 SPI Mode 0: CPOL = 0, CPHA = 0 • 01 SPI Mode 1: CPOL = 0, CPHA = 1 • 10 SPI Mode 2: CPOL = 1, CPHA = 0 • 11 SPI Mode 3: CPOL = 1, CPHA = 1
bits 13...8	U:6	`ffCnt`	-	-	Number of bytes containing 0xFF to receive before switching off reception. Range: 0 (mechanism off) - 63
8	U1[4]	`reserved1`	-	-	Reserved
12	X2	`inProtoMask`	-	-	A mask describing which input protocols are active. Each bit of this mask is used for a protocol. Through that, multiple protocols can be defined on a single port. (The bitfield inRtcm3 is not supported for protocol versions less than 20.00)
bit 0	U:1	`inUbx`	-	-	
bit 1	U:1	`inNmea`	-	-	
bit 2	U:1	`inRtcm`	-	-	
bit 5	U:1	`inRtcm3`	-	-	
14	X2	`outProtoMask`	-	-	A mask describing which output protocols are active. Each bit of this mask is used for a protocol. Through that, multiple protocols can be defined on a single port. (The bitfield outRtcm3 is not supported for protocol versions less than 20.00)
bit 0	U:1	`outUbx`	-	-	
bit 1	U:1	`outNmea`	-	-	
bit 5	U:1	`outRtcm3`	-	-	
16	X2	`flags`	-	-	Flags bit mask
bit 1	U:1	`extendedTx Timeout`	-	-	Extended TX timeout: if set, the port will time out if allocated TX memory >=4 kB and no activity for 1.5 s. (not supported for protocol versions less than 13.01)
18	U1[2]	`reserved2`	-	-	Reserved

3.10.15.5 Port configuration for I2C (DDC) port

Message	**UBX-CFG-PRT** **Port configuration for I2C (DDC) port**
Type	Get/set
Comment	**This message is deprecated in protocol versions greater than 23.01. Use** UBX-CFG-VALSET, UBX-CFG-VALGET, UBX-CFG-VALDEL **instead.** See the Legacy UBX Message Fields Reference for the corresponding configuration item. Several configurations can be concatenated to one input message. In this case the payload length can be a multiple of the normal length (see the other versions of CFG-PRT). Output messages from the module contain only one configuration unit.

	Header	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
Message structure	0xb5 0x62	0x06	0x00	20	*see below*	CK_A CK_B

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	U1	`portID`	-	-	Port identifier number (= 0 for I2C (DDC) port)
1	U1	`reserved0`	-	-	Reserved

2	X2	txReady	-	-	TX ready PIN configuration (not supported for protocol versions less than 13.01)
bit 0	U:1	en	-	-	Enable TX ready feature for this port
bit 1	U:1	pol	-	-	Polarity • 0 High-active • 1 Low-active
bits 6…2	U:5	pin	-	-	PIO to be used (must not be in use by another function)
bits 15…7	U:9	thres	-	-	Threshold The given threshold is multiplied by 8 bytes. The TX ready PIN goes active after >= thres*8 bytes are pending for the port and going inactive after the last pending bytes have been written to hardware (0-4 bytes before end of stream). • 0x000 no threshold • 0x001 8byte • 0x002 16byte • ... • 0x1FE 4080byte • 0x1FF 4088byte
4	X4	mode	-	-	I2C (DDC) Mode Flags
bits 7…1	U:7	slaveAddr	-	-	Slave address Range: 0x07 < slaveAddr < 0x78. Bit 0 must be 0
8	U1[4]	reserved1	-	-	Reserved
12	X2	inProtoMask	-	-	A mask describing which input protocols are active. Each bit of this mask is used for a protocol. Through that, multiple protocols can be defined on a single port. (The bitfield inRtcm3 is not supported for protocol versions less than 20.00)
bit 0	U:1	inUbx	-	-	
bit 1	U:1	inNmea	-	-	
bit 2	U:1	inRtcm	-	-	
bit 5	U:1	inRtcm3	-	-	
14	X2	outProtoMask	-	-	A mask describing which output protocols are active. Each bit of this mask is used for a protocol. Through that, multiple protocols can be defined on a single port. (The bitfield outRtcm3 is not supported for protocol versions less than 20.00)
bit 0	U:1	outUbx	-	-	
bit 1	U:1	outNmea	-	-	
bit 5	U:1	outRtcm3	-	-	
16	X2	flags	-	-	Flags bit mask
bit 1	U:1	extendedTx Timeout	-	-	Extended TX timeout: if set, the port will time out if allocated TX memory >=4 kB and no activity for 1.5 s (not supported for protocol versions less than 13.01).
18	U1[2]	reserved2	-	-	Reserved

3.10.16 UBX-CFG-PWR (0x06 0x57)

3.10.16.1 Put receiver in a defined power state

Message	**UBX-CFG-PWR** **Put receiver in a defined power state**					
Type	Set					
Comment	☞ This message is deprecated in protocol versions greater than 17. Use UBX-CFG-RST for GNSS start/stop and UBX-RXM-PMREQ for software backup.					
Message structure	*Header* 0xb5 0x62	*Class* 0x06	*ID* 0x57	*Length (Bytes)* 8	*Payload* *see below*	*Checksum* CK_A CK_B

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	U1	`version`	-	-	Message version (0x01 for this version)
1	U1[3]	`reserved0`	-	-	Reserved
4	U4	`state`	-	-	Enter system state • 0x52554E20 = GNSS running • 0x53544F50 = GNSS stopped • 0x42434B50 = Software backup. USB interface will be disabled, other wakeup source is needed.

3.10.17 UBX-CFG-RATE (0x06 0x08)

3.10.17.1 Navigation/measurement rate settings

Message	**UBX-CFG-RATE** **Navigation/measurement rate settings**					
Type	Get/set					
Comment	**This message is deprecated in protocol versions greater than 23.01. Use** UBX-CFG-VALSET, UBX-CFG-VALGET, UBX-CFG-VALDEL **instead.** See the Legacy UBX Message Fields Reference for the corresponding configuration item. This message allows the user to alter the rate at which navigation solutions (and the measurements that they depend on) are generated by the receiver. The calculation of the navigation solution will always be aligned to the top of a second zero (first second of the week) of the configured reference time system. (Navigation period is an integer multiple of the measurement period for protocol versions greater than 17.00). • Each measurement triggers the measurements generation and, if available, raw data output. • The navRate value defines that every nth measurement triggers a navigation epoch. • The update rate has a direct influence on the power consumption. The more fixes that are required, the more CPU power and communication resources are required. • For most applications a 1 Hz update rate would be sufficient. • When using power save mode, measurement and navigation rate can differ from the values configured here.					
Message structure	*Header* 0xb5 0x62	*Class* 0x06	*ID* 0x08	*Length (Bytes)* 6	*Payload* *see below*	*Checksum* CK_A CK_B

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	U2	`measRate`	-	ms	The elapsed time between GNSS measurements, which defines the rate, e.g. 100 ms => 10 Hz, 1000 ms => 1 Hz, 10000 ms => 0.1 Hz. Measurement rate should be greater than or equal to 25 ms. (Measurement rate should be greater than or equal to 50 ms for protocol versions less than 24.00).

2	U2	navRate	-	cycles	The ratio between the number of measurements and the number of navigation solutions, e.g. 5 means five measurements for every navigation solution. Maximum value is 127. (This parameter is ignored and the navRate is fixed to 1 for protocol versions less than 18.00).
4	U2	timeRef	-	-	The time system to which measurements are aligned: • 0 = UTC time • 1 = GPS time • 2 = GLONASS time (not supported for protocol versions less than 18.00) • 3 = BeiDou time (not supported for protocol versions less than 18.00) • 4 = Galileo time (not supported for protocol versions less than 18.00) • 5 = NavIC time (not supported for protocol versions less than 29.00)

3.10.18 UBX-CFG-RINV (0x06 0x34)

3.10.18.1 Contents of remote inventory

Message	**UBX-CFG-RINV** **Contents of remote inventory**
Type	Get/set
Comment	**This message is deprecated in protocol versions greater than 23.01. Use** UBX-CFG-VALSET, UBX-CFG-VALGET, UBX-CFG-VALDEL **instead.** If *N* is greater than 30, the excess bytes are discarded. See the Legacy UBX Message Fields Reference for the corresponding configuration item.

Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x06	0x34	1 + [0..n]	*see below*	CK_A CK_B

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	X1	flags	-	-	Flags
bit 0	U:1	dump	-	-	Dump data at startup. Does not work if flag binary is set.
bit 1	U:1	binary	-	-	Data is binary.
Start of repeated group (N times)					
1 + n	U1	data	-	-	Data to store/stored in remote inventory.
End of repeated group (N times)					

3.10.19 UBX-CFG-RST (0x06 0x04)

3.10.19.1 Reset receiver / Clear backup data structures

Message	**UBX-CFG-RST** **Reset receiver / Clear backup data structures**
Type	Command
Comment	Do not expect this message to be acknowledged by the receiver. • Newer FW version will not acknowledge this message at all. • Older FW version will acknowledge this message but the acknowledge may not be sent completely before the receiver is reset.

Message structure	Header	Class	ID	Length (Bytes)	Payload	Checksum
	0xb5 0x62	0x06	0x04	4	see below	CK_A CK_B

Payload description:

Byte offset	Type	Name	Scale	Unit	Description
0	X2	navBbrMask	-	-	BBR sections to clear. The following special sets apply: • 0x0000 Hot start • 0x0001 Warm start • 0xFFFF Cold start
bit 0	U:1	eph	-	-	Ephemeris
bit 1	U:1	alm	-	-	Almanac
bit 2	U:1	health	-	-	Health
bit 3	U:1	klob	-	-	Klobuchar parameters
bit 4	U:1	pos	-	-	Position
bit 5	U:1	clkd	-	-	Clock drift
bit 6	U:1	osc	-	-	Oscillator parameter
bit 7	U:1	utc	-	-	UTC correction + GPS leap seconds parameters
bit 8	U:1	rtc	-	-	RTC
bit 11	U:1	sfdr	-	-	SFDR Parameters (only available on the ADR/UDR/HPS product variant) and weak signal compensation estimates
bit 12	U:1	vmon	-	-	SFDR Vehicle Monitoring Parameter (only available on the ADR/UDR/HPS product variant)
bit 13	U:1	tct	-	-	TCT Parameters (only available on the ADR/UDR/HPS product variant)
bit 15	U:1	aop	-	-	Autonomous orbit parameters
2	U1	resetMode	-	-	Reset Type • 0x00 = Hardware reset (watchdog) immediately • 0x01 = Controlled software reset • 0x02 = Controlled software reset (GNSS only) • 0x04 = Hardware reset (watchdog) after shutdown • 0x08 = Controlled GNSS stop • 0x09 = Controlled GNSS start
3	U1	reserved0	-	-	Reserved

3.10.20 UBX-CFG-SBAS (0x06 0x16)

3.10.20.1 SBAS configuration

Message	**UBX-CFG-SBAS** **SBAS configuration**
Type	Get/set
Comment	**This message is deprecated in protocol versions greater than 23.01. Use UBX-CFG-VALSET, UBX-CFG-VALGET, UBX-CFG-VALDEL instead.** This message configures the SBAS receiver subsystem (i.e. WAAS, EGNOS, MSAS). See SBAS configuration settings description in the integration manual for a detailed description of how these settings affect receiver operation.

Message structure	Header	Class	ID	Length (Bytes)	Payload	Checksum
	0xb5 0x62	0x06	0x16	8	see below	CK_A CK_B

Payload description:

Byte offset	Type	Name	Scale	Unit	Description
0	X1	mode	-	-	SBAS mode
bit 0	U:1	enabled	-	-	SBAS enabled (1) / disabled (0) - This field is deprecated; use UBX-CFG-GNSS to enable/disable SBAS operation
bit 1	U:1	test	-	-	SBAS testbed: Use data anyhow (1) / Ignore data when in test mode (SBAS msg 0)
1	X1	usage	-	-	SBAS usage
bit 0	U:1	range	-	-	Use SBAS GEOs as a ranging source (for navigation)
bit 1	U:1	diffCorr	-	-	Use SBAS differential corrections
bit 2	U:1	integrity	-	-	Use SBAS integrity information. If enabled, the receiver will only use GPS satellites for which integrity information is available.
2	U1	maxSBAS	-	-	Maximum number of SBAS prioritized tracking channels (valid range: 0 - 3) to use (obsolete and superseded by UBX-CFG-GNSS for protocol versions 14.00+).
3	X1	scanmode2	-	-	Continuation of scanmode bitmask below
bit 0	U:1	PRN152	-	-	
bit 1	U:1	PRN153	-	-	
bit 2	U:1	PRN154	-	-	
bit 3	U:1	PRN155	-	-	
bit 4	U:1	PRN156	-	-	
bit 5	U:1	PRN157	-	-	
bit 6	U:1	PRN158	-	-	
4	X4	scanmode1	-	-	Which SBAS PRN numbers to search for (bitmask). If all bits are set to zero, auto-scan (i.e. all valid PRNs) are searched. Every bit corresponds to a PRN number.
bit 0	U:1	PRN120	-	-	
bit 1	U:1	PRN121	-	-	
bit 2	U:1	PRN122	-	-	
bit 3	U:1	PRN123	-	-	
bit 4	U:1	PRN124	-	-	
bit 5	U:1	PRN125	-	-	
bit 6	U:1	PRN126	-	-	
bit 7	U:1	PRN127	-	-	
bit 8	U:1	PRN128	-	-	
bit 9	U:1	PRN129	-	-	
bit 10	U:1	PRN130	-	-	
bit 11	U:1	PRN131	-	-	
bit 12	U:1	PRN132	-	-	
bit 13	U:1	PRN133	-	-	
bit 14	U:1	PRN134	-	-	

bit 15	U:1	PRN135	-	-	
bit 16	U:1	PRN136	-	-	
bit 17	U:1	PRN137	-	-	
bit 18	U:1	PRN138	-	-	
bit 19	U:1	PRN139	-	-	
bit 20	U:1	PRN140	-	-	
bit 21	U:1	PRN141	-	-	
bit 22	U:1	PRN142	-	-	
bit 23	U:1	PRN143	-	-	
bit 24	U:1	PRN144	-	-	
bit 25	U:1	PRN145	-	-	
bit 26	U:1	PRN146	-	-	
bit 27	U:1	PRN147	-	-	
bit 28	U:1	PRN148	-	-	
bit 29	U:1	PRN149	-	-	
bit 30	U:1	PRN150	-	-	
bit 31	U:1	PRN151	-	-	

3.10.21 UBX-CFG-TMODE3 (0x06 0x71)

3.10.21.1 Time mode settings 3

Message	**UBX-CFG-TMODE3** **Time mode settings 3**
Type	Get/set
Comment	**This message is deprecated in protocol versions greater than 23.01. Use** UBX-CFG-VALSET, UBX-CFG-VALGET, UBX-CFG-VALDEL **instead.** See the Legacy UBX Message Fields Reference for the corresponding configuration item. Configures the receiver to be in Time Mode. The position referred to in this message is that of the Antenna Reference Point (ARP). Note that using UBX-CFG-TMODE3 to set the receiver mode to Survey In or to Fixed Mode, will set automatically the dynamic platform model (CFG-NAVSPG-DYNMODEL) to Stationary. Note that using UBX-CFG-TMODE3 to set the receiver mode to Disabled, will set automatically the dynamic platform model (CFG-NAVSPG-DYNMODEL) to Portable.

	Header	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
Message structure	0xb5 0x62	0x06	0x71	40	*see below*	CK_A CK_B

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	U1	version	-	-	Message version (0x00 for this version)
1	U1	reserved0	-	-	Reserved
2	X2	flags	-	-	Receiver mode flags
bits 7...0	U:8	mode	-	-	Receiver Mode: • 0 Disabled • 1 Survey In • 2 Fixed Mode (true ARP position information required) • 3-255 Reserved

bit 8	U:1	lla	-	-	Position is given in LAT/LON/ALT (default is ECEF)
4	I4	ecefXOrLat	-	cm_or_deg*1e-7	WGS84 ECEF X coordinate (or latitude) of the ARP position, depending on flags above
8	I4	ecefYOrLon	-	cm_or_deg*1e-7	WGS84 ECEF Y coordinate (or longitude) of the ARP position, depending on flags above
12	I4	ecefZOrAlt	-	cm	WGS84 ECEF Z coordinate (or altitude) of the ARP position, depending on flags above
16	I1	ecefXOrLatHP	-	0.1_mm_or_deg*1e-9	High-precision WGS84 ECEF X coordinate (or latitude) of the ARP position, depending on flags above. Must be in the range -99..+99. The precise WGS84 ECEF X coordinate in units of cm, or the precise WGS84 ECEF latitude in units of 1e-7 degrees, is given by ecefXOrLat + (ecefXOrLatHP * 1e-2)
17	I1	ecefYOrLonHP	-	0.1_mm_or_deg*1e-9	High-precision WGS84 ECEF Y coordinate (or longitude) of the ARP position, depending on flags above. Must be in the range -99..+99. The precise WGS84 ECEF Y coordinate in units of cm, or the precise WGS84 ECEF longitude in units of 1e-7 degrees, is given by ecefYOrLon + (ecefYOrLonHP * 1e-2)
18	I1	ecefZOrAltHP	-	0.1_mm	High-precision WGS84 ECEF Z coordinate (or altitude) of the ARP position, depending on flags above. Must be in the range -99..+99. The precise WGS84 ECEF Z coordinate, or altitude coordinate, in units of cm is given by ecefZOrAlt + (ecefZOrAltHP * 1e-2)
19	U1	reserved1	-	-	Reserved
20	U4	fixedPosAcc	-	0.1_mm	Fixed position 3D accuracy
24	U4	svinMinDur	-	s	Survey-in minimum duration
28	U4	svinAccLimit	-	0.1_mm	Survey-in position accuracy limit
32	U1[8]	reserved2	-	-	Reserved

3.10.22 UBX-CFG-TP5 (0x06 0x31)

3.10.22.1 Time pulse parameters

Message	**UBX-CFG-TP5** **Time pulse parameters**				
Type	Get/set				
Comment	**This message is deprecated in protocol versions greater than 27. Use** UBX-CFG-VALSET, UBX-CFG-VALGET, UBX-CFG-VALDEL **instead.** See the Legacy UBX Message Fields Reference for the corresponding configuration item.				
Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload* / *Checksum*
	0xb5 0x62	0x06	0x31	32	see below / CK_A CK_B

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	U1	tpIdx	-	-	Time pulse selection (0 = TIMEPULSE, 1 = TIMEPULSE2)
1	U1	version	-	-	Message version (0x01 for this version)

2		U1[2]	reserved0	-	-	Reserved
4		I2	antCableDelay	-	ns	Antenna cable delay
6		I2	rfGroupDelay	-	ns	RF group delay
8		U4	freqPeriod	-	Hz_or_us	Frequency or period time, depending on setting of bit 'isFreq'
12		U4	freqPeriodLock	-	Hz_or_us	Frequency or period time when locked to GNSS time, only used if 'lockedOtherSet' is set
16		U4	pulseLenRatio	-	us_or_ 2^-32	Pulse length or duty cycle, depending on 'isLength'
20		U4	pulseLenRatio Lock	-	us_or_ 2^-32	Pulse length or duty cycle when locked to GNSS time, only used if 'lockedOtherSet' is set
24		I4	userConfig Delay	-	ns	User-configurable time pulse delay
28		X4	flags	-	-	Configuration flags
	bit 0	U:1	active	-	-	If set enable time pulse; if pin assigned to another function, other function takes precedence. Must be set for FTS variant.
	bit 1	U:1	lockGnssFreq	-	-	If set, synchronize time pulse to GNSS as soon as GNSS time is valid. If not set, or before GNSS time is valid, use local clock. This flag is ignored by the FTS product variant; in this case the receiver always locks to the best available time/frequency reference (which is not necessarily GNSS). This flag can be unset only in Timing product variants.
	bit 2	U:1	lockedOtherSet	-	-	If set the receiver switches between the timepulse settings given by 'freqPeriodLocked' & 'pulseLenLocked' and those given by 'freqPeriod' & 'pulseLen'. The 'Locked' settings are used where the receiver has an accurate sense of time. For non-FTS products, this occurs when GNSS solution with a reliable time is available, but for FTS products the setting syncMode field governs behavior. In all cases, the receiver only uses 'freqPeriod' & 'pulseLen' when the flag is unset.
	bit 3	U:1	isFreq	-	-	If set 'freqPeriodLock' and 'freqPeriod' are interpreted as frequency, otherwise interpreted as period.
	bit 4	U:1	isLength	-	-	If set 'pulseLenRatioLock' and 'pulseLenRatio' interpreted as pulse length, otherwise interpreted as duty cycle.
	bit 5	U:1	alignToTow	-	-	Align pulse to top of second (period time must be integer fraction of 1s). Also set 'lockGnssFreq' to use this feature. This flag is ignored by the FTS product variant; it is assumed to be always set (as is lockGnssFreq). Set maxSlewRate and maxPhaseCorrRate fields of **UBX-CFG-SMGR** to 0 to disable alignment.
	bit 6	U:1	polarity	-	-	Pulse polarity: • 0 = falling edge at top of second • 1 = rising edge at top of second
	bits 10…7	U:4	gridUtcGnss	-	-	Timegrid to use: • 0 = UTC • 1 = GPS

					• 2 = GLONASS • 3 = BeiDou • 4 = Galileo (not supported for protocol versions less than 18.00) This flag is only relevant if 'lockGnssFreq' and 'alignToTow' are set. Note that configured GNSS time is estimated by the receiver if locked to any GNSS system. If the receiver has a valid GNSS fix it will attempt to steer the TP to the specified time grid even if the specified time is not based on information from the constellation's satellites. To ensure timing based purely on a given GNSS, restrict the supported constellations in UBX-CFG-GNSS.
bits 13...11	U:3	syncMode	-	-	Sync Manager lock mode to use: • 0 = switch to 'freqPeriodLock' and 'pulseLenRatioLock' as soon as Sync Manager has an accurate time, never switch back to 'freqPeriod' and 'pulseLenRatio' • 1 = switch to 'freqPeriodLock' and 'pulseLenRatioLock' as soon as Sync Manager has an accurate time, and switch back to 'freqPeriod' and 'pulseLenRatio' as soon as time gets inaccurate This field is only relevant for the FTS product variant. This field is only relevant if the flag 'lockedOtherSet' is set.

3.10.23 UBX-CFG-USB (0x06 0x1b)

3.10.23.1 USB configuration

Message	**UBX-CFG-USB** **USB configuration**
Type	Get/set
Comment	**This message is deprecated in protocol versions greater than 23.01. Use UBX-CFG-VALSET, UBX-CFG-VALGET, UBX-CFG-VALDEL instead.** See the Legacy UBX Message Fields Reference for the corresponding configuration item.

Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x06	0x1b	108	*see below*	CK_A CK_B

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	U2	vendorID	-	-	Vendor ID. This field shall only be set to registered Vendor IDs. Changing this field requires special Host drivers.
2	U2	productID	-	-	Product ID. Changing this field requires special Host drivers.
4	U1[2]	reserved0	-	-	Reserved
6	U1[2]	reserved1	-	-	Reserved
8	U2	power Consumption	-	mA	Power consumed by the device
10	X2	flags	-	-	various configuration flags
bit 0	U:1	reEnum	-	-	force re-enumeration
bit 1	U:1	powerMode	-	-	self-powered (1), bus-powered (0)

12	CH[32]	`vendorString`	-	-	String containing the vendor name. 32 ASCII bytes including 0-termination.
44	CH[32]	`productString`	-	-	String containing the product name. 32 ASCII bytes including 0-termination.
76	CH[32]	`serialNumber`	-	-	String containing the serial number. 32 ASCII bytes including 0-termination. Changing the String fields requires special Host drivers.

3.10.24 UBX-CFG-VALDEL (0x06 0x8c)

3.10.24.1 Delete configuration item values

Message	**UBX-CFG-VALDEL** **Delete configuration item values**
Type	Set
Comment	Overview: • This message can be used to delete saved configuration to effectively revert the item values to defaults. • This message can delete saved configuration from the flash configuration layer and the BBR configuration layer. The changes will not be effective until these layers are loaded into the RAM layer. • This message is limited to containing a maximum of 64 keys up for deletion; i.e. N is a maximum of 64. • This message can be used multiple times and every time the result will be applied immediately. To send this message multiple times with the result being applied at the end, see version 1 of UBX-CFG-VALDEL that supports transactions. • This message does not check if the resulting configuration is valid. • See Receiver configuration for details. This message returns a UBX-ACK-NAK and no configuration is applied: • if any key is unknown to the receiver FW • if the layer's bitfield does not specify a layer to delete a value from. Notes: • If a key is sent multiple times within the same message, then the value is effectively deleted only once. • Attempting to delete items that have not been set before, or that have already been deleted, is considered a valid request.

Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x06	0x8c	4 + [0..n]·4	*see below*	CK_A CK_B

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	U1	`version`	-	-	Message version (0x00 for this version)
1	X1	`layers`	-	-	The layers where the configuration should be deleted from
bit 1	U:1	`bbr`	-	-	Delete configuration from the BBR layer
bit 2	U:1	`flash`	-	-	Delete configuration from the Flash layer
2	U1[2]	`reserved0`	-	-	Reserved
Start of repeated group (N times)					
4 + n·4	U4	`keys`	-	-	Configuration key IDs of the configuration items to be deleted
End of repeated group (N times)					

3.10.24.2 Delete configuration item values (with transaction)

Message	**UBX-CFG-VALDEL**
	Delete configuration item values (with transaction)
Type	Set
Comment	Overview: • This message can be used to delete saved configuration to effectively revert them to defaults. • This message can delete saved configuration from the flash configuration layer and the BBR configuration layer. The changes will not be effective until these layers are loaded into the RAM layer. • This message is limited to containing a maximum of 64 keys up for deletion; i.e. N is a maximum of 64. • This message can be used multiple times with the result being managed within a transaction. • This message does not check if the resulting configuration is valid. • See Receiver configuration for details. • See version 0 of UBX-CFG-VALDEL for simplified version of this message. This message returns a UBX-ACK-NAK, cancels any started transaction, and no configuration is applied: • if any key within a transaction is unknown to the receiver FW • if an invalid transaction state transition is requested • if the layer's bitfield changes within a transaction • if the layer's bitfield does not specify a layer to delete a value from. Notes: • Any request for another UBX-CFG- message type (including UBX-CFG-VALSET and UBX-CFG-VALGET) will cancel any started transaction, and no configuration is applied. • This message can be sent with no keys to delete for the purposes of managing the transaction state transition. • If a key is sent multiple times within the same message or within the same transaction, then the value is effectively deleted only once. • Attempting to delete items that have not been set before, or that have already been deleted, is considered a valid request.

Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x06	0x8c	4 + [0..n]·4	*see below*	CK_A CK_B

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	U1	`version`	-	-	Message version (0x01 for this version)
1	X1	`layers`	-	-	The layers where the configuration should be deleted from
bit 1	U:1	`bbr`	-	-	Delete configuration from the BBR layer
bit 2	U:1	`flash`	-	-	Delete configuration from the Flash layer
2	X1	`transaction`	-	-	Transaction action to be applied:
bits 1…0	U:2	`action`	-	-	Transaction action to be applied: • 0 = Transactionless UBX-CFG-VALDEL: In the next UBX-CFG-VALDEL, it can be either 0 or 1. If a transaction has not yet been started, the incoming configuration is applied. If a transaction has already been started, cancels any started transaction and the incoming configuration is applied. • 1 = (Re)Start deletion transaction: In the next UBX-CFG-VALDEL, it can be either 0, 1, 2 or 3. If a transaction has not yet been started, a transaction will be started. If a transaction has already been started, restarts the transaction, effectively removing all previous non-applied UBX-CFG-VALDEL messages. • 2 = Deletion transaction ongoing: In the next UBX-CFG-VALDEL, it can be either 0, 1, 2 or 3. • 3 = Apply and end a deletion transaction: In the next UBX-CFG-VALDEL, it can be either 0 or 1.

3	U1	reserved0	-	-	Reserved
Start of repeated group (N times)					
4 + n·4	U4	keys	-	-	Configuration key IDs of the configuration items to be deleted
End of repeated group (N times)					

3.10.25 UBX-CFG-VALGET (0x06 0x8b)

3.10.25.1 Get configuration items

Message	**UBX-CFG-VALGET** **Get configuration items**
Type	Poll request
Comment	Overview: • This message is used to get configuration values by providing a list of configuration key IDs, which identify the configuration items to retrieve. • This message can specify the configuration layer where the values of the specified configuration items are retrieved from. • This message is limited to containing a maximum of 64 key IDs. • See Receiver configuration for details. This message returns a UBX-ACK-NAK: • if any key is unknown to the receiver FW • if the layer field specifies an invalid layer to get the value from • if the keys array specifies more than 64 key IDs. Notes: • If a value is requested multiple times within the same poll request, then the reply will contain it multiple times. • The provided keys can be complete key values (group and item specifiers) or wild-card specifications. A complete key value will constitute a request for one key-value pair. A key value that has a valid group specifier and 0xffff in the item part of the key value (bits 0-15) constitutes a request for all items in the specified group. A key with a value of 0xfff in the group part of the key value (bits 16-27) is a request for all items known to the receiver in all groups. • The response message is limited to containing a maximum of 64 key-value pairs. If there are wild-card specifications then there may be more than 64 possible responses. In order to handle this, the 'position' field can specify that the response message should skip this number of key-value pairs before it starts constructing the message. This allows a large set of values to be retrieved 64 at a time. If the response contains less than 64 key-value pairs then all values have been reported, otherwise there may be more to read. • It is not possible to retrieve configuration values for the same configuration item from multiple configuration layers. Separate poll requests must be made for each desired layer.

Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x06	0x8b	4 + [0..n]·4	*see below*	CK_A CK_B

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	U1	version	-	-	Message version (0x00 for this version)
1	U1	layer	-	-	The layer from which the configuration items should be retrieved: • 0 - RAM layer • 1 - BBR layer • 2 - Flash layer • 7 - Default layer
2	U2	position	-	-	Skip this many key values before constructing output message
Start of repeated group (N times)					

4 + n·4	U4	keys	-	-	Configuration key IDs of the configuration items to be retrieved
End of repeated group (N times)					

3.10.25.2 Configuration items

Message	**UBX-CFG-VALGET** **Configuration items**
Type	Polled
Comment	This message is output by the receiver to return requested configuration data (key and value pairs). See Receiver configuration for details.

Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x06	0x8b	4 + [0..n]	*see below*	CK_A CK_B

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	U1	version	-	-	Message version (0x01 for this version)
1	U1	layer	-	-	The layer from which the configuration item was retrieved: • 0 - RAM layer • 1 - BBR • 2 - Flash • 7 - Default
2	U2	position	-	-	Number of configuration items skipped in the result set before constructing this message (mirrors the equivalent field in the request message)
Start of repeated group (N times)					
4 + n	U1	cfgData	-	-	Configuration data (key and value pairs)
End of repeated group (N times)					

3.10.26 UBX-CFG-VALSET (0x06 0x8a)

3.10.26.1 Set configuration item values

Message	**UBX-CFG-VALSET** **Set configuration item values**
Type	Set
Comment	Overview: • This message is used to set a configuration by providing configuration data (a list of key and value pairs), which identify the configuration items to change, and their new values. • This message is limited to containing a maximum of 64 key-value pairs. • This message can be used multiple times and every time the result will be applied immediately. To send this message multiple times with the result being applied at the end, see version 1 of UBX-CFG-VALSET that supports transactions. • See Receiver configuration for details. This message returns a UBX-ACK-NAK and no configuration is applied: • if any key is unknown to the receiver FW • if the layer's bitfield does not specify a layer to save a value to • if the requested configuration is not valid. The validity of a configuration is checked only if the message requests to apply the configuration to the RAM configuration layer. Notes: • If a key is sent multiple times within the same message, then the value eventually being applied is the last sent.

Message structure	Header	Class	ID	Length (Bytes)	Payload	Checksum
	0xb5 0x62	0x06	0x8a	4 + [0..n]	see below	CK_A CK_B

Payload description:

Byte offset	Type	Name	Scale	Unit	Description
0	U1	`version`	-	-	Message version (0x00 for this version)
1	X1	`layers`	-	-	The layers where the configuration should be applied
bit 0	U:1	`ram`	-	-	Update configuration in the RAM layer
bit 1	U:1	`bbr`	-	-	Update configuration in the BBR layer
bit 2	U:1	`flash`	-	-	Update configuration in the Flash layer
2	U1[2]	`reserved0`	-	-	Reserved
Start of repeated group (N times)					
4 + n	U1	`cfgData`	-	-	Configuration data (key and value pairs)
End of repeated group (N times)					

3.10.26.2 Set configuration item values (with transaction)

Message	**UBX-CFG-VALSET** **Set configuration item values (with transaction)**
Type	Set

Comment

Overview:

- This message is used to set a configuration by providing configuration data (a list of key and value pairs), which identify the configuration items to change, and their new values.
- This message is limited to containing a maximum of 64 key-value pairs.
- This message can be used multiple times with the result being managed within a transaction. Within a transaction there is no limit on the number key-value pairs; a transaction is effectively limited to the number of known keys.
- See Receiver configuration for details.
- See version 0 of UBX-CFG-VALSET for simplified version of this message.

This message returns a UBX-ACK-NAK, cancels any started transaction, and no configuration is applied:

- if any key within a transaction is unknown to the receiver FW
- if an invalid transaction state transition is requested
- if the layer's bitfield changes within a transaction
- if the layer's bitfield does not specify a layer to save a value to

This message returns a UBX-ACK-NAK, and no configuration is applied:

- if the requested configuration is not valid. While in a transaction context, only the last message that requests to apply the transaction returns a UBX-ACK-NAK. The validity of a configuration is checked only if the message requests to apply the configuration to the RAM configuration layer. This also applies to a transactionless request.

Notes:

- Any request for another UBX-CFG-message type (including UBX-CFG-VALDEL and UBX-CFG-VALGET) will cancel any started transaction, and no configuration is applied.
- This message can be sent with no key/values to set for the purposes of managing the transaction state transition.
- If a key is sent multiple times within the same message or within the same transaction, then the value eventually being applied is the last sent.

Message structure	Header	Class	ID	Length (Bytes)	Payload	Checksum
	0xb5 0x62	0x06	0x8a	4 + [0..n]	see below	CK_A CK_B

Payload description:

Byte offset	Type	Name	Scale	Unit	Description
0	U1	`version`	-	-	Message version (0x01 for this version)
1	X1	`layers`	-	-	The layers where the configuration should be applied
bit 0	U:1	`ram`	-	-	Update configuration in the RAM layer

bit 1	U:1	bbr	-	-	Update configuration in the BBR layer
bit 2	U:1	flash	-	-	Update configuration in the Flash layer
2	U1	transaction	-	-	Transaction action to be applied
bits 1...0	U:2	action	-	-	Transaction action to be applied: • 0 = Transactionless UBX-CFG-VALSET: In the next UBX-CFG-VALSET, it can be either 0 or 1. If a transaction has not yet been started, the incoming configuration is applied (if valid). If a transaction has already been started, cancels any started transaction and the incoming configuration is applied (if valid). • 1 = (Re)Start set transaction: In the next UBX-CFG-VALSET, it can be either 0, 1, 2 or 3. If a transaction has not yet been started, a transaction will be started. If a transaction has already been started, restarts the transaction, effectively removing all previous non-applied UBX-CFG-VALSET messages. • 2 = Set transaction ongoing: In the next UBX-CFG-VALSET, it can be either 0, 1, 2 or 3. • 3 = Apply and end a set transaction: In the next UBX-CFG-VALSET, it can be either 0 or 1.
3	U1	reserved0	-	-	Reserved
Start of repeated group (N times)					
4 + n	U1	cfgData	-	-	Configuration data (key and value pairs)
End of repeated group (N times)					

3.11 UBX-INF (0x04)

Messages in the UBX-INF class are used to output strings from the firmware or application code. All messages have an associated type to indicate the nature or priority of the message.

3.11.1 UBX-INF-DEBUG (0x04 0x04)

3.11.1.1 ASCII output with debug contents

Message	**UBX-INF-DEBUG** **ASCII output with debug contents**						
Type	Output						
Comment	This message has a variable length payload, representing an ASCII string.						
Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*		*Payload*	*Checksum*
	0xb5 0x62	0x04	0x04	[0..n]		see below	CK_A CK_B

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
Start of repeated group (N times)					
0 + n	CH	str	-	-	ASCII Character
End of repeated group (N times)					

3.11.2 UBX-INF-ERROR (0x04 0x00)

3.11.2.1 ASCII output with error contents

Message	**UBX-INF-ERROR**					
	ASCII output with error contents					
Type	Output					
Comment	This message has a variable length payload, representing an ASCII string.					
Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x04	0x00	[0..n]	*see below*	CK_A CK_B

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
Start of repeated group (N times)					
0 + n	CH	`str`	-	-	ASCII Character
End of repeated group (N times)					

3.11.3 UBX-INF-NOTICE (0x04 0x02)

3.11.3.1 ASCII output with informational contents

Message	**UBX-INF-NOTICE**					
	ASCII output with informational contents					
Type	Output					
Comment	This message has a variable length payload, representing an ASCII string.					
Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x04	0x02	[0..n]	*see below*	CK_A CK_B

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
Start of repeated group (N times)					
0 + n	CH	`str`	-	-	ASCII Character
End of repeated group (N times)					

3.11.4 UBX-INF-TEST (0x04 0x03)

3.11.4.1 ASCII output with test contents

Message	**UBX-INF-TEST**					
	ASCII output with test contents					
Type	Output					
Comment	This message has a variable length payload, representing an ASCII string.					
Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x04	0x03	[0..n]	*see below*	CK_A CK_B

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
Start of repeated group (N times)					
0 + n	CH	`str`	-	-	ASCII Character
End of repeated group (N times)					

3.11.5 UBX-INF-WARNING (0x04 0x01)

3.11.5.1 ASCII output with warning contents

Message	**UBX-INF-WARNING**					
	ASCII output with warning contents					
Type	Output					
Comment	This message has a variable length payload, representing an ASCII string.					
Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x04	0x01	[0..n]	*see below*	CK_A CK_B

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
Start of repeated group (N times)					
0 + n	CH	`str`	-	-	ASCII Character
End of repeated group (N times)					

3.12 UBX-LOG (0x21)

The messages in the UBX-LOG class are used to configure and report status information of the logging and data batching features.

3.12.1 UBX-LOG-CREATE (0x21 0x07)

3.12.1.1 Create log file

Message	**UBX-LOG-CREATE**					
	Create log file					
Type	Command					
Comment	This message is used to create an initial logging file and activate the logging subsystem. UBX-ACK-ACK or UBX-ACK-NAK are returned to indicate success or failure. This message does not handle activation of recording or filtering of log entries (see UBX-CFG-LOGFILTER).					
Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x21	0x07	8	*see below*	CK_A CK_B

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	U1	`version`	-	-	Message version (0x00 for this version)
1	X1	`logCfg`	-	-	Config flags
bit 0	U:1	`circular`	-	-	Log is circular (new entries overwrite old ones in a full log) if this bit set
2	U1	`reserved0`	-	-	Reserved
3	U1	`logSize`	-	-	Indicates the size of the log: • 0 (maximum safe size) = Ensures that logging will not be interrupted and enough space will be left available for all other uses of the filestore • 1 (minimum size) = • 2 (user-defined) = See 'userDefinedSize' below
4	U4	`userDefinedSize`	-	bytes	Sets the maximum amount of space in the filestore that can be used by the logging task. This field is only applicable if logSize is set to user-defined.

3.12.2 UBX-LOG-ERASE (0x21 0x03)

3.12.2.1 Erase logged data

Message	**UBX-LOG-ERASE** **Erase logged data**
Type	Command
Comment	This message deactivates the logging system and erases all logged data. UBX-ACK-ACK or UBX-ACK-NAK are returned to indicate success or failure.

Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x21	0x03	0	*see below*	CK_A CK_B

Payload This message has no payload.

3.12.3 UBX-LOG-FINDTIME (0x21 0x0e)

3.12.3.1 Find index of a log entry based on a given time

Message	**UBX-LOG-FINDTIME** **Find index of a log entry based on a given time**
Type	Input
Comment	This message can be used for a time-based search of a log. It can find the index of the first log entry with time equal to the given time, otherwise the index of the most recent entry with time less than the given time. This index can then be used with the UBX-LOG-RETRIEVE message to provide time-based retrieval of log entries. Searching a log is effective for a given time later than the base date (January 1st, 2004). Searching a log for a given time earlier than the base date will result in an 'entry not found' response. (Searching a log for a given time earlier than the base date will result in a UBX-ACK-NAK message for protocol versions less than 18.00). Searching a log for a given time greater than the last recorded entry's time will return the index of the last recorded entry. (If the logging has stopped due to lack of file space, such a search will result in a UBX-ACK-NAK message for protocol versions less than 18.00).

Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x21	0x0e	10	*see below*	CK_A CK_B

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	U1	`version`	-	-	Message version (0x00 for this version)
1	U1	`type`	-	-	Message type, 0 for request
2	U2	`year`	-	-	Year (1-65635) of UTC time
4	U1	`month`	-	-	Month (1-12) of UTC time
5	U1	`day`	-	-	Day (1-31) of UTC time
6	U1	`hour`	-	-	Hour (0-23) of UTC time
7	U1	`minute`	-	-	Minute (0-59) of UTC time
8	U1	`second`	-	-	Second (0-60) of UTC time
9	U1	`reserved0`	-	-	Reserved

3.12.3.2 Response to FINDTIME request

Message	**UBX-LOG-FINDTIME** **Response to FINDTIME request**
Type	Output
Comment	

Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x21	0x0e	8	*see below*	CK_A CK_B

Payload description:

Byte offset	Type	Name	Scale	Unit	Description
0	U1	`version`	-	-	Message version (0x01 for this version)
1	U1	`type`	-	-	Message type, 1 for response
2	U1[2]	`reserved0`	-	-	Reserved
4	U4	`entryNumber`	-	-	Index of the first log entry with time = given time, otherwise index of the most recent entry with time < given time. If 0xFFFFFFFF, no log entry found with time <= given time. The indexing of log entries is zero-based.

3.12.4 UBX-LOG-INFO (0x21 0x08)

3.12.4.1 Poll for log information

Message	**UBX-LOG-INFO** **Poll for log information**
Type	Poll request
Comment	Upon sending of this message, the receiver returns UBX-LOG-INFO as defined below.

Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x21	0x08	0	*see below*	CK_A CK_B

Payload This message has no payload.

3.12.4.2 Log information

Message	**UBX-LOG-INFO** **Log information**
Type	Output
Comment	This message is used to report information about the logging subsystem. Note: • The reported maximum log size will be smaller than that originally specified in LOG-CREATE due to logging and filestore implementation overheads. • Log entries are compressed in a variable length fashion, so it may be difficult to predict log space usage with any precision. • There may be times when the receiver does not have an accurate time (e.g. if the week number is not yet known), in which case some entries will not have a timestamp. This may result in the oldest/newest entry time values not taking account of these entries.

Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x21	0x08	48	*see below*	CK_A CK_B

Payload description:

Byte offset	Type	Name	Scale	Unit	Description
0	U1	`version`	-	-	Message version (0x01 for this version)
1	U1[3]	`reserved0`	-	-	Reserved
4	U4	`filestoreCapacity`	-	bytes	The capacity of the filestore
8	U1[8]	`reserved1`	-	-	Reserved
16	U4	`currentMaxLogSize`	-	bytes	The maximum size the current log is allowed to grow to
20	U4	`currentLogSize`	-	bytes	Approximate amount of space in log currently occupied

24	U4	entryCount	-	-	Number of entries in the log. Note: for circular logs this value will decrease when a group of entries is deleted to make space for new ones.
28	U2	oldestYear	-	-	Oldest entry UTC year (1-65635) or zero if there are no entries with known time
30	U1	oldestMonth	-	-	Oldest month (1-12)
31	U1	oldestDay	-	-	Oldest day (1-31)
32	U1	oldestHour	-	-	Oldest hour (0-23)
33	U1	oldestMinute	-	-	Oldest minute (0-59)
34	U1	oldestSecond	-	-	Oldest second (0-60)
35	U1	reserved2	-	-	Reserved
36	U2	newestYear	-	-	Newest year (1-65635) or zero if there are no entries with known time
38	U1	newestMonth	-	-	Newest month (1-12)
39	U1	newestDay	-	-	Newest day (1-31)
40	U1	newestHour	-	-	Newest hour (0-23)
41	U1	newestMinute	-	-	Newest minute (0-59)
42	U1	newestSecond	-	-	Newest second (0-60)
43	U1	reserved3	-	-	Reserved
44	X1	status	-	-	Log status flags
bit 3	U:1	recording	-	-	Log entry recording is currently turned on
bit 4	U:1	inactive	-	-	Logging system not active - no log present
bit 5	U:1	circular	-	-	The current log is circular
45	U1[3]	reserved4	-	-	Reserved

3.12.5 UBX-LOG-RETRIEVE (0x21 0x09)

3.12.5.1 Request log data

Message	**UBX-LOG-RETRIEVE** **Request log data**
Type	Command
Comment	This message is used to request logged data (log recording must first be disabled, see UBX-CFG-LOGFILTER). Log entries are returned in chronological order, using the messages UBX-LOG-RETRIEVEPOS and UBX-LOG-RETRIEVESTRING. If the odometer was enabled at the time a position was logged, then message UBX-LOG-RETRIEVEPOSEXTRA will also be used. The maximum number of entries that can be returned in response to a single UBX-LOG-RETRIEVE message is 256. If more entries than this are required the message will need to be sent multiple times with different startNumbers. The retrieve will be stopped if any UBX-LOG message is received. The speed of transfer can be maximized by using a high data rate and temporarily stopping the GPS processing (see UBX-CFG-RST).

Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x21	0x09	12	*see below*	CK_A CK_B

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	U4	startNumber	-	-	Index of first log entry to be transferred. If it is larger than the index of the last available log entry, then the first log entry to be transferred is the last available log entry. The indexing of log entries is zero-based.

4	U4	`entryCount`	-	-	Number of log entries to transfer in total including the first entry to be transferred. If it is larger than the log entries available starting from the first entry to be transferred, then only the available log entries are transferred followed by a UBX-ACK-NAK. The maximum is 256.
8	U1	`version`	-	-	Message version (0x00 for this version)
9	U1[3]	`reserved0`	-	-	Reserved

3.12.6 UBX-LOG-RETRIEVEPOS (0x21 0x0b)

3.12.6.1 Position fix log entry

Message	**UBX-LOG-RETRIEVEPOS** **Position fix log entry**				
Type	Output				
Comment	This message is used to report a position fix log entry				
Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x21	0x0b	40	*see below*	CK_A CK_B

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	U4	`entryIndex`	-	-	The index of this log entry
4	I4	`lon`	1e-7	deg	Longitude
8	I4	`lat`	1e-7	deg	Latitude
12	I4	`hMSL`	-	mm	Height above mean sea level
16	U4	`hAcc`	-	mm	Horizontal accuracy estimate
20	U4	`gSpeed`	-	mm/s	Ground speed (2-D)
24	U4	`heading`	1e-5	deg	Heading
28	U1	`version`	-	-	Message version (0x00 for this version)
29	U1	`fixType`	-	-	Fix type: • 0x01 = Dead Reckoning only • 0x02 = 2D-Fix • 0x03 = 3D-Fix • 0x04 = GNSS + Dead Reckoning combined
30	U2	`year`	-	-	Year (1-65635) of UTC time
32	U1	`month`	-	-	Month (1-12) of UTC time
33	U1	`day`	-	-	Day (1-31) of UTC time
34	U1	`hour`	-	-	Hour (0-23) of UTC time
35	U1	`minute`	-	-	Minute (0-59) of UTC time
36	U1	`second`	-	-	Second (0-60) of UTC time
37	U1	`reserved0`	-	-	Reserved
38	U1	`numSV`	-	-	Number of satellites used in the position fix
39	U1	`reserved1`	-	-	Reserved

3.12.7 UBX-LOG-RETRIEVEPOSEXTRA (0x21 0x0f)

3.12.7.1 Odometer log entry

Message	**UBX-LOG-RETRIEVEPOSEXTRA**				
	Odometer log entry				
Type	Output				
Comment	This message is used to report an odometer log entry				
Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x21	0x0f	32	*see below*	CK_A CK_B

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	U4	entryIndex	-	-	The index of this log entry
4	U1	version	-	-	Message version (0x00 for this version)
5	U1	reserved0	-	-	Reserved
6	U2	year	-	-	Year (1-65635) of UTC time. Will be zero if time not known
8	U1	month	-	-	Month (1-12) of UTC time
9	U1	day	-	-	Day (1-31) of UTC time
10	U1	hour	-	-	Hour (0-23) of UTC time
11	U1	minute	-	-	Minute (0-59) of UTC time
12	U1	second	-	-	Second (0-60) of UTC time
13	U1[3]	reserved1	-	-	Reserved
16	U4	distance	-	-	Odometer distance traveled since the last time the odometer was reset by a UBX-NAV-RESETODO
20	U1[12]	reserved2	-	-	Reserved

3.12.8 UBX-LOG-RETRIEVESTRING (0x21 0x0d)

3.12.8.1 Byte string log entry

Message	**UBX-LOG-RETRIEVESTRING**					
	Byte string log entry					
Type	Output					
Comment	This message is used to report a byte string log entry					
Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x21	0x0d	16 + byteCount	*see below*	CK_A CK_B

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	U4	entryIndex	-	-	The index of this log entry
4	U1	version	-	-	Message version (0x00 for this version)
5	U1	reserved0	-	-	Reserved
6	U2	year	-	-	Year (1-65635) of UTC time. Will be zero if time not known
8	U1	month	-	-	Month (1-12) of UTC time
9	U1	day	-	-	Day (1-31) of UTC time
10	U1	hour	-	-	Hour (0-23) of UTC time
11	U1	minute	-	-	Minute (0-59) of UTC time

12	U1	second	-	-	Second (0-60) of UTC time
13	U1	reserved1	-	-	Reserved
14	U2	byteCount	-	-	Size of string in bytes
Start of repeated group (byteCount times)					
16 + n	U1	bytes	-	-	The bytes of the string
End of repeated group (byteCount times)					

3.12.9 UBX-LOG-STRING (0x21 0x04)

3.12.9.1 Store arbitrary string in on-board flash

Message	**UBX-LOG-STRING** **Store arbitrary string in on-board flash**				
Type	Command				
Comment	This message can be used to store an arbitrary byte string in the on-board flash memory. The maximum length that can be stored is 256 bytes.				
Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x21	0x04	[0..n]	*see below*	CK_A CK_B

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
Start of repeated group (N times)					
0 + n	U1	bytes	-	-	The string of bytes to be logged (maximum 256)
End of repeated group (N times)					

3.13 UBX-MGA (0x13)

The messages in the UBX-MGA class are used for sending GNSS assistance (A-GNSS, aiding) information to the receiver as well as backing up the navigation database from the receiver to a host.

3.13.1 UBX-MGA-ACK (0x13 0x60)

3.13.1.1 Multiple GNSS acknowledge message

Message	**UBX-MGA-ACK-DATA0** **Multiple GNSS acknowledge message**					
Type	Output					
Comment	This message is sent by a u-blox receiver to acknowledge the receipt of an assistance message. Acknowledgments are enabled by setting the CFG-NAVSPG-ACKAIDING item. See section Flow control in the integration manual for details.					
Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x13	0x60	8	*see below*	CK_A CK_B

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	U1	type	-	-	Type of acknowledgment: • 0 = The message was not used by the receiver (see infoCode field for an indication of why) • 1 = The message was accepted for use by the receiver (the infoCode field will be 0)
1	U1	version	-	-	Message version (0x00 for this version)

2	U1	`infoCode`	-	-	Provides greater information on what the receiver chose to do with the message contents: • 0 = The receiver accepted the data • 1 = The receiver does not know the time so it cannot use the data (To resolve this a UBX-MGA-INI-TIME_UTC message should be supplied first) • 2 = The message version is not supported by the receiver • 3 = The message size does not match the message version • 4 = The message data could not be stored to the database • 5 = The receiver is not ready to use the message data • 6 = The message type is unknown
3	U1	`msgId`	-	-	UBX message ID of the acknowledged message
4	U1[4]	`msgPayloadStart`	-	-	The first 4 bytes of the acknowledged message's payload

3.13.2 UBX-MGA-BDS (0x13 0x03)

3.13.2.1 BeiDou ephemeris assistance

Message	**UBX-MGA-BDS-EPH** **BeiDou ephemeris assistance**
Type	Input
Comment	This message allows the delivery of BeiDou ephemeris assistance to a receiver. See section AssistNow online in the integration manual for details.

Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x13	0x03	88	*see below*	CK_A CK_B

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	U1	`type`	-	-	Message type (0x01 for this type)
1	U1	`version`	-	-	Message version (0x00 for this version)
2	U1	`svId`	-	-	BeiDou satellite identifier (see Satellite Numbering)
3	U1	`reserved0`	-	-	Reserved
4	U1	`SatH1`	-	-	Autonomous satellite Health flag
5	U1	`IODC`	-	-	Issue of Data, Clock
6	I2	`a2`	2^-66	s/s^2	Time polynomial coefficient 2
8	I4	`a1`	2^-50	s/s	Time polynomial coefficient 1
12	I4	`a0`	2^-33	s	Time polynomial coefficient 0
16	U4	`toc`	2^3	s	Clock data reference time
20	I2	`TGD1`	0.1	ns	Equipment Group Delay Differential
22	U1	`URAI`	-	-	User Range Accuracy Index
23	U1	`IODE`	-	-	Issue of Data, Ephemeris
24	U4	`toe`	2^3	s	Ephemeris reference time
28	U4	`sqrtA`	2^-19	m^0.5	Square root of semi-major axis
32	U4	`e`	2^-33	-	Eccentricity

36	I4	omega	2^-31	semi-circles	Argument of perigee
40	I2	Deltan	2^-43	semi-circles/s	Mean motion difference from computed value
42	I2	IDOT	2^-43	semi-circles/s	Rate of inclination angle
44	I4	M0	2^-31	semi-circles	Mean anomaly at reference time
48	I4	Omega0	2^-31	semi-circles	Longitude of ascending node of orbital of plane computed according to reference time
52	I4	OmegaDot	2^-43	semi-circles/s	Rate of right ascension
56	I4	i0	2^-31	semi-circles	Inclination angle at reference time
60	I4	Cuc	2^-31	radians	Amplitude of cosine harmonic correction term to the argument of latitude
64	I4	Cus	2^-31	radians	Amplitude of sine harmonic correction term to the argument of latitude
68	I4	Crc	2^-6	m	Amplitude of cosine harmonic correction term to the orbit radius
72	I4	Crs	2^-6	m	Amplitude of sine harmonic correction term to the orbit radius
76	I4	Cic	2^-31	radians	Amplitude of cosine harmonic correction term to the angle of inclination
80	I4	Cis	2^-31	radians	Amplitude of sine harmonic correction term to the angle of inclination
84	U1[4]	reserved1	-	-	Reserved

3.13.2.2 BeiDou almanac assistance

Message	**UBX-MGA-BDS-ALM** **BeiDou almanac assistance**					
Type	Input					
Comment	This message allows the delivery of BeiDou almanac assistance to a receiver. See section AssistNow online in the integration manual for details.					
Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x13	0x03	40	*see below*	CK_A CK_B

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	U1	type	-	-	Message type (0x02 for this version)
1	U1	version	-	-	Message version (0x00 for this version)
2	U1	svId	-	-	BeiDou satellite identifier (see Satellite Numbering)
3	U1	reserved0	-	-	Reserved
4	U1	Wna	-	week	Almanac Week Number
5	U1	toa	2^12	s	Almanac reference time
6	I2	deltaI	2^-19	semi-circles	Almanac correction of orbit reference inclination at reference time
8	U4	sqrtA	2^-11	m^0.5	Almanac square root of semi-major axis
12	U4	e	2^-21	-	Almanac eccentricity

16	I4	omega	2^-23	semi-circles	Almanac argument of perigee
20	I4	M0	2^-23	semi-circles	Almanac mean anomaly at reference time
24	I4	Omega0	2^-23	semi-circles	Almanac longitude of ascending node of orbit plane at computed according to reference time
28	I4	omegaDot	2^-38	semi-circles/s	Almanac rate of right ascension
32	I2	a0	2^-20	s	Almanac satellite clock bias
34	I2	a1	2^-38	s/s	Almanac satellite clock rate
36	U1[4]	reserved1	-	-	Reserved

3.13.2.3 BeiDou health assistance

Message	**UBX-MGA-BDS-HEALTH** **BeiDou health assistance**
Type	Input
Comment	This message allows the delivery of BeiDou health assistance to a receiver. See section AssistNow online in the integration manual for details.

Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x13	0x03	68	*see below*	CK_A CK_B

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	U1	type	-	-	Message type (0x04 for this type)
1	U1	version	-	-	Message version (0x00 for this version)
2	U1[2]	reserved0	-	-	Reserved
4	U2[30]	healthCode	-	-	Each two-byte value represents a BeiDou SV (1-30). The 9 LSBs of each byte contain the 9 bit health code from subframe 5 pages 7,8 of the D1 message, and from subframe 5 pages 35,36 of the D1 message.
64	U1[4]	reserved1	-	-	Reserved

3.13.2.4 BeiDou UTC assistance

Message	**UBX-MGA-BDS-UTC** **BeiDou UTC assistance**
Type	Input
Comment	This message allows the delivery of BeiDou UTC assistance to a receiver. See section AssistNow online in the integration manual for details.

Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x13	0x03	20	*see below*	CK_A CK_B

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	U1	type	-	-	Message type (0x05 for this type)
1	U1	version	-	-	Message version (0x00 for this version)
2	U1[2]	reserved0	-	-	Reserved
4	I4	a0UTC	2^-30	s	BDT clock bias relative to UTC
8	I4	a1UTC	2^-50	s/s	BDT clock rate relative to UTC

12	I1	`dtLS`	-	s	Delta time due to leap seconds before the new leap second effective
13	U1	`reserved1`	-	-	Reserved
14	U1	`wnRec`	-	week	BeiDou week number of reception of this UTC parameter set (8-bit truncated)
15	U1	`wnLSF`	-	week	Week number of the new leap second
16	U1	`dN`	-	day	Day number of the new leap second
17	I1	`dtLSF`	-	s	Delta time due to leap seconds after the new leap second effective
18	U1[2]	`reserved2`	-	-	Reserved

3.13.2.5 BeiDou ionosphere assistance

Message	**UBX-MGA-BDS-IONO** **BeiDou ionosphere assistance**
Type	Input
Comment	This message allows the delivery of BeiDou ionospheric assistance to a receiver. See section AssistNow online in the integration manual for details.

Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x13	0x03	16	*see below*	CK_A CK_B

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	U1	`type`	-	-	Message type (0x06 for this type)
1	U1	`version`	-	-	Message version (0x00 for this version)
2	U1[2]	`reserved0`	-	-	Reserved
4	I1	`alpha0`	2^-30	s	Ionospheric parameter alpha0
5	I1	`alpha1`	2^-27	s/pi	Ionospheric parameter alpha1
6	I1	`alpha2`	2^-24	s/pi^2	Ionospheric parameter alpha2
7	I1	`alpha3`	2^-24	s/pi^3	Ionospheric parameter alpha3
8	I1	`beta0`	2^11	s	Ionospheric parameter beta0
9	I1	`beta1`	2^14	s/pi	Ionospheric parameter beta1
10	I1	`beta2`	2^16	s/pi^2	Ionospheric parameter beta2
11	I1	`beta3`	2^16	s/pi^3	Ionospheric parameter beta3
12	U1[4]	`reserved1`	-	-	Reserved

3.13.3 UBX-MGA-DBD (0x13 0x80)

3.13.3.1 Poll the navigation database

Message	**UBX-MGA-DBD** **Poll the navigation database**
Type	Poll request
Comment	Poll the whole navigation data base. The receiver will send all available data from its internal database. The receiver will indicate the finish of the transmission with a UBX-MGA-ACK. The msgPayloadStart field of the UBX-MGA-ACK message will contain a U4 representing the number of UBX-MGA-DBD-DATA* messages sent.

Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x13	0x80	0	*see below*	CK_A CK_B

Payload	This message has no payload.

3.13.3.2 Navigation database dump entry

Message	**UBX-MGA-DBD** **Navigation database dump entry**
Type	Input/output
Comment	Navigation database entry. The data fields are firmware-specific. Transmission of this type of message will be acknowledged by UBX-MGA-ACK messages, if acknowledgment has been enabled. See section AssistNow online in the integration manual for details. The maximum payload size for firmware 2.01 onwards is 164 bytes (which makes the maximum message size 172 bytes). ☞ UBX-MGA-DBD messages are only intended to be sent back to the same receiver that generated them.

Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x13	0x80	12 + [0..n]	*see below*	CK_A CK_B

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	U1[12]	`reserved0`	-	-	Reserved
Start of repeated group (N times)					
12 + n	U1	`data`	-	-	firmware-specific data
End of repeated group (N times)					

3.13.4 UBX-MGA-GAL (0x13 0x02)

3.13.4.1 Galileo ephemeris assistance

Message	**UBX-MGA-GAL-EPH** **Galileo ephemeris assistance**
Type	Input
Comment	This message allows the delivery of Galileo ephemeris assistance to a receiver. See section AssistNow online in the integration manual for details.

Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x13	0x02	76	*see below*	CK_A CK_B

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	U1	`type`	-	-	Message type (0x01 for this type)
1	U1	`version`	-	-	Message version (0x00 for this version)
2	U1	`svId`	-	-	Galileo Satellite identifier (see Satellite Numbering)
3	U1	`reserved0`	-	-	Reserved
4	U2	`iodNav`	-	-	Ephemeris and clock correction Issue of Data
6	I2	`deltaN`	2^-43	semi-circles/s	Mean motion difference from computed value
8	I4	`m0`	2^-31	semi-circles	Mean anomaly at reference time
12	U4	`e`	2^-33	-	Eccentricity
16	U4	`sqrtA`	2^-19	m^0.5	Square root of the semi-major axis
20	I4	`omega0`	2^-31	semi-circles	Longitude of ascending node of orbital plane at weekly epoch
24	I4	`i0`	2^-31	semi-circles	Inclination angle at reference time

28	I4	omega	2^-31	semi-circles	Argument of perigee
32	I4	omegaDot	2^-43	semi-circles/s	Rate of change of right ascension
36	I2	iDot	2^-43	semi-circles/s	Rate of change of inclination angle
38	I2	cuc	2^-29	radians	Amplitude of the cosine harmonic correction term to the argument of latitude
40	I2	cus	2^-29	radians	Amplitude of the sine harmonic correction term to the argument of latitude
42	I2	crc	2^-5	radians	Amplitude of the cosine harmonic correction term to the orbit radius
44	I2	crs	2^-5	radians	Amplitude of the sine harmonic correction term to the orbit radius
46	I2	cic	2^-29	radians	Amplitude of the cosine harmonic correction term to the angle of inclination
48	I2	cis	2^-29	radians	Amplitude of the sine harmonic correction term to the angle of inclination
50	U2	toe	60	s	Ephemeris reference time
52	I4	af0	2^-34	s	SV clock bias correction coefficient
56	I4	af1	2^-46	s/s	SV clock drift correction coefficient
60	I1	af2	2^-59	s/s squared	SV clock drift rate correction coefficient
61	U1	sisaIndexE1E5b	-	-	Signal-In-Space Accuracy index for dual frequency E1-E5b
62	U2	toc	60	s	Clock correction data reference Time of Week
64	I2	bgdE1E5b	2^-32	s	E1-E5b Broadcast Group Delay
66	U1[2]	reserved1	-	-	Reserved
68	U1	healthE1B	-	-	E1-B Signal Health Status
69	U1	dataValidityE1B	-	-	E1-B Data Validity Status
70	U1	healthE5b	-	-	E5b Signal Health Status
71	U1	dataValidityE5b	-	-	E5b Data Validity Status
72	U1[4]	reserved2	-	-	Reserved

3.13.4.2 Galileo almanac assistance

Message	**UBX-MGA-GAL-ALM** **Galileo almanac assistance**					
Type	Input					
Comment	This message allows the delivery of Galileo almanac assistance to a receiver. See section AssistNow online in the integration manual for details.					
Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x13	0x02	32	*see below*	CK_A CK_B

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	U1	type	-	-	Message type (0x02 for this type)
1	U1	version	-	-	Message version (0x00 for this version)

2	U1	`svId`	-	-	Galileo Satellite identifier (see Satellite Numbering)
3	U1	`reserved0`	-	-	Reserved
4	U1	`ioda`	-	-	Almanac Issue of Data
5	U1	`almWNa`	-	week	Almanac reference week number
6	U2	`toa`	600	s	Almanac reference time
8	I2	`deltaSqrtA`	2^-9	m^0.5	Difference with respect to the square root of the nominal semi-major axis (29 600 km)
10	U2	`e`	2^-16	-	Eccentricity
12	I2	`deltaI`	2^-14	semi-circles	Inclination at reference time relative to i0 = 56 degree
14	I2	`omega0`	2^-15	semi-circles	Longitude of ascending node of orbital plane at weekly epoch
16	I2	`omegaDot`	2^-33	semi-circles/s	Rate of change of right ascension
18	I2	`omega`	2^-15	semi-circles	Argument of perigee
20	I2	`m0`	2^-15	semi-circles	Satellite mean anomaly at reference time
22	I2	`af0`	2^-19	s	Satellite clock correction bias 'truncated'
24	I2	`af1`	2^-38	s/s	Satellite clock correction linear 'truncated'
26	U1	`healthE1B`	-	-	Satellite E1-B signal health status
27	U1	`healthE5b`	-	-	Satellite E5b signal health status
28	U1[4]	`reserved1`	-	-	Reserved

3.13.4.3 Galileo GPS time offset assistance

Message	**UBX-MGA-GAL-TIMEOFFSET** **Galileo GPS time offset assistance**
Type	Input
Comment	This message allows the delivery of Galileo time to GPS time offset. See section AssistNow online in the integration manual for details.

Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x13	0x02	12	*see below*	CK_A CK_B

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	U1	`type`	-	-	Message type (0x03 for this type)
1	U1	`version`	-	-	Message version (0x00 for this version)
2	U1[2]	`reserved0`	-	-	Reserved
4	I2	`a0G`	2^-35	s	Constant term of the polynomial describing the offset
6	I2	`a1G`	2^-51	s/s	Rate of change of the offset
8	U1	`t0G`	3600	s	Reference time for GGTO data
9	U1	`wn0G`	-	weeks	Week Number of GGTO reference
10	U1[2]	`reserved1`	-	-	Reserved

3.13.4.4 Galileo UTC assistance

Message	**UBX-MGA-GAL-UTC** **Galileo UTC assistance**				
Type	Input				
Comment	This message allows the delivery of Galileo UTC assistance to a receiver. See section AssistNow online in the integration manual for details.				
Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x13	0x02	20	*see below*	CK_A CK_B

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	U1	`type`	-	-	Message type (0x05 for this type)
1	U1	`version`	-	-	Message version (0x00 for this version)
2	U1[2]	`reserved0`	-	-	Reserved
4	I4	`a0`	2^-30	s	First parameter of UTC polynomial
8	I4	`a1`	2^-50	s/s	Second parameter of UTC polynomial
12	I1	`dtLS`	-	s	Delta time due to current leap seconds
13	U1	`tot`	3600	s	UTC parameters reference time of week (Galileo time)
14	U1	`wnt`	-	weeks	UTC parameters reference week number (the 8-bit WNt field)
15	U1	`wnLSF`	-	weeks	Week number at the end of which the future leap second becomes effective (the 8-bit WNLSF field)
16	U1	`dN`	-	days	Day number at the end of which the future leap second becomes effective
17	I1	`dTLSF`	-	s	Delta time due to future leap seconds
18	U1[2]	`reserved1`	-	-	Reserved

3.13.5 UBX-MGA-GLO (0x13 0x06)

3.13.5.1 GLONASS ephemeris assistance

Message	**UBX-MGA-GLO-EPH** **GLONASS ephemeris assistance**				
Type	Input				
Comment	This message allows the delivery of GLONASS ephemeris assistance to a receiver. See section AssistNow online in the integration manual for details.				
Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x13	0x06	48	*see below*	CK_A CK_B

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	U1	`type`	-	-	Message type (0x01 for this type)
1	U1	`version`	-	-	Message version (0x00 for this version)
2	U1	`svId`	-	-	GLONASS Satellite identifier (see Satellite Numbering)
3	U1	`reserved0`	-	-	Reserved
4	U1	`FT`	-	-	User range accuracy
5	U1	`B`	-	-	Health flag from string 2

6	U1	M	-	-	Type of GLONASS satellite (1 indicates GLONASS-M)
7	I1	H	-	-	Carrier frequency number of navigation RF signal, Range=(-7 .. 6), -128 for unknown
8	I4	x	2^-11	km	X component of the SV position in PZ-90.02 coordinate System
12	I4	y	2^-11	km	Y component of the SV position in PZ-90.02 coordinate System
16	I4	z	2^-11	km	Z component of the SV position in PZ-90.02 coordinate System
20	I4	dx	2^-20	km/s	X component of the SV velocity in PZ-90.02 coordinate System
24	I4	dy	2^-20	km/s	Y component of the SV velocity in PZ-90.02 coordinate System
28	I4	dz	2^-20	km/s	Z component of the SV velocity in PZ-90.02 coordinate System
32	I1	ddx	2^-30	km/s^2	X component of the SV acceleration in PZ-90.02 coordinate System
33	I1	ddy	2^-30	km/s^2	Y component of the SV acceleration in PZ-90.02 coordinate System
34	I1	ddz	2^-30	km/s^2	Z component of the SV acceleration in PZ-90.02 coordinate System
35	U1	tb	15	minutes	Index of a time interval within current day according to UTC(SU)
36	I2	gamma	2^-40	-	Relative carrier frequency deviation
38	U1	E	-	days	Ephemeris data age indicator
39	I1	deltaTau	2^-30	s	Time difference between L2 and L1 band
40	I4	tau	2^-30	s	SV clock bias
44	U1[4]	reserved1	-	-	Reserved

3.13.5.2 GLONASS almanac assistance

Message	**UBX-MGA-GLO-ALM** **GLONASS almanac assistance**				
Type	Input				
Comment	This message allows the delivery of GLONASS almanac assistance to a receiver. See section AssistNow online in the integration manual for details.				
Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x13	0x06	36	*see below*	CK_A CK_B

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	U1	type	-	-	Message type (0x02 for this type)
1	U1	version	-	-	Message version (0x00 for this version)
2	U1	svId	-	-	GLONASS Satellite identifier (see Satellite Numbering)
3	U1	reserved0	-	-	Reserved
4	U2	N	-	days	Reference calender day number of almanac within the four-year period (from string 5)
6	U1	M	-	-	Type of GLONASS satellite (1 indicates GLONASS-M)

7	U1	C	-	-	Unhealthy flag at instant of almanac upload (1 indicates operability of satellite)
8	I2	tau	2^-18	s	Coarse time correction to GLONASS time
10	U2	epsilon	2^-20	-	Eccentricity
12	I4	lambda	2^-20	semi-circles	Longitude of the first (within the N-day) ascending node of satellite orbit in PC-90.02 coordinate system
16	I4	deltaI	2^-20	semi-circles	Correction to the mean value of inclination
20	U4	tLambda	2^-5	s	Time of the first ascending node passage
24	I4	deltaT	2^-9	s/orbital-period	Correction to the mean value of Draconian period
28	I1	deltaDT	2^-14	s/orbital-period^2	Rate of change of Draconian period
29	I1	H	-	-	Carrier frequency number of navigation RF signal, Range=(-7 .. 6)
30	I2	omega	-	-	Argument of perigee
32	U1[4]	reserved1	-	-	Reserved

3.13.5.3 GLONASS auxiliary time offset assistance

Message	**UBX-MGA-GLO-TIMEOFFSET** **GLONASS auxiliary time offset assistance**
Type	Input
Comment	This message allows the delivery of auxiliary GLONASS assistance (including the GLONASS time offsets to other GNSS systems) to a receiver. See section AssistNow online in the integration manual for details.

Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x13	0x06	20	*see below*	CK_A CK_B

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	U1	type	-	-	Message type (0x03 for this type)
1	U1	version	-	-	Message version (0x00 for this version)
2	U2	N	-	days	Reference calendar day number within the four-year period of almanac (from string 5)
4	I4	tauC	2^-27	s	Time scale correction to UTC(SU) time
8	I4	tauGps	2^-31	s	Correction to GPS time relative to GLONASS time
12	I2	B1	2^-10	s	Coefficient to determine delta UT1
14	I2	B2	2^-16	s/msd	Rate of change of delta UT1
16	U1[4]	reserved0	-	-	Reserved

3.13.6 UBX-MGA-GPS (0x13 0x00)

3.13.6.1 GPS ephemeris assistance

Message	**UBX-MGA-GPS-EPH** **GPS ephemeris assistance**
Type	Input
Comment	This message allows the delivery of GPS ephemeris assistance to a receiver. See section AssistNow online in the integration manual for details.

Message structure	Header	Class	ID	Length (Bytes)	Payload	Checksum
	0xb5 0x62	0x13	0x00	68	see below	CK_A CK_B

Payload description:

Byte offset	Type	Name	Scale	Unit	Description
0	U1	type	-	-	Message type (0x01 for this type)
1	U1	version	-	-	Message version (0x00 for this version)
2	U1	svId	-	-	GPS Satellite identifier (see Satellite Numbering)
3	U1	reserved0	-	-	Reserved
4	U1	fitInterval	-	-	Fit interval flag
5	U1	uraIndex	-	-	URA index
6	U1	svHealth	-	-	SV health
7	I1	tgd	2^-31	s	Group delay differential
8	U2	iodc	-	-	IODC
10	U2	toc	2^4	s	Clock data reference time
12	U1	reserved1	-	-	Reserved
13	I1	af2	2^-55	s/s squared	Time polynomial coefficient 2
14	I2	af1	2^-43	s/s	Time polynomial coefficient 1
16	I4	af0	2^-31	s	Time polynomial coefficient 0
20	I2	crs	2^-5	m	Crs
22	I2	deltaN	2^-43	semi-circles/s	Mean motion difference from computed value
24	I4	m0	2^-31	semi-circles	Mean anomaly at reference time
28	I2	cuc	2^-29	radians	Amplitude of cosine harmonic correction term to argument of latitude
30	I2	cus	2^-29	radians	Amplitude of sine harmonic correction term to argument of latitude
32	U4	e	2^-33	-	Eccentricity
36	U4	sqrtA	2^-19	m^0.5	Square root of the semi-major axis
40	U2	toe	2^4	s	Reference time of ephemeris
42	I2	cic	2^-29	radians	Amplitude of cos harmonic correction term to angle of inclination
44	I4	omega0	2^-31	semi-circles	Longitude of ascending node of orbit plane at weekly epoch
48	I2	cis	2^-29	radians	Amplitude of sine harmonic correction term to angle of inclination
50	I2	crc	2^-5	m	Amplitude of cosine harmonic correction term to orbit radius
52	I4	i0	2^-31	semi-circles	Inclination angle at reference time
56	I4	omega	2^-31	semi-circles	Argument of perigee
60	I4	omegaDot	2^-43	semi-circles/s	Rate of right ascension
64	I2	idot	2^-43	semi-circles/s	Rate of inclination angle

66	U1[2]	reserved2	-	-	Reserved

3.13.6.2 GPS almanac assistance

Message	**UBX-MGA-GPS-ALM** **GPS almanac assistance**					
Type	Input					
Comment	This message allows the delivery of GPS almanac assistance to a receiver. See section AssistNow online in the integration manual for details.					
Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x13	0x00	36	*see below*	CK_A CK_B

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	U1	type	-	-	Message type (0x02 for this type)
1	U1	version	-	-	Message version (0x00 for this version)
2	U1	svId	-	-	GPS Satellite identifier (see Satellite Numbering)
3	U1	svHealth	-	-	SV health information
4	U2	e	2^-21	-	Eccentricity
6	U1	almWNa	-	week	Reference week number of almanac (the 8-bit WNa field)
7	U1	toa	2^12	s	Reference time of almanac
8	I2	deltaI	2^-19	semi-circles	Delta inclination angle at reference time
10	I2	omegaDot	2^-38	semi-circles/s	Rate of right ascension
12	U4	sqrtA	2^-11	m^0.5	Square root of the semi-major axis
16	I4	omega0	2^-23	semi-circles	Longitude of ascending node of orbit plane
20	I4	omega	2^-23	semi-circles	Argument of perigee
24	I4	m0	2^-23	semi-circles	Mean anomaly at reference time
28	I2	af0	2^-20	s	Time polynomial coefficient 0 (8 MSBs)
30	I2	af1	2^-38	s/s	Time polynomial coefficient 1
32	U1[4]	reserved0	-	-	Reserved

3.13.6.3 GPS health assistance

Message	**UBX-MGA-GPS-HEALTH** **GPS health assistance**					
Type	Input					
Comment	This message allows the delivery of GPS health assistance to a receiver. See section AssistNow online in the integration manual for details.					
Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x13	0x00	40	*see below*	CK_A CK_B

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	U1	type	-	-	Message type (0x04 for this type)
1	U1	version	-	-	Message version (0x00 for this version)

2	U1[2]	reserved0	-	-	Reserved
4	U1[32]	healthCode	-	-	Each byte represents a GPS SV (1-32). The 6 LSBs of each byte contains the 6 bit health code from subframes 4/5 page 25.
36	U1[4]	reserved1	-	-	Reserved

3.13.6.4 GPS UTC assistance

Message	**UBX-MGA-GPS-UTC** **GPS UTC assistance**
Type	Input
Comment	This message allows the delivery of GPS UTC assistance to a receiver. See section AssistNow online in the integration manual for details.

Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x13	0x00	20	*see below*	CK_A CK_B

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	U1	type	-	-	Message type (0x05 for this type)
1	U1	version	-	-	Message version (0x00 for this version)
2	U1[2]	reserved0	-	-	Reserved
4	I4	utcA0	2^-30	s	First parameter of UTC polynomial
8	I4	utcA1	2^-50	s/s	Second parameter of UTC polynomial
12	I1	utcDtLS	-	s	Delta time due to current leap seconds
13	U1	utcTot	2^12	s	UTC parameters reference time of week (GPS time)
14	U1	utcWNt	-	weeks	UTC parameters reference week number (the 8-bit WNt field)
15	U1	utcWNlsf	-	weeks	Week number at the end of which the future leap second becomes effective (the 8-bit WNLSF field)
16	U1	utcDn	-	days	Day number at the end of which the future leap second becomes effective
17	I1	utcDtLSF	-	s	Delta time due to future leap seconds
18	U1[2]	reserved1	-	-	Reserved

3.13.6.5 GPS ionosphere assistance

Message	**UBX-MGA-GPS-IONO** **GPS ionosphere assistance**
Type	Input
Comment	This message allows the delivery of GPS ionospheric assistance to a receiver. See section AssistNow online in the integration manual for details.

Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x13	0x00	16	*see below*	CK_A CK_B

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	U1	type	-	-	Message type (0x06 for this type)
1	U1	version	-	-	Message version (0x00 for this version)
2	U1[2]	reserved0	-	-	Reserved
4	I1	ionoAlpha0	2^-30	s	Ionospheric parameter alpha0 [s]

5	I1	`ionoAlpha1`	2^-27	s/semi-circle	Ionospheric parameter alpha1 [s/semi-circle]
6	I1	`ionoAlpha2`	2^-24	s/(semi-circle^2)	Ionospheric parameter alpha2 [s/semi-circle^2]
7	I1	`ionoAlpha3`	2^-24	s/(semi-circle^3)	Ionospheric parameter alpha3 [s/semi-circle^3]
8	I1	`ionoBeta0`	2^11	s	Ionospheric parameter beta0 [s]
9	I1	`ionoBeta1`	2^14	s/semi-circle	Ionospheric parameter beta1 [s/semi-circle]
10	I1	`ionoBeta2`	2^16	s/(semi-circle^2)	Ionospheric parameter beta2 [s/semi-circle^2]
11	I1	`ionoBeta3`	2^16	s/(semi-circle^3)	Ionospheric parameter beta3 [s/semi-circle^3]
12	U1[4]	`reserved1`	-	-	Reserved

3.13.7 UBX-MGA-INI (0x13 0x40)

3.13.7.1 Initial position assistance

Message	**UBX-MGA-INI-POS_XYZ** **Initial position assistance**
Type	Input
Comment	This message allows the delivery of initial position assistance to a receiver in cartesian ECEF coordinates. This message is equivalent to the UBX-MGA-INI-POS_LLH message, except for the coordinate system. See section AssistNow Online in the integration manual for details. ☞ Supplying position assistance that is inaccurate by more than the specified position accuracy, may lead to substantially degraded receiver performance.

Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x13	0x40	20	*see below*	CK_A CK_B

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	U1	`type`	-	-	Message type (0x00 for this type)
1	U1	`version`	-	-	Message version (0x00 for this version)
2	U1[2]	`reserved0`	-	-	Reserved
4	I4	`ecefX`	-	cm	WGS84 ECEF X coordinate
8	I4	`ecefY`	-	cm	WGS84 ECEF Y coordinate
12	I4	`ecefZ`	-	cm	WGS84 ECEF Z coordinate
16	U4	`posAcc`	-	cm	Position accuracy (stddev)

3.13.7.2 Initial position assistance

Message	**UBX-MGA-INI-POS_LLH** **Initial position assistance**
Type	Input
Comment	This message allows the delivery of initial position assistance to a receiver in WGS84 lat/long/alt coordinates. This message is equivalent to the UBX-MGA-INI-POS_XYZ message, except for the coordinate system. See section AssistNow online in the integration manual for details. ☞ Supplying position assistance that is inaccurate by more than the specified position accuracy, may lead to substantially degraded receiver performance.

Message structure	Header	Class	ID	Length (Bytes)	Payload	Checksum
	0xb5 0x62	0x13	0x40	20	see below	CK_A CK_B

Payload description:

Byte offset	Type	Name	Scale	Unit	Description
0	U1	type	-	-	Message type (0x01 for this type)
1	U1	version	-	-	Message version (0x00 for this version)
2	U1[2]	reserved0	-	-	Reserved
4	I4	lat	1e-7	deg	WGS84 Latitude
8	I4	lon	1e-7	deg	WGS84 Longitude
12	I4	alt	-	cm	WGS84 Altitude
16	U4	posAcc	-	cm	Position accuracy (stddev)

3.13.7.3 Initial time assistance

Message	**UBX-MGA-INI-TIME_UTC** **Initial time assistance**
Type	Input
Comment	This message allows the delivery of UTC time assistance to a receiver. This message is equivalent to the UBX-MGA-INI-TIME_GNSS message, except for the time base. See section AssistNow online in the integration manual for details. ☞ Supplying time assistance that is inaccurate by more than the specified time accuracy, may lead to substantially degraded receiver performance.

Message structure	Header	Class	ID	Length (Bytes)	Payload	Checksum
	0xb5 0x62	0x13	0x40	24	see below	CK_A CK_B

Payload description:

Byte offset	Type	Name	Scale	Unit	Description
0	U1	type	-	-	Message type (0x10 for this type)
1	U1	version	-	-	Message version (0x00 for this version)
2	X1	ref	-	-	Reference to be used to set time
bits 3…0	U:4	source	-	-	• 0 = none, i.e. on receipt of message (will be inaccurate!) • 1 = relative to pulse sent to EXTINT0 • 2 = relative to pulse sent to EXTINT1 • 3-15 = reserved
bit 4	U:1	fall	-	-	use falling edge of EXTINT pulse (default rising) - only if source is EXTINT
bit 5	U:1	last	-	-	use last EXTINT pulse (default next pulse) - only if source is EXTINT
3	I1	leapSecs	-	s	Number of leap seconds since 1980 (or 0x80 = -128 if unknown)
4	U2	year	-	-	Year
6	U1	month	-	-	Month, starting at 1
7	U1	day	-	-	Day, starting at 1
8	U1	hour	-	-	Hour, from 0 to 23
9	U1	minute	-	-	Minute, from 0 to 59
10	U1	second	-	s	Seconds, from 0 to 59
11	U1	reserved0	-	-	Reserved

12	U4	`ns`	-	ns	Nanoseconds, from 0 to 999,999,999
16	U2	`tAccS`	-	s	Seconds part of time accuracy
18	U1[2]	`reserved1`	-	-	Reserved
20	U4	`tAccNs`	-	ns	Nanoseconds part of time accuracy, from 0 to 999,999,999

3.13.7.4 Initial time assistance

Message	**UBX-MGA-INI-TIME_GNSS** **Initial time assistance**
Type	Input
Comment	This message allows the delivery of time assistance to a receiver in a chosen GNSS timebase. This message is equivalent to the UBX-MGA-INI-TIME_UTC message, except for the time base. See section AssistNow online in the integration manual for details. ☞ Supplying time assistance that is inaccurate by more than the specified time accuracy, may lead to substantially degraded receiver performance.

Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x13	0x40	24	*see below*	CK_A CK_B

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	U1	`type`	-	-	Message type (0x11 for this type)
1	U1	`version`	-	-	Message version (0x00 for this version)
2	X1	`ref`	-	-	Reference to be used to set time
bits 3...0	U:4	`source`	-	-	• 0 = none, i.e. on receipt of message (will be inaccurate!) • 1 = relative to pulse sent to EXTINT0 • 2 = relative to pulse sent to EXTINT1 • 3-15 = reserved
bit 4	U:1	`fall`	-	-	use falling edge of EXTINT pulse (default rising) - only if source is EXTINT
bit 5	U:1	`last`	-	-	use last EXTINT pulse (default next pulse) - only if source is EXTINT
3	U1	`gnssId`	-	-	Source of time information. Currently supported: • 0 = GPS time • 2 = Galileo time • 3 = BeiDou time • 6 = GLONASS time • 7 = NavIC time
4	U1[2]	`reserved0`	-	-	Reserved
6	U2	`week`	-	-	GNSS week number
8	U4	`tow`	-	s	GNSS time of week
12	U4	`ns`	-	ns	GNSS time of week, nanosecond part from 0 to 999,999,999
16	U2	`tAccS`	-	s	Seconds part of time accuracy
18	U1[2]	`reserved1`	-	-	Reserved
20	U4	`tAccNs`	-	ns	Nanoseconds part of time accuracy, from 0 to 999,999,999

3.13.7.5 Initial clock drift assistance

Message	**UBX-MGA-INI-CLKD**
	Initial clock drift assistance
Type	Input
Comment	This message allows the delivery of clock drift assistance to a receiver. See section AssistNow online in the integration manual for details. ☞ Supplying clock drift assistance that is inaccurate by more than the specified accuracy, may lead to substantially degraded receiver performance.

Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x13	0x40	12	*see below*	CK_A CK_B

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	U1	`type`	-	-	Message type (0x20 for this type)
1	U1	`version`	-	-	Message version (0x00 for this version)
2	U1[2]	`reserved0`	-	-	Reserved
4	I4	`clkD`	-	ns/s	Clock drift
8	U4	`clkDAcc`	-	ns/s	Clock drift accuracy

3.13.7.6 Initial frequency assistance

Message	**UBX-MGA-INI-FREQ**
	Initial frequency assistance
Type	Input
Comment	This message allows the delivery of external frequency assistance to a receiver. See section AssistNow online in the integration manual for details. ☞ Supplying external frequency assistance that is inaccurate by more than the specified accuracy, may lead to substantially degraded receiver performance.

Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x13	0x40	12	*see below*	CK_A CK_B

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	U1	`type`	-	-	Message type (0x21 for this type)
1	U1	`version`	-	-	Message version (0x00 for this version)
2	U1	`reserved0`	-	-	Reserved
3	X1	`flags`	-	-	Frequency reference
bits 3…0	U:4	`source`	-	-	• 0 = frequency available on EXTINT0 • 1 = frequency available on EXTINT1 • 2-15 = reserved
bit 4	U:1	`fall`	-	-	use falling edge of EXTINT pulse (default rising)
4	I4	`freq`	1e-2	Hz	Frequency
8	U4	`freqAcc`	-	ppb	Frequency accuracy

3.13.8 UBX-MGA-QZSS (0x13 0x05)

3.13.8.1 QZSS ephemeris assistance

Message	**UBX-MGA-QZSS-EPH** **QZSS ephemeris assistance**				
Type	Input				
Comment	This message allows the delivery of QZSS ephemeris assistance to a receiver. See section AssistNow Online in the integration manual for details.				
Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x13	0x05	68	*see below*	CK_A CK_B

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	U1	`type`	-	-	Message type (0x01 for this type)
1	U1	`version`	-	-	Message version (0x00 for this version)
2	U1	`svId`	-	-	QZSS Satellite identifier (see Satellite Numbering), Range 1-5
3	U1	`reserved0`	-	-	Reserved
4	U1	`fitInterval`	-	-	Fit interval flag
5	U1	`uraIndex`	-	-	URA index
6	U1	`svHealth`	-	-	SV health
7	I1	`tgd`	2^-31	s	Group delay differential
8	U2	`iodc`	-	-	IODC
10	U2	`toc`	2^4	s	Clock data reference time
12	U1	`reserved1`	-	-	Reserved
13	I1	`af2`	2^-55	s/s squared	Time polynomial coefficient 2
14	I2	`af1`	2^-43	s/s	Time polynomial coefficient 1
16	I4	`af0`	2^-31	s	Time polynomial coefficient 0
20	I2	`crs`	2^-5	m	Crs
22	I2	`deltaN`	2^-43	semi-circles/s	Mean motion difference from computed value
24	I4	`m0`	2^-31	semi-circles	Mean anomaly at reference time
28	I2	`cuc`	2^-29	radians	Amp of cosine harmonic corr term to arg of lat
30	I2	`cus`	2^-29	radians	Amp of sine harmonic corr term to arg of lat
32	U4	`e`	2^-33	-	eccentricity
36	U4	`sqrtA`	2^-19	m^0.5	Square root of the semi-major axis A
40	U2	`toe`	2^4	s	Reference time of ephemeris
42	I2	`cic`	2^-29	radians	Amp of cos harmonic corr term to angle of inclination
44	I4	`omega0`	2^-31	semi-circles	Long of asc node of orbit plane at weekly epoch
48	I2	`cis`	2^-29	radians	Amp of sine harmonic corr term to angle of inclination
50	I2	`crc`	2^-5	m	Amp of cosine harmonic corr term to orbit radius
52	I4	`i0`	2^-31	semi-circles	Inclination angle at reference time
56	I4	`omega`	2^-31	semi-circles	Argument of perigee

60	I4	omegaDot	2^-43	semi-circles/s	Rate of right ascension
64	I2	idot	2^-43	semi-circles/s	Rate of inclination angle
66	U1[2]	reserved2	-	-	Reserved

3.13.8.2 QZSS almanac assistance

Message	**UBX-MGA-QZSS-ALM** **QZSS almanac assistance**				
Type	Input				
Comment	This message allows the delivery of QZSS almanac assistance to a receiver. See section AssistNow Online in the integration manual for details.				
Message structure	*Header* 0xb5 0x62	*Class* 0x13	*ID* 0x05	*Length (Bytes)* 36	*Payload* *see below*	*Checksum* CK_A CK_B

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	U1	type	-	-	Message type (0x02 for this type)
1	U1	version	-	-	Message version (0x00 for this version)
2	U1	svId	-	-	QZSS Satellite identifier (see Satellite Numbering), Range 1-5
3	U1	svHealth	-	-	Almanac SV health information
4	U2	e	2^-21	-	Almanac eccentricity
6	U1	almWNa	-	week	Reference week number of almanac (the 8-bit WNa field)
7	U1	toa	2^12	s	Reference time of almanac
8	I2	deltaI	2^-19	semi-circles	Delta inclination angle at reference time
10	I2	omegaDot	2^-38	semi-circles/s	Almanac rate of right ascension
12	U4	sqrtA	2^-11	m^0.5	Almanac square root of the semi-major axis A
16	I4	omega0	2^-23	semi-circles	Almanac long of asc node of orbit plane at weekly
20	I4	omega	2^-23	semi-circles	Almanac argument of perigee
24	I4	m0	2^-23	semi-circles	Almanac mean anomaly at reference time
28	I2	af0	2^-20	s	Almanac time polynomial coefficient 0 (8 MSBs)
30	I2	af1	2^-38	s/s	Almanac time polynomial coefficient 1
32	U1[4]	reserved0	-	-	Reserved

3.13.8.3 QZSS health assistance

Message	**UBX-MGA-QZSS-HEALTH** **QZSS health assistance**				
Type	Input				
Comment	This message allows the delivery of QZSS health assistance to a receiver. See section AssistNow Online in the integration manual for details.				
Message structure	*Header* 0xb5 0x62	*Class* 0x13	*ID* 0x05	*Length (Bytes)* 12	*Payload* *see below*	*Checksum* CK_A CK_B

Payload description:					
Byte offset	Type	Name	Scale	Unit	Description
0	U1	type	-	-	Message type (0x04 for this type)
1	U1	version	-	-	Message version (0x00 for this version)
2	U1[2]	reserved0	-	-	Reserved
4	U1[5]	healthCode	-	-	Each byte represents a QZSS SV (1-5). The 6 LSBs of each byte contains the 6 bit health code from subframes 4/5, data ID = 3, SV ID = 51
9	U1[3]	reserved1	-	-	Reserved

3.14 UBX-MON (0x0a)

The messages in the UBX-MON class are used to report the receiver status, such as hardware status or I/O subsystem statistics.

3.14.1 UBX-MON-COMMS (0x0a 0x36)

3.14.1.1 Communication port information

Message	**UBX-MON-COMMS** **Communication port information**				
Type	Periodic/polled				
Comment	Consolidated communications information for all ports. The size of the message is determined by the number of ports that are in use on the receiver. A port is only included if communication, either send or receive, has been initiated on that port.				

Message structure	Header	Class	ID	Length (Bytes)	Payload	Checksum
	0xb5 0x62	0x0a	0x36	8 + nPorts·40	see below	CK_A CK_B

Payload description:					
Byte offset	Type	Name	Scale	Unit	Description
0	U1	version	-	-	Message version (0x00 for this version)
1	U1	nPorts	-	-	Number of ports included
2	X1	txErrors	-	-	TX error bitmask
bit 0	U:1	mem	-	-	Memory Allocation error
bit 1	U:1	alloc	-	-	Allocation error (TX buffer full)
3	U1	reserved0	-	-	Reserved
4	U1[4]	protIds	-		The identifiers of the protocols reported in the msgs array. 0: UBX, 1: NMEA, 2: RTCM2, 5: RTCM3, 6: SPARTN, 0xFF: No protocol reported.
Start of repeated group (nPorts times)					
8 + n·40	U2	portId	-	-	Unique identifier for the port. See section Communications ports in the integration manual for details.
10 + n·40	U2	txPending	-	bytes	Number of bytes pending in transmitter buffer
12 + n·40	U4	txBytes	-	bytes	Number of bytes ever sent
16 + n·40	U1	txUsage	-	%	Maximum usage transmitter buffer during the last sysmon period
17 + n·40	U1	txPeakUsage	-	%	Maximum usage transmitter buffer
18 + n·40	U2	rxPending	-	bytes	Number of bytes in receiver buffer

20 + n·40	U4	`rxBytes`	-	bytes	Number of bytes ever received
24 + n·40	U1	`rxUsage`	-	%	Maximum usage receiver buffer during the last sysmon period
25 + n·40	U1	`rxPeakUsage`	-	%	Maximum usage receiver buffer
26 + n·40	U2	`overrunErrs`	-	-	Number of 100 ms timeslots with overrun errors
28 + n·40	U2[4]	`msgs`	-	msg	Number of successfully parsed messages for each protocol. The reported protocols are identified through the protIds field.
36 + n·40	U1[8]	`reserved1`	-	-	Reserved
44 + n·40	U4	`skipped`	-	bytes	Number of skipped bytes
End of repeated group (`nPorts` times)					

3.14.2 UBX-MON-GNSS (0x0a 0x28)

3.14.2.1 Information message major GNSS selection

Message	**UBX-MON-GNSS** **Information message major GNSS selection**					
Type	Polled					
Comment	This message reports major GNSS selection. It does this by means of bit masks in U1 fields. Each bit in a bit mask corresponds to one major GNSS. Augmentation systems are not reported.					
Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x0a	0x28	8	*see below*	CK_A CK_B

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	U1	`version`	-	-	Message version (0x00 for this version)
1	X1	`supported`	-	-	A bit mask showing the major GNSS that can be supported by this receiver
bit 0	U:1	`GPSSup`	-	-	GPS is supported
bit 1	U:1	`GlonassSup`	-	-	GLONASS is supported
bit 2	U:1	`BeidouSup`	-	-	BeiDou is supported
bit 3	U:1	`GalileoSup`	-	-	Galileo is supported
2	X1	`defaultGnss`	-	-	A bit mask showing the default major GNSS selection. If the default major GNSS selection is currently configured in the efuse for this receiver, it takes precedence over the default major GNSS selection configured in the executing firmware of this receiver.
bit 0	U:1	`GPSDef`	-	-	GPS is default-enabled
bit 1	U:1	`GlonassDef`	-	-	GLONASS is default-enabled
bit 2	U:1	`BeidouDef`	-	-	BeiDou is default-enabled
bit 3	U:1	`GalileoDef`	-	-	Galileo is default-enabled
3	X1	`enabled`	-	-	A bit mask showing the current major GNSS selection enabled for this receiver
bit 0	U:1	`GPSEna`	-	-	GPS is enabled
bit 1	U:1	`GlonassEna`	-	-	GLONASS is enabled
bit 2	U:1	`BeidouEna`	-	-	BeiDou is enabled
bit 3	U:1	`GalileoEna`	-	-	Galileo is enabled

4	U1	`simultaneous`	-	-	Maximum number of concurrent major GNSS that can be supported by this receiver
5	U1[3]	`reserved0`	-	-	Reserved

3.14.3 UBX-MON-HW (0x0a 0x09)

3.14.3.1 Hardware status

Message	**UBX-MON-HW** **Hardware status**
Type	Periodic/polled
Comment	**This message is deprecated in this protocol version. Use UBX-MON-HW3 and UBX-MON-RF instead.** Status of different aspects of the hardware, such as antenna, PIO/peripheral pins, noise level, automatic gain control (AGC)

Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x0a	0x09	60	*see below*	CK_A CK_B

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	X4	`pinSel`	-	-	Mask of pins set as peripheral/PIO
4	X4	`pinBank`	-	-	Mask of pins set as bank A/B
8	X4	`pinDir`	-	-	Mask of pins set as input/output
12	X4	`pinVal`	-	-	Mask of pins value low/high
16	U2	`noisePerMS`	-	-	Noise level as measured by the GPS core
18	U2	`agcCnt`	-	-	AGC monitor (counts SIGHI xor SIGLO, range 0 to 8191)
20	U1	`aStatus`	-	-	Status of the antenna supervisor state machine (0=INIT, 1=DONTKNOW, 2=OK, 3=SHORT, 4=OPEN)
21	U1	`aPower`	-	-	Current power status of antenna (0=OFF, 1=ON, 2=DONTKNOW)
22	X1	`flags`	-	-	Flags
bit 0	U:1	`rtcCalib`	-	-	RTC is calibrated
bit 1	U:1	`safeBoot`	-	-	Safeboot mode (0 = inactive, 1 = active)
bits 3…2	U:2	`jammingState`	-	-	Output from jamming/interference monitor (0 = unknown or feature disabled, 1 = ok - no significant jamming, 2 = warning - interference visible but fix OK, 3 = critical - interference visible and no fix)
bit 4	U:1	`xtalAbsent`	-	-	RTC xtal has been determined to be absent (not supported for protocol versions less than 18.00)
23	U1	`reserved0`	-	-	Reserved
24	X4	`usedMask`	-	-	Mask of pins that are used by the virtual pin manager
28	U1[17]	`VP`	-	-	Array of pin mappings for each of the 17 physical pins
45	U1	`jamInd`	-	-	CW jamming indicator, scaled (0 = no CW jamming, 255 = strong CW jamming)
46	U1[2]	`reserved1`	-	-	Reserved
48	X4	`pinIrq`	-	-	Mask of pins value using the PIO Irq
52	X4	`pullH`	-	-	Mask of pins value using the PIO pull high resistor

56	X4	`pullL`	-	-	Mask of pins value using the PIO pull low resistor

3.14.4 UBX-MON-HW2 (0x0a 0x0b)

3.14.4.1 Extended hardware status

Message	**UBX-MON-HW2** **Extended hardware status**
Type	Periodic/polled
Comment	**This message is deprecated in this protocol version. Use UBX-MON-HW3 and UBX-MON-RF instead.** Status of different aspects of the hardware such as Imbalance, Low-Level Configuration and POST Results. The first four parameters of this message represent the complex signal from the RF front end. The following rules of thumb apply: • The smaller the absolute value of the variable `ofsI` and `ofsQ`, the better. • Ideally, the magnitude of the I-part (`magI`) and the Q-part (`magQ`) of the complex signal should be the same.

Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x0a	0x0b	28	*see below*	CK_A CK_B

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	I1	`ofsI`	-	-	Imbalance of I-part of complex signal, scaled (-128 = max. negative imbalance, 127 = max. positive imbalance)
1	U1	`magI`	-	-	Magnitude of I-part of complex signal, scaled (0 = no signal, 255 = max. magnitude)
2	I1	`ofsQ`	-	-	Imbalance of Q-part of complex signal, scaled (-128 = max. negative imbalance, 127 = max. positive imbalance)
3	U1	`magQ`	-	-	Magnitude of Q-part of complex signal, scaled (0 = no signal, 255 = max. magnitude)
4	U1	`cfgSource`	-	-	Source of low-level configuration (114 = ROM, 111 = OTP, 112 = config pins, 102 = flash image)
5	U1[3]	`reserved0`	-	-	Reserved
8	U4	`lowLevCfg`	-	-	Low-level configuration (obsolete for protocol versions greater than 15.00)
12	U1[8]	`reserved1`	-	-	Reserved
20	U4	`postStatus`	-	-	POST status word
24	U1[4]	`reserved2`	-	-	Reserved

3.14.5 UBX-MON-HW3 (0x0a 0x37)

3.14.5.1 I/O pin status

Message	**UBX-MON-HW3** **I/O pin status**
Type	Periodic/polled
Comment	This message contains information specific to each HW I/O pin, for example whether the pin is set as Input or Output. For the antenna supervisor status and other RF status information, see the UBX-MON-RF message.

Message structure	Header	Class	ID	Length (Bytes)	Payload	Checksum
	0xb5 0x62	0x0a	0x37	22 + nPins·6	see below	CK_A CK_B

Payload description:

Byte offset	Type	Name	Scale	Unit	Description
0	U1	version	-	-	Message version (0x00 for this version)
1	U1	nPins	-	-	The number of I/O pins included
2	X1	flags	-	-	Flags
bit 0	U:1	rtcCalib	-	-	RTC is calibrated
bit 1	U:1	safeBoot	-	-	Safeboot mode (0 = inactive, 1 = active)
bit 2	U:1	xtalAbsent	-	-	RTC xtal has been determined to be absent
3	CH[10]	hwVersion	-	-	Zero-terminated hardware version string (same as that returned in the UBX-MON-VER message)
13	U1[9]	reserved0	-	-	Reserved
Start of repeated group (nPins times)					
22 + n·6	U2	pinId	-	-	Identifier for the pin, including both external and internal pins.
24 + n·6	X2	pinMask	-	-	Pin mask
bit 0	U:1	periphPIO	-	-	Pin is set to peripheral or PIO? 0=Peripheral 1=PIO
bits 3…1	U:3	pinBank	-	-	Bank the pin belongs to, where 0=A 1=B 2=C 3=D 4=E 5=F 6=G 7=H
bit 4	U:1	direction	-	-	Pin direction? 0=Input 1=Output
bit 5	U:1	value	-	-	Pin value? 0=Low 1=High
bit 6	U:1	vpManager	-	-	Used by virtual pin manager? 0=No 1=Yes
bit 7	U:1	pioIrq	-	-	Interrupt enabled? 0=No 1=Yes
bit 8	U:1	pioPullHigh	-	-	Using pull high resistor? 0=No 1=Yes
bit 9	U:1	pioPullLow	-	-	Using pull low resistor 0=No 1=Yes
26 + n·6	U1	VP	-	-	Virtual pin mapping
27 + n·6	U1	reserved1	-	-	Reserved
End of repeated group (nPins times)					

3.14.6 UBX-MON-IO (0x0a 0x02)

3.14.6.1 I/O system status

Message	**UBX-MON-IO** **I/O system status**
Type	Periodic/polled
Comment	**This message is deprecated in this protocol version. Use UBX-MON-COMMS instead.** The size of the message is determined by the number of ports 'N' the receiver supports, i.e. on u-blox 5 the number of ports is 6.

Message structure	Header	Class	ID	Length (Bytes)	Payload	Checksum
	0xb5 0x62	0x0a	0x02	[0..n]·20	see below	CK_A CK_B

Payload description:

Byte offset	Type	Name	Scale	Unit	Description
Start of repeated group (N times)					
0 + n·20	U4	rxBytes	-	bytes	Number of bytes ever received

4 + n·20	U4	`txBytes`	-	bytes	Number of bytes ever sent
8 + n·20	U2	`parityErrs`	-	-	Number of 100 ms timeslots with parity errors
10 + n·20	U2	`framingErrs`	-	-	Number of 100 ms timeslots with framing errors
12 + n·20	U2	`overrunErrs`	-	-	Number of 100 ms timeslots with overrun errors
14 + n·20	U2	`breakCond`	-	-	Number of 100 ms timeslots with break conditions
16 + n·20	U1[4]	`reserved0`	-	-	Reserved
End of repeated group (N times)					

3.14.7 UBX-MON-MSGPP (0x0a 0x06)

3.14.7.1 Message parse and process status

Message	**UBX-MON-MSGPP** **Message parse and process status**				
Type	Periodic/polled				
Comment	**This message is deprecated in this protocol version. Use UBX-MON-COMMS instead.**				
Message structure	*Header* 0xb5 0x62	*Class* 0x0a	*ID* 0x06	*Length (Bytes)* 120	*Payload* *see below*	*Checksum* CK_A CK_B

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	U2[8]	`msg1`	-	msgs	Number of successfully parsed messages for each protocol on port0
16	U2[8]	`msg2`	-	msgs	Number of successfully parsed messages for each protocol on port1
32	U2[8]	`msg3`	-	msgs	Number of successfully parsed messages for each protocol on port2
48	U2[8]	`msg4`	-	msgs	Number of successfully parsed messages for each protocol on port3
64	U2[8]	`msg5`	-	msgs	Number of successfully parsed messages for each protocol on port4
80	U2[8]	`msg6`	-	msgs	Number of successfully parsed messages for each protocol on port5
96	U4[6]	`skipped`	-	bytes	Number skipped bytes for each port

3.14.8 UBX-MON-PATCH (0x0a 0x27)

3.14.8.1 Installed patches

Message	**UBX-MON-PATCH** **Installed patches**				
Type	Polled				
Comment	This message reports information about patches installed and currently enabled on the receiver. It does not report on patches installed and then disabled. An enabled patch is considered active when the receiver executes from the code space where the patch resides on. For example, a ROM patch is reported active only when the system runs from ROM.				
Message structure	*Header* 0xb5 0x62	*Class* 0x0a	*ID* 0x27	*Length (Bytes)* 4 + nEntries·16	*Payload* *see below*	*Checksum* CK_A CK_B

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*

0	U2	version	-	-	Message version (0x0001 for this version)
2	U2	nEntries	-	-	Total number of reported patches
Start of repeated group (nEntries times)					
4 + n·16	X4	patchInfo	-	-	Status information about the reported patch
bit 0	U:1	activated	-	-	1: the patch is active, 0: otherwise
bits 2...1	U:2	location	-	-	Indicates where the patch is stored. 0: eFuse, 1: ROM, 2: BBR, 3: file system
8 + n·16	U4	comparator Number	-	-	The number of the comparator
12 + n·16	U4	patchAddress	-	-	The address that is targeted by the patch
16 + n·16	U4	patchData	-	-	The data that is inserted at the patchAddress
End of repeated group (nEntries times)					

3.14.9 UBX-MON-RF (0x0a 0x38)

3.14.9.1 RF information

Message	**UBX-MON-RF** **RF information**				
Type	Periodic/polled				
Comment	Information for each RF block. There are as many RF blocks reported as bands supported by this receiver.				
Message structure	*Header* 0xb5 0x62	*Class* 0x0a	*ID* 0x38	*Length (Bytes)* 4 + nBlocks·24	*Payload* *see below*	*Checksum* CK_A CK_B

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	U1	version	-	-	Message version (0x00 for this version)
1	U1	nBlocks	-	-	The number of RF blocks included
2	U1[2]	reserved0	-	-	Reserved
Start of repeated group (nBlocks times)					
4 + n·24	U1	blockId	-	-	RF block ID (0 = L1 band, 1 = L2 or L5 band depending on product configuration)
5 + n·24	X1	flags	-	-	Flags
bits 1...0	U:2	jammingState	-	-	output from Jamming/Interference Monitor (0 = unknown or feature disabled, 1 = ok - no significant jamming, 2 = warning - interference visible but fix OK, 3 = critical - interference visible and no fix)
6 + n·24	U1	antStatus	-	-	Status of the antenna supervisor state machine (0x00=INIT, 0x01=DONTKNOW, 0x02=OK, 0x03=SHORT, 0x04=OPEN)
7 + n·24	U1	antPower	-	-	Current power status of antenna (0x00=OFF, 0x01=ON, 0x02=DONTKNOW)
8 + n·24	U4	postStatus	-	-	POST status word
12 + n·24	U1[4]	reserved1	-	-	Reserved
16 + n·24	U2	noisePerMS	-	-	Noise level as measured by the GPS core
18 + n·24	U2	agcCnt	-	-	AGC Monitor (counts SIGHI xor SIGLO, range 0 to 8191)

20 + n·24	U1	`jamInd`	-	-	CW jamming indicator, scaled (0=no CW jamming, 255 = strong CW jamming)
21 + n·24	I1	`ofsI`	-	-	Imbalance of I-part of complex signal, scaled (-128 = max. negative imbalance, 127 = max. positive imbalance)
22 + n·24	U1	`magI`	-	-	Magnitude of I-part of complex signal, scaled (0 = no signal, 255 = max.magnitude)
23 + n·24	I1	`ofsQ`	-	-	Imbalance of Q-part of complex signal, scaled (-128 = max. negative imbalance, 127 = max. positive imbalance)
24 + n·24	U1	`magQ`	-	-	Magnitude of Q-part of complex signal, scaled (0 = no signal, 255 = max.magnitude)
25 + n·24	U1[3]	`reserved2`	-	-	Reserved
End of repeated group (`nBlocks` times)					

3.14.10 UBX-MON-RXBUF (0x0a 0x07)

3.14.10.1 Receiver buffer status

Message	**UBX-MON-RXBUF** **Receiver buffer status**				
Type	Periodic/polled				
Comment	**This message is deprecated in this protocol version. Use UBX-MON-COMMS instead.**				
Message structure	*Header* 0xb5 0x62	*Class* 0x0a	*ID* 0x07	*Length (Bytes)* 24	*Payload* see below	*Checksum* CK_A CK_B

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	U2[6]	`pending`	-	bytes	Number of bytes pending in receiver buffer for each target
12	U1[6]	`usage`	-	%	Maximum usage receiver buffer during the last sysmon period for each target
18	U1[6]	`peakUsage`	-	%	Maximum usage receiver buffer for each target

3.14.11 UBX-MON-RXR (0x0a 0x21)

3.14.11.1 Receiver status information

Message	**UBX-MON-RXR** **Receiver status information**					
Type	Output					
Comment	The receiver ready message is sent when the receiver changes from or to backup mode.					
Message structure	*Header* 0xb5 0x62	*Class* 0x0a	*ID* 0x21	*Length (Bytes)* 1	*Payload* see below	*Checksum* CK_A CK_B

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	X1	`flags`	-	-	Receiver status flags
bit 0	U:1	`awake`	-	-	not in backup mode

3.14.12 UBX-MON-SPAN (0x0a 0x31)

3.14.12.1 Signal characteristics

Message	**UBX-MON-SPAN** **Signal characteristics**
Type	Periodic/polled
Comment	This message is to be used as a basic spectrum analyzer, where it displays one spectrum for each of the receiver's existing RF paths. The spectrum is conveyed with the following parameters: The frequency span in Hz, the frequency bin resolution in Hz, the center frequency in Hz, and 256 bins with amplitude data. Additionally, in order to give further insight on the signal captured by the receiver, the current gain of the internal programmable gain amplifier (PGA) is provided. **This message gives information for comparative analysis rather than absolute and precise spectrum overview. Users should not expect highly accurate spectrum amplitude.** Note that the PGA gain is not included in the spectrum data but is available as a separate field. Neither the spectrum, nor the PGA gain considers the internal fixed LNA gain or an external third-party LNA. The center frequency at each bin, assuming a zero-based bin count, can be computed as *f(i) = center + span * (i - 127) / 256*

Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x0a	0x31	4 + numRfBlocks·272	*see below*	CK_A CK_B

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	U1	`version`	-	-	Message version (0x00 for this version)
1	U1	`numRfBlocks`	-	-	Number of RF blocks included
2	U1[2]	`reserved0`	-	-	Reserved
Start of repeated group (`numRfBlocks` times)					
4 + n·272	U1[256]	`spectrum`	-	dB	Spectrum data (number of points = span/res)
260 + n·272	U4	`span`	-	Hz	Spectrum span
264 + n·272	U4	`res`	-	Hz	Resolution of the spectrum
268 + n·272	U4	`center`	-	Hz	Center of spectrum span
272 + n·272	U1	`pga`	-	dB	Programmable gain amplifier
273 + n·272	U1[3]	`reserved1`	-	-	Reserved
End of repeated group (`numRfBlocks` times)					

3.14.13 UBX-MON-SYS (0x0a 0x39)

3.14.13.1 Current system performance information

Message	**UBX-MON-SYS** **Current system performance information**
Type	Periodic/polled
Comment	This message contains operationally relevant system information for monitoring purposes. cpuLoadMax value is only valid, if 1 second output frequency is set. Detailed information about ioUsage/ioUsageMax are available in UBX-MON-COMMS message. tempValue has an accuracy of +/- 2 deg.

Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x0a	0x39	24	*see below*	CK_A CK_B

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	U1	`msgVer`	-	-	Message Version (0x01)

1	U1	bootType	-	-	Boot type of master chip 0-Unknown 1-Cold Start 2-Watchdog 3-Hardware reset 4-Hardware backup 5-Software backup 6-Software reset 7-VIO fail 8-VDD_X fail 9-VDD_RF fail 10-V_CORE_HIGH fail
2	U1	cpuLoad	-	-	Highest actual load of realtime tasks of all CPUs in %
3	U1	cpuLoadMax	-	-	Maximal CPU load value in % seen since last restart
4	U1	memUsage	-	-	Highest actual dynamic memory usage of all CPUs in %
5	U1	memUsageMax	-	-	Maximal dynamic memory usage in % seen since last restart
6	U1	ioUsage	-	-	Highest actual IO bandwidth usage of all rx/tx interfaces in %
7	U1	ioUsageMax	-	-	Maximal bandwidth usage of all rx/tx interfaces in % seen since last restart
8	U4	runTime	-	sec	Time since last restart
12	U2	noticeCount	-	-	Number of notices occured since last restart
14	U2	warnCount	-	-	Number of warnings occured since last restart
16	U2	errorCount	-	-	Number of errors occured since last restart
18	I1	tempValue	-	-	Temperature value [C]
19	U1[5]	reserved0	-	-	Reserved

3.14.14 UBX-MON-TXBUF (0x0a 0x08)

3.14.14.1 Transmitter buffer status

Message	**UBX-MON-TXBUF** **Transmitter buffer status**				
Type	Periodic/polled				
Comment	**This message is deprecated in this protocol version. Use UBX-MON-COMMS instead.**				
Message structure	*Header* 0xb5 0x62	*Class* 0x0a	*ID* 0x08	*Length (Bytes)* 28	*Payload* *see below*	*Checksum* CK_A CK_B

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	U2[6]	pending	-	bytes	Number of bytes pending in transmitter buffer for each target
12	U1[6]	usage	-	%	Maximum usage transmitter buffer during the last sysmon period for each target
18	U1[6]	peakUsage	-	%	Maximum usage transmitter buffer for each target
24	U1	tUsage	-	%	Maximum usage of transmitter buffer during the last sysmon period for all targets

25	U1	tPeakusage	-	%	Maximum usage of transmitter buffer for all targets
26	X1	errors	-	-	Error bitmask
bits 5…0	U:6	limit	-	-	Buffer limit of corresponding target reached
bit 6	U:1	mem	-	-	Memory Allocation error
bit 7	U:1	alloc	-	-	Allocation error (TX buffer full)
27	U1	reserved0	-	-	Reserved

3.14.15 UBX-MON-VER (0x0a 0x04)

3.14.15.1 Receiver and software version

Message	**UBX-MON-VER** **Receiver and software version**					
Type	Polled					
Comment						
Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x0a	0x04	40 + [0..n]·30	*see below*	CK_A CK_B

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	CH[30]	swVersion	-	-	Nul-terminated software version string.
30	CH[10]	hwVersion	-	-	Nul-terminated hardware version string
Start of repeated group (N times)					
40 + n·30	CH[30]	extension	-	-	Extended software information strings. A series of nul-terminated strings. Each extension field is 30 characters long and contains varying software information. Not all extension fields may appear. Examples of reported information: the software version string of the underlying ROM (when the receiver's firmware is running from flash), the firmware version, the supported protocol version, the module identifier, the flash information structure (FIS) file information, the supported major GNSS, the supported augmentation systems. See Firmware and protocol versions for details.
End of repeated group (N times)					

3.15 UBX-NAV (0x01)

The messages in the UBX-NAV class are used to output navigation results and data, such as position, altitude and velocity in a number of formats, and status flags and accuracy estimate figures, or satellite and signal information. The messages are generated with the configured navigation rate.

3.15.1 UBX-NAV-CLOCK (0x01 0x22)

3.15.1.1 Clock solution

Message	**UBX-NAV-CLOCK** **Clock solution**
Type	Periodic/polled

Comment						
Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x01	0x22	20	*see below*	CK_A CK_B

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	U4	`iTOW`	-	ms	GPS time of week of the navigation epoch. See section Navigation epochs in the integration manual for details. See section iTOW timestamps in the integration manual for details.
4	I4	`clkB`	-	ns	Clock bias
8	I4	`clkD`	-	ns/s	Clock drift
12	U4	`tAcc`	-	ns	Time accuracy estimate
16	U4	`fAcc`	-	ps/s	Frequency accuracy estimate

3.15.2 UBX-NAV-COV (0x01 0x36)

3.15.2.1 Covariance matrices

Message	**UBX-NAV-COV** **Covariance matrices**					
Type	Periodic/polled					
Comment	This message outputs the covariance matrices for the position and velocity solutions in the topocentric coordinate system defined as the local-level North (N), East (E), Down (D) frame. As the covariance matrices are symmetric, only the upper triangular part is output.					
Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x01	0x36	64	*see below*	CK_A CK_B

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	U4	`iTOW`	-	ms	GPS time of week of the navigation epoch. See section iTOW timestamps in the integration manual for details.
4	U1	`version`	-	-	Message version (0x00 for this version)
5	U1	`posCovValid`	-	-	Position covariance matrix validity flag
6	U1	`velCovValid`	-	-	Velocity covariance matrix validity flag
7	U1[9]	`reserved0`	-	-	Reserved
16	R4	`posCovNN`	-	m^2	Position covariance matrix value p_NN
20	R4	`posCovNE`	-	m^2	Position covariance matrix value p_NE
24	R4	`posCovND`	-	m^2	Position covariance matrix value p_ND
28	R4	`posCovEE`	-	m^2	Position covariance matrix value p_EE
32	R4	`posCovED`	-	m^2	Position covariance matrix value p_ED
36	R4	`posCovDD`	-	m^2	Position covariance matrix value p_DD
40	R4	`velCovNN`	-	m^2/s^2	Velocity covariance matrix value v_NN
44	R4	`velCovNE`	-	m^2/s^2	Velocity covariance matrix value v_NE
48	R4	`velCovND`	-	m^2/s^2	Velocity covariance matrix value v_ND
52	R4	`velCovEE`	-	m^2/s^2	Velocity covariance matrix value v_EE

56	R4	velCovED	-	m^2/s^2	Velocity covariance matrix value v_ED
60	R4	velCovDD	-	m^2/s^2	Velocity covariance matrix value v_DD

3.15.3 UBX-NAV-DOP (0x01 0x04)

3.15.3.1 Dilution of precision

Message	**UBX-NAV-DOP** **Dilution of precision**				
Type	Periodic/polled				
Comment	• DOP values are dimensionless. • All DOP values are scaled by a factor of 100. If the unit transmits a value of e.g. 156, the DOP value is 1.56.				
Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x01	0x04	18	*see below*	CK_A CK_B

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	U4	iTOW	-	ms	GPS time of week of the navigation epoch. See section iTOW timestamps in the integration manual for details.
4	U2	gDOP	0.01	-	Geometric DOP
6	U2	pDOP	0.01	-	Position DOP
8	U2	tDOP	0.01	-	Time DOP
10	U2	vDOP	0.01	-	Vertical DOP
12	U2	hDOP	0.01	-	Horizontal DOP
14	U2	nDOP	0.01	-	Northing DOP
16	U2	eDOP	0.01	-	Easting DOP

3.15.4 UBX-NAV-EOE (0x01 0x61)

3.15.4.1 End of epoch

Message	**UBX-NAV-EOE** **End of epoch**				
Type	Periodic				
Comment	This message is intended to be used as a marker to collect all navigation messages of an epoch. It is output after all enabled NAV class messages (except UBX-NAV-HNR) and after all enabled NMEA messages.				
Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x01	0x61	4	*see below*	CK_A CK_B

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	U4	iTOW	-	ms	GPS time of week of the navigation epoch. See section iTOW timestamps in the integration manual for details.

3.15.5 UBX-NAV-GEOFENCE (0x01 0x39)

3.15.5.1 Geofencing status

Message	**UBX-NAV-GEOFENCE** **Geofencing status**					
Type	Periodic/polled					
Comment	This message outputs the evaluated states of all configured geofences for the current epoch's position. See section Geofencing in the integration manual for feature details.					
Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x01	0x39	8 + numFences·2	*see below*	CK_A CK_B

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	U4	`iTOW`	-	ms	GPS time of week of the navigation epoch. See section iTOW timestamps in the integration manual for details.
4	U1	`version`	-	-	Message version (0x00 for this version)
5	U1	`status`	-	-	Geofencing status • 0 - Geofencing not available or not reliable • 1 - Geofencing active
6	U1	`numFences`	-	-	Number of geofences
7	U1	`combState`	-	-	Combined (logical OR) state of all geofences • 0 - Unknown • 1 - Inside • 2 - Outside
Start of repeated group (`numFences` times)					
8 + n·2	U1	`state`	-	-	Geofence state • 0 - Unknown • 1 - Inside • 2 - Outside
9 + n·2	U1	`id`	-	-	Geofence ID (0 = not available)
End of repeated group (`numFences` times)					

3.15.6 UBX-NAV-HPPOSECEF (0x01 0x13)

3.15.6.1 High precision position solution in ECEF

Message	**UBX-NAV-HPPOSECEF** **High precision position solution in ECEF**					
Type	Periodic/polled					
Comment	See important comments concerning validity of position given in section Navigation output filters in the integration manual.					
Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x01	0x13	28	*see below*	CK_A CK_B

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	U1	`version`	-	-	Message version (0x00 for this version)
1	U1[3]	`reserved0`	-	-	Reserved
4	U4	`iTOW`	-	ms	GPS time of week of the navigation epoch. See section iTOW timestamps in the integration manual for details.

8	I4	`ecefX`	-	cm	ECEF X coordinate
12	I4	`ecefY`	-	cm	ECEF Y coordinate
16	I4	`ecefZ`	-	cm	ECEF Z coordinate
20	I1	`ecefXHp`	0.1	mm	High precision component of ECEF X coordinate. Must be in the range of -99..+99. Precise coordinate in cm = ecefX + (ecefXHp * 1e-2).
21	I1	`ecefYHp`	0.1	mm	High precision component of ECEF Y coordinate. Must be in the range of -99..+99. Precise coordinate in cm = ecefY + (ecefYHp * 1e-2).
22	I1	`ecefZHp`	0.1	mm	High precision component of ECEF Z coordinate. Must be in the range of -99..+99. Precise coordinate in cm = ecefZ + (ecefZHp * 1e-2).
23	X1	`flags`	-	-	Additional flags
bit 0	U:1	`invalidEcef`	-	-	1 = Invalid ecefX, ecefY, ecefZ, ecefXHp, ecefYHp and ecefZHp
24	U4	`pAcc`	0.1	mm	Position Accuracy Estimate

3.15.7 UBX-NAV-HPPOSLLH (0x01 0x14)

3.15.7.1 High precision geodetic position solution

Message	**UBX-NAV-HPPOSLLH**				
	High precision geodetic position solution				
Type	Periodic/polled				
Comment	See important comments concerning validity of position given in section Navigation output filters in the integration manual. This message outputs the Geodetic position in the currently selected ellipsoid. The default is the WGS84 Ellipsoid, but can be changed with the message CFG-NAVSPG-USE_USRDAT.				
Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x01	0x14	36	*see below*	CK_A CK_B

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	U1	`version`	-	-	Message version (0x00 for this version)
1	U1[2]	`reserved0`	-	-	Reserved
3	X1	`flags`	-	-	Additional flags
bit 0	U:1	`invalidLlh`	-	-	1 = Invalid lon, lat, height, hMSL, lonHp, latHp, heightHp and hMSLHp
4	U4	`iTOW`	-	ms	GPS time of week of the navigation epoch. See section iTOW timestamps in the integration manual for details.
8	I4	`lon`	1e-7	deg	Longitude
12	I4	`lat`	1e-7	deg	Latitude
16	I4	`height`	-	mm	Height above ellipsoid.
20	I4	`hMSL`	-	mm	Height above mean sea level
24	I1	`lonHp`	1e-9	deg	High precision component of longitude. Must be in the range -99..+99. Precise longitude in deg * 1e-7 = lon + (lonHp * 1e-2).

25	I1	`latHp`	1e-9	deg	High precision component of latitude. Must be in the range -99..+99. Precise latitude in deg * 1e-7 = lat + (latHp * 1e-2).
26	I1	`heightHp`	0.1	mm	High precision component of height above ellipsoid. Must be in the range -9..+9. Precise height in mm = height + (heightHp * 0.1).
27	I1	`hMSLHp`	0.1	mm	High precision component of height above mean sea level. Must be in range -9..+9. Precise height in mm = hMSL + (hMSLHp * 0.1)
28	U4	`hAcc`	0.1	mm	Horizontal accuracy estimate
32	U4	`vAcc`	0.1	mm	Vertical accuracy estimate

3.15.8 UBX-NAV-ODO (0x01 0x09)

3.15.8.1 Odometer solution

Message	**UBX-NAV-ODO** **Odometer solution**					
Type	Periodic/polled					
Comment	This message outputs the traveled distance since last reset (see UBX-NAV-RESETODO) together with an associated estimated accuracy and the total cumulated ground distance (can only be reset by a cold start of the receiver).					
Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x01	0x09	20	*see below*	CK_A CK_B

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	U1	`version`	-	-	Message version (0x00 for this version)
1	U1[3]	`reserved0`	-	-	Reserved
4	U4	`iTOW`	-	ms	GPS time of week of the navigation epoch. See section iTOW timestamps in the integration manual for details.
8	U4	`distance`	-	m	Ground distance since last reset
12	U4	`totalDistance`	-	m	Total cumulative ground distance
16	U4	`distanceStd`	-	m	Ground distance accuracy (1-sigma)

3.15.9 UBX-NAV-ORB (0x01 0x34)

3.15.9.1 GNSS orbit database info

Message	**UBX-NAV-ORB** **GNSS orbit database info**					
Type	Periodic/polled					
Comment	Status of the GNSS orbit database knowledge.					
Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x01	0x34	8 + numSv·6	*see below*	CK_A CK_B

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	U4	`iTOW`	-	ms	GPS time of week of the navigation epoch. See section iTOW timestamps in the integration manual for details.

4	U1	version	-	-	Message version (0x01 for this version)
5	U1	numSv	-	-	Number of SVs in the database
6	U1[2]	reserved0	-	-	Reserved
Start of repeated group (numSv times)					
8 + n·6	U1	gnssId	-	-	GNSS ID
9 + n·6	U1	svId	-	-	Satellite ID
10 + n·6	X1	svFlag	-	-	Information Flags
bits 1…0	U:2	health	-	-	SV health: • 0 = unknown • 1 = healthy • 2 = not healty
bits 3…2	U:2	visibility	-	-	SV health: • 0 = unknown • 1 = below horizon • 2 = above horizon • 3 = above elevation mask
11 + n·6	X1	eph	-	-	Ephemeris data In products supporting L5 signals, the receiver may store multiple ephemeris data sets per satellite. ephUsability and ephSource fields show information on one of the data sets. It is not possible to choose which data set's status is shown.
bits 4…0	U:5	ephUsability	-	-	How long the receiver will be able to use the stored ephemeris data from now on: • 31 = The usability period is unknown • 30 = The usability period is more than 450 minutes • 30 > n > 0 = The usability period is between (n-1)*15 and n*15 minutes • 0 = Ephemeris can no longer be used
bits 7…5	U:3	ephSource	-	-	• 0 = not available • 1 = GNSS transmission • 2 = external aiding • 3-7 = other
12 + n·6	X1	alm	-	-	Almanac data
bits 4…0	U:5	almUsability	-	-	How long the receiver will be able to use the stored almanac data from now on: • 31 = The usability period is unknown • 30 = The usability period is more than 30 days • 30 > n > 0 = The usability period is between n-1 and n days • 0 = Almanac can no longer be used
bits 7…5	U:3	almSource	-	-	• 0 = not available • 1 = GNSS transmission • 2 = external aiding • 3-7 = other
13 + n·6	X1	otherOrb	-	-	Other orbit data available
bits 4…0	U:5	anoAop Usability	-	-	How long the receiver will be able to use the orbit data from now on: • 31 = The usability period is unknown • 30 = The usability period is more than 30 days • 30 > n > 0 = The usability period is between n-1 and n days

					• 0 = Data can no longer be used
bits 7…5	U:3	`type`	-	-	Type of orbit data: • 0 = No orbit data available • 1 = AssistNow Offline data • 2 = AssistNow Autonomous data • 3-7 = Other orbit data
End of repeated group (`numSv` times)					

3.15.10 UBX-NAV-PL (0x01 0x62)

3.15.10.1 Protection level information

Message	**UBX-NAV-PL** **Protection level information**					
Type	Periodic					
Comment	This message provides protection level (PL) values per protection level state (e.g. position ECEF X/Y/Z) and w.r.t. the given target misleading information risk (TMIR) per coordinate axis. Target misleading information risk is expressed as X [%MI/epoch] (read: X% probability of having an MI per epoch). Misleading information (MI) occurs when the Protection Level value is smaller than the true position error.					
Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x01	0x62	52	*see below*	CK_A CK_B

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	U1	`msgVersion`	-	-	Message version (0x01 for this version)
1	U1	`tmirCoeff`	-	-	Target misleading information risk (TMIR) [%MI/epoch], coefficient integer number of base 10 scientific notation (see e.g. plPos field)
2	I1	`tmirExp`	-	-	Target misleading information risk (TMIR) [%MI/epoch], exponent integer number of base 10 scientific notation (see e.g. plPos field)
3	U1	`plPosValid`	-	-	Position protection level validity • 0 Invalid (Protection level should not be used) • 1 Protection level is valid
4	U1	`plPosFrame`	-	-	Position protection level frame: • 0 Invalid (not possible to calculate frame conversion) • 1 North-East-Down • 2 Longitudinal-Lateral-Vertical • 3 HorizSemiMajorAxis-HorizSemiMinorAxis-Vertical
5	U1	`plVelValid`	-	-	Velocity protection level validity • 0 Invalid (Protection level should not be used) • 1 Protection level is valid
6	U1	`plVelFrame`	-	-	Velocity protection level frame: • 0 Invalid (not possible to calculate frame conversion) • 1 North-East-Down • 2 Longitudinal-Lateral-Vertical • 3 HorizSemiMajorAxis-HorizSemiMinorAxis-Vertical

7	U1	`plTimeValid`	-	-	Time protection level validity • 0 Invalid (Protection level should not be used) • 1 Protection level is valid
8	U1[4]	`reserved0`	-	-	Reserved
12	U4	`iTow`	-	ms	GPS time of week
16	U4	`plPos1`	-	mm	First axis of position protection level value, given in coordinate frame of plPosFrame (see plPosFrame field for value order), w.r.t. the given target misleading information risk (TMIR) of [tmirCoeff * 10^(tmirExp)]
20	U4	`plPos2`	-	mm	Second axis of position protection level value, given in coordinate frame of plPosFrame (see plPosFrame field for value order), w.r.t. the given target misleading information risk (TMIR) of [tmirCoeff * 10^(tmirExp)]
24	U4	`plPos3`	-	mm	Third axis of position protection level value, given in coordinate frame of plPosFrame (see plPosFrame field for value order), w.r.t. the given target misleading information risk (TMIR) of [tmirCoeff * 10^(tmirExp)]
28	U4	`plVel1`	-	mm/s	First axis of velocity protection level value, given in coordinate frame of plVelFrame (see plVelFrame field for value order), w.r.t. the given target misleading information risk (TMIR) of [tmirCoeff * 10^(tmirExp)]
32	U4	`plVel2`	-	mm/s	Second axis of velocity protection level value, given in coordinate frame of plVelFrame (see plVelFrame field for value order), w.r.t. the given target misleading information risk (TMIR) of [tmirCoeff * 10^(tmirExp)]
36	U4	`plVel3`	-	mm/s	Third axis of velocity protection level value, given in coordinate frame of plVelFrame (see plVelFrame field for value order), w.r.t. the given target misleading information risk (TMIR) of [tmirCoeff * 10^(tmirExp)]
40	U2	`plPosHorizOrient`	1e-2	deg	Orientation of HorizSemiMajorAxis (see plPosFrame) of horizontal ellipse position protection level (clockwise degrees from true North), if plPosFrame==3; zero otherwise.
42	U2	`plVelHorizOrient`	1e-2	deg	Orientation of HorizSemiMajorAxis (see plVelFrame) of horizontal ellipse velocity protection level (clockwise degrees from true North), if plVelFrame==3; zero otherwise.
44	U4	`plTime`	-	ns	Time protection level value, w.r.t. the given target misleading information risk (TMIR) of [tmirCoeff * 10^(tmirExp)]
48	U1[4]	`reserved1`	-	-	Reserved

3.15.11 UBX-NAV-POSECEF (0x01 0x01)

3.15.11.1 Position solution in ECEF

Message	**UBX-NAV-POSECEF** **Position solution in ECEF**					
Type	Periodic/polled					
Comment	See important comments concerning validity of position given in section Navigation output filters in the integration manual.					
Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x01	0x01	20	*see below*	CK_A CK_B

Payload description:

Byte offset	Type	Name	Scale	Unit	Description
0	U4	`iTOW`	-	ms	GPS time of week of the navigation epoch. See section iTOW timestamps in the integration manual for details.
4	I4	`ecefX`	-	cm	ECEF X coordinate
8	I4	`ecefY`	-	cm	ECEF Y coordinate
12	I4	`ecefZ`	-	cm	ECEF Z coordinate
16	U4	`pAcc`	-	cm	Position Accuracy Estimate

3.15.12 UBX-NAV-POSLLH (0x01 0x02)

3.15.12.1 Geodetic position solution

Message	**UBX-NAV-POSLLH** **Geodetic position solution**
Type	Periodic/polled
Comment	See important comments concerning validity of position given in section Navigation output filters in the integration manual. This message outputs the Geodetic position in the currently selected ellipsoid. The default is the WGS84 Ellipsoid, but can be changed with the message CFG-NAVSPG-USE_USRDAT.

Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x01	0x02	28	*see below*	CK_A CK_B

Payload description:

Byte offset	Type	Name	Scale	Unit	Description
0	U4	`iTOW`	-	ms	GPS time of week of the navigation epoch. See section iTOW timestamps in the integration manual for details.
4	I4	`lon`	1e-7	deg	Longitude
8	I4	`lat`	1e-7	deg	Latitude
12	I4	`height`	-	mm	Height above ellipsoid
16	I4	`hMSL`	-	mm	Height above mean sea level
20	U4	`hAcc`	-	mm	Horizontal accuracy estimate
24	U4	`vAcc`	-	mm	Vertical accuracy estimate

3.15.13 UBX-NAV-PVT (0x01 0x07)

3.15.13.1 Navigation position velocity time solution

Message	**UBX-NAV-PVT** **Navigation position velocity time solution**
Type	Periodic/polled
Comment	This message combines position, velocity and time solution, including accuracy figures. Note that during a leap second there may be more or less than 60 seconds in a minute. See description of leap seconds in the integration manual for details.

Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x01	0x07	92	*see below*	CK_A CK_B

Payload description:

Byte offset	Type	Name	Scale	Unit	Description
0	U4	iTOW	-	ms	GPS time of week of the navigation epoch. See section iTOW timestamps in the integration manual for details.
4	U2	year	-	y	Year (UTC)
6	U1	month	-	month	Month, range 1..12 (UTC)
7	U1	day	-	d	Day of month, range 1..31 (UTC)
8	U1	hour	-	h	Hour of day, range 0..23 (UTC)
9	U1	min	-	min	Minute of hour, range 0..59 (UTC)
10	U1	sec	-	s	Seconds of minute, range 0..60 (UTC)
11	X1	valid	-	-	Validity flags
bit 0	U:1	validDate	-	-	1 = valid UTC Date (see section Time validity in the integration manual for details)
bit 1	U:1	validTime	-	-	1 = valid UTC time of day (see section Time validity in the integration manual for details)
bit 2	U:1	fullyResolved	-	-	1 = UTC time of day has been fully resolved (no seconds uncertainty). Cannot be used to check if time is completely solved.
bit 3	U:1	validMag	-	-	1 = valid magnetic declination
12	U4	tAcc	-	ns	Time accuracy estimate (UTC)
16	I4	nano	-	ns	Fraction of second, range -1e9 .. 1e9 (UTC)
20	U1	fixType	-	-	GNSSfix Type: • 0 = no fix • 1 = dead reckoning only • 2 = 2D-fix • 3 = 3D-fix • 4 = GNSS + dead reckoning combined • 5 = time only fix
21	X1	flags	-	-	Fix status flags
bit 0	U:1	gnssFixOK	-	-	1 = valid fix (i.e within DOP & accuracy masks)
bit 1	U:1	diffSoln	-	-	1 = differential corrections were applied
bits 4…2	U:3	psmState	-	-	Power save mode state (see Power management section in the integration manual for details. • 0 = PSM is not active • 1 = Enabled (an intermediate state before Acquisition state • 2 = Acquisition • 3 = Tracking • 4 = Power Optimized Tracking • 5 = Inactive
bit 5	U:1	headVehValid	-	-	1 = heading of vehicle is valid, only set if the receiver is in sensor fusion mode
bits 7…6	U:2	carrSoln	-	-	Carrier phase range solution status: • 0 = no carrier phase range solution • 1 = carrier phase range solution with floating ambiguities • 2 = carrier phase range solution with fixed ambiguities (not supported for protocol versions less than 20.00)
22	X1	flags2	-	-	Additional flags

bit 5	U:1	confirmedAvai	-	-	1 = information about UTC Date and Time of Day validity confirmation is available (see section Time validity in the integration manual for details) This flag is only supported in Protocol Versions 19.00, 19.10, 20.10, 20.20, 20.30, 22.00, 23.00, 23.01, 27 and 28.
bit 6	U:1	confirmedDate	-	-	1 = UTC Date validity could be confirmed (see section Time validity in the integration manual for details)
bit 7	U:1	confirmedTime	-	-	1 = UTC Time of Day could be confirmed (see section Time validity in the integration manual for details)
23	U1	numSV	-	-	Number of satellites used in Nav Solution
24	I4	lon	1e-7	deg	Longitude
28	I4	lat	1e-7	deg	Latitude
32	I4	height	-	mm	Height above ellipsoid
36	I4	hMSL	-	mm	Height above mean sea level
40	U4	hAcc	-	mm	Horizontal accuracy estimate
44	U4	vAcc	-	mm	Vertical accuracy estimate
48	I4	velN	-	mm/s	NED north velocity
52	I4	velE	-	mm/s	NED east velocity
56	I4	velD	-	mm/s	NED down velocity
60	I4	gSpeed	-	mm/s	Ground Speed (2-D)
64	I4	headMot	1e-5	deg	Heading of motion (2-D)
68	U4	sAcc	-	mm/s	Speed accuracy estimate
72	U4	headAcc	1e-5	deg	Heading accuracy estimate (both motion and vehicle)
76	U2	pDOP	0.01	-	Position DOP
78	X2	flags3	-	-	Additional flags
bit 0	U:1	invalidLlh	-	-	1 = Invalid lon, lat, height and hMSL
bits 4…1	U:4	lastCorrectionAge	-	-	Age of the most recently received differential correction: • 0 = Not available • 1 = Age between 0 and 1 second • 2 = Age between 1 (inclusive) and 2 seconds • 3 = Age between 2 (inclusive) and 5 seconds • 4 = Age between 5 (inclusive) and 10 seconds • 5 = Age between 10 (inclusive) and 15 seconds • 6 = Age between 15 (inclusive) and 20 seconds • 7 = Age between 20 (inclusive) and 30 seconds • 8 = Age between 30 (inclusive) and 45 seconds • 9 = Age between 45 (inclusive) and 60 seconds • 10 = Age between 60 (inclusive) and 90 seconds • 11 = Age between 90 (inclusive) and 120 seconds • >=12 = Age greater or equal than 120 seconds
80	U1[4]	reserved0	-	-	Reserved
84	I4	headVeh	1e-5	deg	Heading of vehicle (2-D), this is only valid when headVehValid is set, otherwise the output is set to the heading of motion
88	I2	magDec	1e-2	deg	Magnetic declination. Only supported in ADR 4.10 and later.

90	U2	magAcc	1e-2	deg	Magnetic declination accuracy. Only supported in ADR 4.10 and later.

3.15.14 UBX-NAV-RELPOSNED (0x01 0x3c)

3.15.14.1 Relative positioning information in NED frame

Message	**UBX-NAV-RELPOSNED** **Relative positioning information in NED frame**				
Type	Periodic/polled				
Comment	This message contains the relative position vector from the reference station to the rover, including accuracy figures, in the local topological system defined at the reference station. ☞ The NED frame is defined as the local topological system at the reference station. The relative position vector components in this message, along with their associated accuracies, are given in that local topological system.				
Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x01	0x3c	64	see below	CK_A CK_B

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	U1	version	-	-	Message version (0x01 for this version)
1	U1	reserved0	-	-	Reserved
2	U2	refStationId	-	-	Reference station ID. Must be in the range 0..4095.
4	U4	iTOW	-	ms	GPS time of week of the navigation epoch. See section iTOW timestamps in the integration manual for details.
8	I4	relPosN	-	cm	North component of relative position vector
12	I4	relPosE	-	cm	East component of relative position vector
16	I4	relPosD	-	cm	Down component of relative position vector
20	I4	relPosLength	-	cm	Length of the relative position vector
24	I4	relPosHeading	1e-5	deg	Heading of the relative position vector
28	U1[4]	reserved1	-	-	Reserved
32	I1	relPosHPN	0.1	mm	High-precision North component of relative position vector. Must be in the range -99 to +99. The full North component of the relative position vector, in units of cm, is given by relPosN + (relPosHPN * 1e-2)
33	I1	relPosHPE	0.1	mm	High-precision East component of relative position vector. Must be in the range -99 to +99. The full East component of the relative position vector, in units of cm, is given by relPosE + (relPosHPE * 1e-2)
34	I1	relPosHPD	0.1	mm	High-precision Down component of relative position vector. Must be in the range -99 to +99. The full Down component of the relative position vector, in units of cm, is given by relPosD + (relPosHPD * 1e-2)

35	I1	relPosHPLength	0.1	mm	High-precision component of the length of the relative position vector. Must be in the range -99 to +99. The full length of the relative position vector, in units of cm, is given by relPosLength + (relPosHPLength * 1e-2)
36	U4	accN	0.1	mm	Accuracy of relative position North component
40	U4	accE	0.1	mm	Accuracy of relative position East component
44	U4	accD	0.1	mm	Accuracy of relative position Down component
48	U4	accLength	0.1	mm	Accuracy of length of the relative position vector
52	U4	accHeading	1e-5	deg	Accuracy of heading of the relative position vector
56	U1[4]	reserved2	-	-	Reserved
60	X4	flags	-	-	Flags
bit 0	U:1	gnssFixOK	-	-	A valid fix (i.e within DOP & accuracy masks)
bit 1	U:1	diffSoln	-	-	1 if differential corrections were applied
bit 2	U:1	relPosValid	-	-	1 if relative position components and accuracies are valid and, in moving base mode only, if baseline is valid
bits 4…3	U:2	carrSoln	-	-	Carrier phase range solution status: • 0 = no carrier phase range solution • 1 = carrier phase range solution with floating ambiguities • 2 = carrier phase range solution with fixed ambiguities
bit 5	U:1	isMoving	-	-	1 if the receiver is operating in moving base mode
bit 6	U:1	refPosMiss	-	-	1 if extrapolated reference position was used to compute moving base solution this epoch. (Flag set for protocol versions 27.10, and 27.11, and 31.11)
bit 7	U:1	refObsMiss	-	-	1 if extrapolated reference observations were used to compute moving base solution this epoch. (Flag set for protocol versions 27.10, and 27.11, and 31.11)
bit 8	U:1	relPosHeadingValid	-	-	1 if relPosHeading is valid
bit 9	U:1	relPosNormalized	-	-	1 if the components of the relative position vector (including the high-precision parts) are normalized

3.15.15 UBX-NAV-RESETODO (0x01 0x10)

3.15.15.1 Reset odometer

Message	**UBX-NAV-RESETODO** **Reset odometer**					
Type	Command					
Comment	This message resets the traveled distance computed by the odometer (see UBX-NAV-ODO). UBX-ACK-ACK or UBX-ACK-NAK are returned to indicate success or failure.					
Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x01	0x10	0	*see below*	CK_A CK_B
Payload	This message has no payload.					

3.15.16 UBX-NAV-SAT (0x01 0x35)

3.15.16.1 Satellite information

Message	**UBX-NAV-SAT**					
	Satellite information					
Type	Periodic/polled					
Comment	This message displays information about SVs that are either known to be visible or currently tracked by the receiver. All signal related information corresponds to the subset of signals specified in Signal Identifiers.					
Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x01	0x35	8 + numSvs·12	*see below*	CK_A CK_B

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	U4	`iTOW`	-	ms	GPS time of week of the navigation epoch. See section iTOW timestamps in the integration manual for details.
4	U1	`version`	-	-	Message version (0x01 for this version)
5	U1	`numSvs`	-	-	Number of satellites
6	U1[2]	`reserved0`	-	-	Reserved
Start of repeated group (`numSvs` times)					
8 + n·12	U1	`gnssId`	-	-	GNSS identifier (see Satellite Numbering) for assignment
9 + n·12	U1	`svId`	-	-	Satellite identifier (see Satellite Numbering) for assignment
10 + n·12	U1	`cno`	-	dBHz	Carrier to noise ratio (signal strength)
11 + n·12	I1	`elev`	-	deg	Elevation (range: +/-90), unknown if out of range
12 + n·12	I2	`azim`	-	deg	Azimuth (range 0-360), unknown if elevation is out of range
14 + n·12	I2	`prRes`	0.1	m	Pseudorange residual
16 + n·12	X4	`flags`	-	-	Bitmask
bits 2…0	U:3	`qualityInd`	-	-	Signal quality indicator: • 0 = no signal • 1 = searching signal • 2 = signal acquired • 3 = signal detected but unusable • 4 = code locked and time synchronized • 5, 6, 7 = code and carrier locked and time synchronized
bit 3	U:1	`svUsed`	-	-	1 = Signal in the subset specified in Signal Identifiers is currently being used for navigation
bits 5…4	U:2	`health`	-	-	Signal health flag: • 0 = unknown • 1 = healthy • 2 = unhealthy
bit 6	U:1	`diffCorr`	-	-	1 = differential correction data is available for this SV
bit 7	U:1	`smoothed`	-	-	1 = carrier smoothed pseudorange used
bits 10…8	U:3	`orbitSource`	-	-	Orbit source: • 0 = no orbit information is available for this SV • 1 = ephemeris is used • 2 = almanac is used • 3 = AssistNow Offline orbit is used • 4 = AssistNow Autonomous orbit is used

					• 5, 6, 7 = other orbit information is used
bit 11	U:1	ephAvail	-	-	1 = ephemeris is available for this SV
bit 12	U:1	almAvail	-	-	1 = almanac is available for this SV
bit 13	U:1	anoAvail	-	-	1 = AssistNow Offline data is available for this SV
bit 14	U:1	aopAvail	-	-	1 = AssistNow Autonomous data is available for this SV
bit 16	U:1	sbasCorrUsed	-	-	1 = SBAS corrections have been used for a signal in the subset specified in Signal Identifiers
bit 17	U:1	rtcmCorrUsed	-	-	1 = RTCM corrections have been used for a signal in the subset specified in Signal Identifiers
bit 18	U:1	slasCorrUsed	-	-	1 = QZSS SLAS corrections have been used for a signal in the subset specified in Signal Identifiers
bit 19	U:1	spartnCorrUsed	-	-	1 = SPARTN corrections have been used for a signal in the subset specified in Signal Identifiers
bit 20	U:1	prCorrUsed	-	-	1 = Pseudorange corrections have been used for a signal in the subset specified in Signal Identifiers
bit 21	U:1	crCorrUsed	-	-	1 = Carrier range corrections have been used for a signal in the subset specified in Signal Identifiers
bit 22	U:1	doCorrUsed	-	-	1 = Range rate (Doppler) corrections have been used for a signal in the subset specified in Signal Identifiers
bit 23	U:1	clasCorrUsed	-	-	1 = CLAS corrections have been used for a signal in the subset specified in Signal Identifiers
End of repeated group (numSvs times)					

3.15.17 UBX-NAV-SBAS (0x01 0x32)

3.15.17.1 SBAS status data

Message	**UBX-NAV-SBAS** **SBAS status data**				
Type	Periodic/polled				
Comment	This message outputs the status of the SBAS sub system				

Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x01	0x32	12 + cnt·12	*see below*	CK_A CK_B

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	U4	iTOW	-	ms	GPS time of week of the navigation epoch. See the description of iTOW for details.
4	U1	geo	-	-	PRN Number of the GEO where correction and integrity data is used from
5	U1	mode	-	-	SBAS Mode • 0 Disabled • 1 Enabled integrity • 3 Enabled test mode

6	I1	`sys`	-	-	SBAS System (WAAS/EGNOS/...) • -1 Unknown • 0 WAAS • 1 EGNOS • 2 MSAS • 3 GAGAN • 16 GPS
7	X1	`service`	-	-	SBAS Services available
bit 0	U:1	`Ranging`	-	-	GEO may be used as ranging source
bit 1	U:1	`Corrections`	-	-	GEO is providing correction data
bit 2	U:1	`Integrity`	-	-	GEO is providing integrity
bit 3	U:1	`Testmode`	-	-	GEO is in test mode
bit 4	U:1	`Bad`	-	-	Problem with signal or broadcast data indicated
8	U1	`cnt`	-	-	Number of SV data following
9	X1	`statusFlags`	-	-	SBAS status flags
bits 1...0	U:2	`integrityUsed`	-	-	SBAS integrity used • 0 = Unknown • 1 = Integrity information is not available or SBAS integrity is not enabled • 2 = Receiver uses only GPS satellites for which integrity information is available
10	U1[2]	`reserved0`	-	-	Reserved
Start of repeated group (`cnt` times)					
12 + n·12	U1	`svid`	-	-	SV ID
13 + n·12	U1	`reserved1`	-	-	Reserved
14 + n·12	U1	`udre`	-	-	Monitoring status
15 + n·12	U1	`svSys`	-	-	System (WAAS/EGNOS/...) same as SYS
16 + n·12	U1	`svService`	-	-	Services available same as SERVICE
17 + n·12	U1	`reserved2`	-	-	Reserved
18 + n·12	I2	`prc`	-	cm	Pseudo Range correction in [cm]
20 + n·12	U1[2]	`reserved3`	-	-	Reserved
22 + n·12	I2	`ic`	-	cm	Ionosphere correction in [cm]
End of repeated group (`cnt` times)					

3.15.18 UBX-NAV-SIG (0x01 0x43)

3.15.18.1 Signal information

Message	**UBX-NAV-SIG** **Signal information**					
Type	Periodic/polled					
Comment	This message displays information about signals currently tracked by the receiver. On the F9 platform the maximum number of signals is 120.					
Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x01	0x43	8 + numSigs·16	*see below*	CK_A CK_B

Payload description:

Byte offset	Type	Name	Scale	Unit	Description
0	U4	iTOW	-	ms	GPS time of week of the navigation epoch. See section iTOW timestamps in the integration manual for details.
4	U1	version	-	-	Message version (0x00 for this version)
5	U1	numSigs	-	-	Number of signals
6	U1[2]	reserved0	-	-	Reserved
Start of repeated group (numSigs times)					
8 + n·16	U1	gnssId	-	-	GNSS identifier (see Satellite Numbering) for assignment
9 + n·16	U1	svId	-	-	Satellite identifier (see Satellite Numbering) for assignment
10 + n·16	U1	sigId	-	-	New style signal identifier (see Signal Identifiers)
11 + n·16	U1	freqId	-	-	Only used for GLONASS: This is the frequency slot + 7 (range from 0 to 13)
12 + n·16	I2	prRes	0.1	m	Pseudorange residual
14 + n·16	U1	cno	-	dBHz	Carrier-to-noise density ratio (signal strength)
15 + n·16	U1	qualityInd	-	-	Signal quality indicator: • 0 = no signal • 1 = searching signal • 2 = signal acquired • 3 = signal detected but unusable • 4 = code locked and time synchronized • 5, 6, 7 = code and carrier locked and time synchronized
16 + n·16	U1	corrSource	-	-	Correction source: • 0 = no corrections • 1 = SBAS corrections • 2 = BeiDou corrections • 3 = RTCM2 corrections • 4 = RTCM3 OSR corrections • 5 = RTCM3 SSR corrections • 6 = QZSS SLAS corrections • 7 = SPARTN corrections • 8 = CLAS corrections
17 + n·16	U1	ionoModel	-	-	Ionospheric model used: • 0 = no model • 1 = Klobuchar model transmitted by GPS • 2 = SBAS model • 3 = Klobuchar model transmitted by BeiDou • 8 = Iono delay derived from dual frequency observations
18 + n·16	X2	sigFlags	-	-	Signal related flags
bits 1…0	U:2	health	-	-	Signal health flag: • 0 = unknown • 1 = healthy • 2 = unhealthy
bit 2	U:1	prSmoothed	-	-	1 = Pseudorange has been smoothed
bit 3	U:1	prUsed	-	-	1 = Pseudorange has been used for this signal
bit 4	U:1	crUsed	-	-	1 = Carrier range has been used for this signal
bit 5	U:1	doUsed	-	-	1 = Range rate (Doppler) has been used for this signal

bit 6	U:1	prCorrUsed	-	-	1 = Pseudorange corrections have been used for this signal
bit 7	U:1	crCorrUsed	-	-	1 = Carrier range corrections have been used for this signal
bit 8	U:1	doCorrUsed	-	-	1 = Range rate (Doppler) corrections have been used for this signal
20 + n·16	U1[4]	reserved1	-	-	Reserved
End of repeated group (numSigs times)					

3.15.19 UBX-NAV-SLAS (0x01 0x42)

3.15.19.1 QZSS L1S SLAS status data

Message	**UBX-NAV-SLAS** **QZSS L1S SLAS status data**				
Type	Periodic/polled				
Comment	This message outputs the status of the QZSS L1S SLAS sub system				
Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x01	0x42	20 + cnt·8	see below	CK_A CK_B

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	U4	iTOW	-	ms	GPS time of week of the navigation epoch. See the description of iTOW for details.
4	U1	version	-	-	Message version (0x00 for this version)
5	U1[3]	reserved0	-	-	Reserved
8	I4	gmsLon	1e-3	deg	Longitude of the used ground monitoring station
12	I4	gmsLat	1e-3	deg	Latitude of the used ground monitoring station
16	U1	gmsCode	-	-	Code of the used ground monitoring station according to the QZSS SLAS Interface Specification, available from qzss.go.jp/en/
17	U1	qzssSvId	-	-	Satellite identifier of the QZS/GEO whose correction data is used (see Satellite Numbering)
18	X1	serviceFlags	-	-	Flags regarding SLAS service
bit 0	U:1	gmsAvailable	-	-	1 = Ground monitoring station available
bit 1	U:1	qzssSvAvailable	-	-	1 = Correction providing QZSS SV available
bit 2	U:1	testMode	-	-	1 = Currently used QZSS SV in test mode
19	U1	cnt	-	-	Number of pseudorange corrections following
Start of repeated group (cnt times)					
20 + n·8	U1	gnssId	-	-	GNSS identifier (see Satellite Numbering)
21 + n·8	U1	svId	-	-	Satellite identifier (see Satellite Numbering)
22 + n·8	U1	reserved1	-	-	Reserved
23 + n·8	U1[3]	reserved2	-	-	Reserved
26 + n·8	I2	prc	-	cm	Pseudorange correction

End of repeated group (cnt times)

3.15.20 UBX-NAV-STATUS (0x01 0x03)

3.15.20.1 Receiver navigation status

Message	**UBX-NAV-STATUS**				
	Receiver navigation status				
Type	Periodic/polled				
Comment	See important comments concerning validity of position given in section Navigation output filters in the integration manual.				
Message structure	*Header*	*Class ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x01 0x03	16	*see below*	CK_A CK_B

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	U4	iTOW	-	ms	GPS time of week of the navigation epoch. See section iTOW timestamps in the integration manual for details.
4	U1	gpsFix	-	-	GPSfix Type, this value does **not** qualify a fix as valid and within the limits. See note on flag gpsFixOk below. • 0x00 = no fix • 0x01 = dead reckoning only • 0x02 = 2D-fix • 0x03 = 3D-fix • 0x04 = GPS + dead reckoning combined • 0x05 = Time only fix • 0x06..0xff = reserved
5	X1	flags	-	-	Navigation Status Flags
bit 0	U:1	gpsFixOk	-	-	1 = position and velocity valid and within DOP and ACC Masks.
bit 1	U:1	diffSoln	-	-	1 = differential corrections were applied
bit 2	U:1	wknSet	-	-	1 = Week Number valid (see section Time validity in the integration manual for details)
bit 3	U:1	towSet	-	-	1 = Time of Week valid (see section Time validity in the integration manual for details)
6	X1	fixStat	-	-	Fix Status Information
bit 0	U:1	diffCorr	-	-	1 = differential corrections available
bit 1	U:1	carrSolnValid	-	-	1 = valid carrSoln
bits 7...6	U:2	mapMatching	-	-	map matching status: • 00: none • 01: valid but not used, i.e. map matching data was received, but was too old • 10: valid and used, map matching data was applied • 11: valid and used, map matching data was applied. In case of sensor unavailability map matching data enables dead reckoning. This requires map matched latitude/longitude or heading data.
7	X1	flags2	-	-	further information about navigation output
bits 1...0	U:2	psmState	-	-	power save mode state (not supported for protocol versions less than 13.01)

					• 0 = ACQUISITION [or when psm disabled] • 1 = TRACKING • 2 = POWER OPTIMIZED TRACKING • 3 = INACTIVE
bits 4…3	U:2	spoofDetState	-	-	Spoofing detection state (not supported for protocol versions less than 18.00) • 0: Unknown or deactivated • 1: No spoofing indicated • 2: Spoofing indicated • 3: Multiple spoofing indications Note that the spoofing state value only reflects the detector state for the current navigation epoch. As spoofing can be detected most easily at the transition from real signal to spoofing signal, this is also where the detector is triggered the most. I.e. a value of *1 - No spoofing indicated* does not mean that the receiver is not spoofed, it simply states that the detector was not triggered in this epoch.
bits 7…6	U:2	carrSoln	-	-	Carrier phase range solution status: • 0 = no carrier phase range solution • 1 = carrier phase range solution with floating ambiguities • 2 = carrier phase range solution with fixed ambiguities
8	U4	ttff	-	ms	Time to first fix (millisecond time tag)
12	U4	msss	-	ms	Milliseconds since Startup / Reset

3.15.21 UBX-NAV-SVIN (0x01 0x3b)

3.15.21.1 Survey-in data

Message	**UBX-NAV-SVIN** **Survey-in data**				
Type	Periodic/polled				
Comment	This message contains information about survey-in parameters.				
Message structure	*Header* 0xb5 0x62	*Class* 0x01	*ID* 0x3b	*Length (Bytes)* 40	*Payload* *see below*	*Checksum* CK_A CK_B

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	U1	version	-	-	Message version (0x00 for this version)
1	U1[3]	reserved0	-	-	Reserved
4	U4	iTOW	-	ms	GPS time of week of the navigation epoch. See the description of iTOW for details.
8	U4	dur	-	s	Passed survey-in observation time
12	I4	meanX	-	cm	Current survey-in mean position ECEF X coordinate
16	I4	meanY	-	cm	Current survey-in mean position ECEF Y coordinate
20	I4	meanZ	-	cm	Current survey-in mean position ECEF Z coordinate
24	I1	meanXHP	-	0.1_mm	Current high-precision survey-in mean position ECEF X coordinate. Must be in the range -99..+99. The current survey-in mean position ECEF X coordinate, in units of cm, is given by meanX + (0.01 * meanXHP)

25	I1	meanYHP	-	0.1_mm	Current high-precision survey-in mean position ECEF Y coordinate. Must be in the range -99..+99. The current survey-in mean position ECEF Y coordinate, in units of cm, is given by meanY + (0.01 * meanYHP)
26	I1	meanZHP	-	0.1_mm	Current high-precision survey-in mean position ECEF Z coordinate. Must be in the range -99..+99. The current survey-in mean position ECEF Z coordinate, in units of cm, is given by meanZ + (0.01 * meanZHP)
27	U1	reserved1	-	-	Reserved
28	U4	meanAcc	-	0.1_mm	Current survey-in mean position accuracy
32	U4	obs	-	-	Number of position observations used during survey-in
36	U1	valid	-	-	Survey-in position validity flag, 1 = valid, otherwise 0
37	U1	active	-	-	Survey-in in progress flag, 1 = in-progress, otherwise 0
38	U1[2]	reserved2	-	-	Reserved

3.15.22 UBX-NAV-TIMEBDS (0x01 0x24)

3.15.22.1 BeiDou time solution

Message	**UBX-NAV-TIMEBDS** **BeiDou time solution**				
Type	Periodic/polled				
Comment	This message reports the precise BDS time of the most recent navigation solution including validity flags and an accuracy estimate.				
Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload* / *Checksum*
	0xb5 0x62	0x01	0x24	20	*see below* / CK_A CK_B

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	U4	iTOW	-	ms	GPS time of week of the navigation epoch. See section iTOW timestamps in the integration manual for details.
4	U4	SOW	-	s	BDS time of week (rounded to seconds)
8	I4	fSOW	-	ns	Fractional part of SOW (range: +/-500000000). The precise BDS time of week in seconds is: SOW + fSOW * 1e-9
12	I2	week	-	-	BDS week number of the navigation epoch
14	I1	leapS	-	s	BDS leap seconds (BDS-UTC)
15	X1	valid	-	-	Validity Flags
bit 0	U:1	sowValid	-	-	1 = Valid SOW and fSOW (see section Time validity in the integration manual for details)
bit 1	U:1	weekValid	-	-	1 = Valid week (see section Time validity in the integration manual for details)
bit 2	U:1	leapSValid	-	-	1 = Valid leap second

16	U4	`tAcc`	-	ns	Time Accuracy Estimate

3.15.23 UBX-NAV-TIMEGAL (0x01 0x25)

3.15.23.1 Galileo time solution

Message	**UBX-NAV-TIMEGAL** **Galileo time solution**					
Type	Periodic/polled					
Comment	This message reports the precise Galileo time of the most recent navigation solution including validity flags and an accuracy estimate.					
Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x01	0x25	20	*see below*	CK_A CK_B

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	U4	`iTOW`	-	ms	GPS time of week of the navigation epoch. See section iTOW timestamps in the integration manual for details.
4	U4	`galTow`	-	s	Galileo time of week (rounded to seconds)
8	I4	`fGalTow`	-	ns	Fractional part of the Galileo time of week (range: +/-500000000). The precise Galileo time of week in seconds is: `galTow + fGalTow * 1e-9`
12	I2	`galWno`	-	-	Galileo week number
14	I1	`leapS`	-	s	Galileo leap seconds (Galileo-UTC)
15	X1	`valid`	-	-	Validity Flags
bit 0	U:1	`galTowValid`	-	-	1 = Valid galTow and fGalTow (see section Time validity in the integration manual for details)
bit 1	U:1	`galWnoValid`	-	-	1 = Valid galWno (see section Time validity in the integration manual for details)
bit 2	U:1	`leapSValid`	-	-	1 = Valid leapS
16	U4	`tAcc`	-	ns	Time Accuracy Estimate

3.15.24 UBX-NAV-TIMEGLO (0x01 0x23)

3.15.24.1 GLONASS time solution

Message	**UBX-NAV-TIMEGLO** **GLONASS time solution**					
Type	Periodic/polled					
Comment	This message reports the precise GLO time of the most recent navigation solution including validity flags and an accuracy estimate.					
Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x01	0x23	20	*see below*	CK_A CK_B

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	U4	`iTOW`	-	ms	GPS time of week of the navigation epoch. See section iTOW timestamps in the integration manual for details.

4	U4	TOD	-	s	GLONASS time of day (rounded to integer seconds)
8	I4	fTOD	-	ns	Fractional part of TOD (range: +/-500000000). The precise GLONASS time of day in seconds is: `TOD + fTOD * 1e-9`
12	U2	Nt	-	days	Current date (range: 1-1461), starting at 1 from the 1st Jan of the year indicated by N4 and ending at 1461 at the 31st Dec of the third year after that indicated by N4
14	U1	N4	-	-	Four-year interval number starting from 1996 (1=1996, 2=2000, 3=2004...)
15	X1	valid	-	-	Validity flags
bit 0	U:1	todValid	-	-	1 = Valid TOD and fTOD (see section Time validity in the integration manual for details)
bit 1	U:1	dateValid	-	-	1 = Valid N4 and Nt (see section Time validity in the integration manual for details)
16	U4	tAcc	-	ns	Time Accuracy Estimate

3.15.25 UBX-NAV-TIMEGPS (0x01 0x20)

3.15.25.1 GPS time solution

Message	**UBX-NAV-TIMEGPS** **GPS time solution**				
Type	Periodic/polled				
Comment	This message reports the precise GPS time of the most recent navigation solution including validity flags and an accuracy estimate.				
Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x01	0x20	16	*see below*	CK_A CK_B

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	U4	iTOW	-	ms	GPS time of week of the navigation epoch. See section iTOW timestamps in the integration manual for details.
4	I4	fTOW	-	ns	Fractional part of iTOW (range: +/-500000). The precise GPS time of week in seconds is: `(iTOW * 1e-3) + (fTOW * 1e-9)`
8	I2	week	-	-	GPS week number of the navigation epoch
10	I1	leapS	-	s	GPS leap seconds (GPS-UTC)
11	X1	valid	-	-	Validity Flags
bit 0	U:1	towValid	-	-	1 = Valid GPS time of week (iTOW & fTOW, (see section Time validity in the integration manual for details)
bit 1	U:1	weekValid	-	-	1 = Valid GPS week number (see section Time validity in the integration manual for details)
bit 2	U:1	leapSValid	-	-	1 = Valid GPS leap seconds
12	U4	tAcc	-	ns	Time Accuracy Estimate

3.15.26 UBX-NAV-TIMELS (0x01 0x26)

3.15.26.1 Leap second event information

Message	UBX-NAV-TIMELS				
	Leap second event information				
Type	Periodic/polled				
Comment	Information about the upcoming leap second event if one is scheduled.				
Message structure	*Header*	*Class* *ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x01 0x26	24	*see below*	CK_A CK_B

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	U4	`iTOW`	-	ms	GPS time of week of the navigation epoch. See section iTOW timestamps in the integration manual for details.
4	U1	`version`	-	-	Message version (0x00 for this version)
5	U1[3]	`reserved0`	-	-	Reserved
8	U1	`srcOfCurrLs`	-	-	Information source for the current number of leap seconds. • 0 = Default (hardcoded in the firmware, can be outdated) • 1 = Derived from time difference between GPS and GLONASS time • 2 = GPS • 3 = SBAS • 4 = BeiDou • 5 = Galileo • 6 = Aided data • 7 = Configured • 8 = NavIC • 255 = Unknown
9	I1	`currLs`	-	s	Current number of leap seconds since start of GPS time (Jan 6, 1980). It reflects how much GPS time is ahead of UTC time. Galileo number of leap seconds is the same as GPS. BeiDou number of leap seconds is 14 less than GPS. GLONASS follows UTC time, so no leap seconds.
10	U1	`srcOfLsChange`	-	-	Information source for the future leap second event. • 0 = No source • 2 = GPS • 3 = SBAS • 4 = BeiDou • 5 = Galileo • 6 = GLONASS • 7 = NavIC
11	I1	`lsChange`	-	s	Future leap second change if one is scheduled. +1 = positive leap second, -1 = negative leap second, 0 = no future leap second event scheduled or no information available.
12	I4	`timeToLsEvent`	-	s	Number of seconds until the next leap second event, or from the last leap second event if no future event scheduled. If > 0 event is in the future, = 0 event is now, < 0 event is in the past. Valid only if validTimeToLsEvent = 1.

16	U2	dateOfLsGpsWn	-	-	GPS week number (WN) of the next leap second event or the last one if no future event scheduled. Valid only if validTimeToLsEvent = 1.
18	U2	dateOfLsGpsDn	-	-	GPS day of week number (DN) for the next leap second event or the last one if no future event scheduled. Valid only if validTimeToLsEvent = 1. (GPS and Galileo DN: from 1 = Sun to 7 = Sat. BeiDou DN: from 0 = Sun to 6 = Sat.)
20	U1[3]	reserved1	-	-	Reserved
23	X1	valid	-	-	Validity flags
bit 0	U:1	validCurrLs	-	-	1 = Valid current number of leap seconds value.
bit 1	U:1	validTimeToLsEvent	-	-	1 = Valid time to next leap second event or from the last leap second event if no future event scheduled.

3.15.27 UBX-NAV-TIMEQZSS (0x01 0x27)

3.15.27.1 QZSS time solution

Message	**UBX-NAV-TIMEQZSS** **QZSS time solution**					
Type	Periodic/polled					
Comment	This message reports the precise QZSS time of the most recent navigation solution including validity flags and an accuracy estimate. See the Clocks and time section in the integration manual for details.					
Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x01	0x27	20	*see below*	CK_A CK_B

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	U4	iTOW	-	ms	GPS time of week of the navigation epoch.
4	U4	qzssTow	-	s	QZSS time of week (rounded to seconds)
8	I4	fQzssTow	-	ns	Fractional part of QZSS time of week (range: +/-500000000). The precise QZSS time of week in seconds is: `qzssTow + (fQzssTow * 1e-9)`
12	I2	qzssWno	-	-	QZSS week number of the navigation epoch
14	I1	leapS	-	s	QZSS leap seconds (QZSS-UTC)
15	X1	valid	-	-	Validity Flags
bit 0	U:1	qzssTowValid	-	-	1 = Valid QZSS time of week (qzssTow and fQzssTow)
bit 1	U:1	qzssWnoValid	-	-	1 = Valid QZSS week number
bit 2	U:1	leapSValid	-	-	1 = Valid QZSS leap seconds
16	U4	tAcc	-	ns	Time Accuracy Estimate

3.15.28 UBX-NAV-TIMEUTC (0x01 0x21)

3.15.28.1 UTC time solution

Message	**UBX-NAV-TIMEUTC** **UTC time solution**
Type	Periodic/polled

Comment	Note that during a leap second there may be more or less than 60 seconds in a minute. See the description of leap seconds in the integration manual for details.					
Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x01	0x21	20	*see below*	CK_A CK_B

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	U4	`iTOW`	-	ms	GPS time of week of the navigation epoch. See section iTOW timestamps in the integration manual for details.
4	U4	`tAcc`	-	ns	Time accuracy estimate (UTC)
8	I4	`nano`	-	ns	Fraction of second, range -1e9 .. 1e9 (UTC)
12	U2	`year`	-	y	Year, range 1999..2099 (UTC)
14	U1	`month`	-	month	Month, range 1..12 (UTC)
15	U1	`day`	-	d	Day of month, range 1..31 (UTC)
16	U1	`hour`	-	h	Hour of day, range 0..23 (UTC)
17	U1	`min`	-	min	Minute of hour, range 0..59 (UTC)
18	U1	`sec`	-	s	Seconds of minute, range 0..60 (UTC)
19	X1	`valid`	-	-	Validity Flags
bit 0	U:1	`validTOW`	-	-	1 = Valid Time of Week (see section Time validity in the integration manual for details)
bit 1	U:1	`validWKN`	-	-	1 = Valid Week Number (see section Time validity in the integration manual for details)
bit 2	U:1	`validUTC`	-	-	1 = Valid UTC Time
bits 7…4	U:4	`utcStandard`	-	-	UTC standard identifier. (Not supported for protocol versions less than 15.00) • 0 = Information not available • 1 = Communications Research Labratory (CRL), Tokyo, Japan • 2 = National Institute of Standards and Technology (NIST) • 3 = U.S. Naval Observatory (USNO) • 4 = International Bureau of Weights and Measures (BIPM) • 5 = European laboratories • 6 = Former Soviet Union (SU) • 7 = National Time Service Center (NTSC), China • 8 = National Physics Laboratory India (NPLI) • 15 = Unknown

3.15.29 UBX-NAV-VELECEF (0x01 0x11)

3.15.29.1 Velocity solution in ECEF

Message	**UBX-NAV-VELECEF** **Velocity solution in ECEF**					
Type	Periodic/polled					
Comment	See important comments concerning validity of position given in section Navigation output filters in the integration manual.					
Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x01	0x11	20	*see below*	CK_A CK_B

Payload description:					
Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	U4	`iTOW`	-	ms	GPS time of week of the navigation epoch. See section iTOW timestamps in the integration manual for details.
4	I4	`ecefVX`	-	cm/s	ECEF X velocity
8	I4	`ecefVY`	-	cm/s	ECEF Y velocity
12	I4	`ecefVZ`	-	cm/s	ECEF Z velocity
16	U4	`sAcc`	-	cm/s	Speed accuracy estimate

3.15.30 UBX-NAV-VELNED (0x01 0x12)

3.15.30.1 Velocity solution in NED frame

Message	**UBX-NAV-VELNED** **Velocity solution in NED frame**					
Type	Periodic/polled					
Comment	See important comments concerning validity of position given in section Navigation output filters in the integration manual.					
Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x01	0x12	36	*see below*	CK_A CK_B

Payload description:					
Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	U4	`iTOW`	-	ms	GPS time of week of the navigation epoch. See section iTOW timestamps in the integration manual for details.
4	I4	`velN`	-	cm/s	North velocity component
8	I4	`velE`	-	cm/s	East velocity component
12	I4	`velD`	-	cm/s	Down velocity component
16	U4	`speed`	-	cm/s	Speed (3-D)
20	U4	`gSpeed`	-	cm/s	Ground speed (2-D)
24	I4	`heading`	1e-5	deg	Heading of motion 2-D
28	U4	`sAcc`	-	cm/s	Speed accuracy Estimate
32	U4	`cAcc`	1e-5	deg	Course / Heading accuracy estimate

3.16 UBX-NAV2 (0x29)

The messages in the UBX-NAV2 class are used to output navigation results and data, such as position, altitude and velocity in a number of formats, and status flags and accuracy estimate figures, or satellite and signal information. The messages are generated with the configured navigation rate.

3.16.1 UBX-NAV2-CLOCK (0x29 0x22)

3.16.1.1 Clock solution

Message	**UBX-NAV2-CLOCK** **Clock solution**
Type	Periodic/polled

Comment							
Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*		*Payload*	*Checksum*
	0xb5 0x62	0x29	0x22	20		*see below*	CK_A CK_B

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	U4	`iTOW`	-	ms	GPS time of week of the navigation epoch. See section Navigation epochs in the integration manual for details. See section iTOW timestamps in the integration manual for details.
4	I4	`clkB`	-	ns	Clock bias
8	I4	`clkD`	-	ns/s	Clock drift
12	U4	`tAcc`	-	ns	Time accuracy estimate
16	U4	`fAcc`	-	ps/s	Frequency accuracy estimate

3.16.2 UBX-NAV2-COV (0x29 0x36)

3.16.2.1 Covariance matrices

Message	**UBX-NAV2-COV** **Covariance matrices**						
Type	Periodic/polled						
Comment	This message outputs the covariance matrices for the position and velocity solutions in the topocentric coordinate system defined as the local-level North (N), East (E), Down (D) frame. As the covariance matrices are symmetric, only the upper triangular part is output.						
Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*		*Payload*	*Checksum*
	0xb5 0x62	0x29	0x36	64		*see below*	CK_A CK_B

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	U4	`iTOW`	-	ms	GPS time of week of the navigation epoch. See section iTOW timestamps in the integration manual for details.
4	U1	`version`	-	-	Message version (0x00 for this version)
5	U1	`posCovValid`	-	-	Position covariance matrix validity flag
6	U1	`velCovValid`	-	-	Velocity covariance matrix validity flag
7	U1[9]	`reserved0`	-	-	Reserved
16	R4	`posCovNN`	-	m^2	Position covariance matrix value p_NN
20	R4	`posCovNE`	-	m^2	Position covariance matrix value p_NE
24	R4	`posCovND`	-	m^2	Position covariance matrix value p_ND
28	R4	`posCovEE`	-	m^2	Position covariance matrix value p_EE
32	R4	`posCovED`	-	m^2	Position covariance matrix value p_ED
36	R4	`posCovDD`	-	m^2	Position covariance matrix value p_DD
40	R4	`velCovNN`	-	m^2/s^2	Velocity covariance matrix value v_NN
44	R4	`velCovNE`	-	m^2/s^2	Velocity covariance matrix value v_NE
48	R4	`velCovND`	-	m^2/s^2	Velocity covariance matrix value v_ND
52	R4	`velCovEE`	-	m^2/s^2	Velocity covariance matrix value v_EE

56	R4	velCovED	-	m^2/s^2	Velocity covariance matrix value v_ED
60	R4	velCovDD	-	m^2/s^2	Velocity covariance matrix value v_DD

3.16.3 UBX-NAV2-DOP (0x29 0x04)

3.16.3.1 Dilution of precision

Message	**UBX-NAV2-DOP** **Dilution of precision**				
Type	Periodic/polled				
Comment	• DOP values are dimensionless. • All DOP values are scaled by a factor of 100. If the unit transmits a value of e.g. 156, the DOP value is 1.56.				
Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x29	0x04	18	*see below*	CK_A CK_B

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	U4	iTOW	-	ms	GPS time of week of the navigation epoch. See section iTOW timestamps in the integration manual for details.
4	U2	gDOP	0.01	-	Geometric DOP
6	U2	pDOP	0.01	-	Position DOP
8	U2	tDOP	0.01	-	Time DOP
10	U2	vDOP	0.01	-	Vertical DOP
12	U2	hDOP	0.01	-	Horizontal DOP
14	U2	nDOP	0.01	-	Northing DOP
16	U2	eDOP	0.01	-	Easting DOP

3.16.4 UBX-NAV2-EOE (0x29 0x61)

3.16.4.1 End of epoch

Message	**UBX-NAV2-EOE** **End of epoch**					
Type	Periodic					
Comment	This message is intended to be used as a marker to collect all navigation messages of an epoch. It is output after all enabled NAV class messages (except UBX-NAV-HNR) and after all enabled NMEA messages.					
Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x29	0x61	4	*see below*	CK_A CK_B

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	U4	iTOW	-	ms	GPS time of week of the navigation epoch. See section iTOW timestamps in the integration manual for details.

3.16.5 UBX-NAV2-ODO (0x29 0x09)

3.16.5.1 Odometer solution

Message	**UBX-NAV2-ODO** **Odometer solution**					
Type	Periodic/polled					
Comment	This message outputs the traveled distance since last reset (see UBX-NAV-RESETODO) together with an associated estimated accuracy and the total cumulated ground distance (can only be reset by a cold start of the receiver).					
Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x29	0x09	20	*see below*	CK_A CK_B

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	U1	`version`	-	-	Message version (0x00 for this version)
1	U1[3]	`reserved0`	-	-	Reserved
4	U4	`iTOW`	-	ms	GPS time of week of the navigation epoch. See section iTOW timestamps in the integration manual for details.
8	U4	`distance`	-	m	Ground distance since last reset
12	U4	`totalDistance`	-	m	Total cumulative ground distance
16	U4	`distanceStd`	-	m	Ground distance accuracy (1-sigma)

3.16.6 UBX-NAV2-POSECEF (0x29 0x01)

3.16.6.1 Position solution in ECEF

Message	**UBX-NAV2-POSECEF** **Position solution in ECEF**					
Type	Periodic/polled					
Comment	See important comments concerning validity of position given in section Navigation output filters in the integration manual.					
Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x29	0x01	20	*see below*	CK_A CK_B

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	U4	`iTOW`	-	ms	GPS time of week of the navigation epoch. See section iTOW timestamps in the integration manual for details.
4	I4	`ecefX`	-	cm	ECEF X coordinate
8	I4	`ecefY`	-	cm	ECEF Y coordinate
12	I4	`ecefZ`	-	cm	ECEF Z coordinate
16	U4	`pAcc`	-	cm	Position Accuracy Estimate

3.16.7 UBX-NAV2-POSLLH (0x29 0x02)

3.16.7.1 Geodetic position solution

Message	**UBX-NAV2-POSLLH** **Geodetic position solution**
Type	Periodic/polled

Comment	See important comments concerning validity of position given in section Navigation output filters in the integration manual. This message outputs the Geodetic position in the currently selected ellipsoid. The default is the WGS84 Ellipsoid, but can be changed with the message CFG-NAVSPG-USE_USRDAT.					
Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x29	0x02	28	*see below*	CK_A CK_B

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	U4	`iTOW`	-	ms	GPS time of week of the navigation epoch. See section iTOW timestamps in the integration manual for details.
4	I4	`lon`	1e-7	deg	Longitude
8	I4	`lat`	1e-7	deg	Latitude
12	I4	`height`	-	mm	Height above ellipsoid
16	I4	`hMSL`	-	mm	Height above mean sea level
20	U4	`hAcc`	-	mm	Horizontal accuracy estimate
24	U4	`vAcc`	-	mm	Vertical accuracy estimate

3.16.8 UBX-NAV2-PVT (0x29 0x07)

3.16.8.1 Navigation position velocity time solution

Message	**UBX-NAV2-PVT** **Navigation position velocity time solution**					
Type	Periodic/polled					
Comment	This message combines position, velocity and time solution, including accuracy figures. Note that during a leap second there may be more or less than 60 seconds in a minute. See description of leap seconds in the integration manual for details.					
Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x29	0x07	92	*see below*	CK_A CK_B

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	U4	`iTOW`	-	ms	GPS time of week of the navigation epoch. See section iTOW timestamps in the integration manual for details.
4	U2	`year`	-	y	Year (UTC)
6	U1	`month`	-	month	Month, range 1..12 (UTC)
7	U1	`day`	-	d	Day of month, range 1..31 (UTC)
8	U1	`hour`	-	h	Hour of day, range 0..23 (UTC)
9	U1	`min`	-	min	Minute of hour, range 0..59 (UTC)
10	U1	`sec`	-	s	Seconds of minute, range 0..60 (UTC)
11	X1	`valid`	-	-	Validity flags
bit 0	U:1	`validDate`	-	-	1 = valid UTC Date (see section Time validity in the integration manual for details)
bit 1	U:1	`validTime`	-	-	1 = valid UTC time of day (see section Time validity in the integration manual for details)

bit 2	U:1	fullyResolved	-	-	1 = UTC time of day has been fully resolved (no seconds uncertainty). Cannot be used to check if time is completely solved.
bit 3	U:1	validMag	-	-	1 = valid magnetic declination
12	U4	tAcc	-	ns	Time accuracy estimate (UTC)
16	I4	nano	-	ns	Fraction of second, range -1e9 .. 1e9 (UTC)
20	U1	fixType	-	-	GNSSfix Type: • 0 = no fix • 1 = dead reckoning only • 2 = 2D-fix • 3 = 3D-fix • 4 = GNSS + dead reckoning combined • 5 = time only fix
21	X1	flags	-	-	Fix status flags
bit 0	U:1	gnssFixOK	-	-	1 = valid fix (i.e within DOP & accuracy masks)
bit 1	U:1	diffSoln	-	-	1 = differential corrections were applied
bits 4…2	U:3	psmState	-	-	Power save mode state (see Power management section in the integration manual for details. • 0 = PSM is not active • 1 = Enabled (an intermediate state before Acquisition state • 2 = Acquisition • 3 = Tracking • 4 = Power Optimized Tracking • 5 = Inactive
bit 5	U:1	headVehValid	-	-	1 = heading of vehicle is valid, only set if the receiver is in sensor fusion mode
bits 7…6	U:2	carrSoln	-	-	Carrier phase range solution status: • 0 = no carrier phase range solution • 1 = carrier phase range solution with floating ambiguities • 2 = carrier phase range solution with fixed ambiguities (not supported for protocol versions less than 20.00)
22	X1	flags2	-	-	Additional flags
bit 5	U:1	confirmedAvai	-	-	1 = information about UTC Date and Time of Day validity confirmation is available (see section Time validity in the integration manual for details) This flag is only supported in Protocol Versions 19.00, 19.10, 20.10, 20.20, 20.30, 22.00, 23.00, 23.01, 27 and 28.
bit 6	U:1	confirmedDate	-	-	1 = UTC Date validity could be confirmed (see section Time validity in the integration manual for details)
bit 7	U:1	confirmedTime	-	-	1 = UTC Time of Day could be confirmed (see section Time validity in the integration manual for details)
23	U1	numSV	-	-	Number of satellites used in Nav Solution
24	I4	lon	1e-7	deg	Longitude
28	I4	lat	1e-7	deg	Latitude
32	I4	height	-	mm	Height above ellipsoid
36	I4	hMSL	-	mm	Height above mean sea level
40	U4	hAcc	-	mm	Horizontal accuracy estimate

44	U4	`vAcc`	-	mm	Vertical accuracy estimate
48	I4	`velN`	-	mm/s	NED north velocity
52	I4	`velE`	-	mm/s	NED east velocity
56	I4	`velD`	-	mm/s	NED down velocity
60	I4	`gSpeed`	-	mm/s	Ground Speed (2-D)
64	I4	`headMot`	1e-5	deg	Heading of motion (2-D)
68	U4	`sAcc`	-	mm/s	Speed accuracy estimate
72	U4	`headAcc`	1e-5	deg	Heading accuracy estimate (both motion and vehicle)
76	U2	`pDOP`	0.01	-	Position DOP
78	X2	`flags3`	-	-	Additional flags
bit 0	U:1	`invalidLlh`	-	-	1 = Invalid lon, lat, height and hMSL
bits 4…1	U:4	`lastCorrectionAge`	-	-	Age of the most recently received differential correction: • 0 = Not available • 1 = Age between 0 and 1 second • 2 = Age between 1 (inclusive) and 2 seconds • 3 = Age between 2 (inclusive) and 5 seconds • 4 = Age between 5 (inclusive) and 10 seconds • 5 = Age between 10 (inclusive) and 15 seconds • 6 = Age between 15 (inclusive) and 20 seconds • 7 = Age between 20 (inclusive) and 30 seconds • 8 = Age between 30 (inclusive) and 45 seconds • 9 = Age between 45 (inclusive) and 60 seconds • 10 = Age between 60 (inclusive) and 90 seconds • 11 = Age between 90 (inclusive) and 120 seconds • >=12 = Age greater or equal than 120 seconds
80	U1[4]	`reserved0`	-	-	Reserved
84	I4	`headVeh`	1e-5	deg	Heading of vehicle (2-D), this is only valid when headVehValid is set, otherwise the output is set to the heading of motion
88	I2	`magDec`	1e-2	deg	Magnetic declination. Only supported in ADR 4.10 and later.
90	U2	`magAcc`	1e-2	deg	Magnetic declination accuracy. Only supported in ADR 4.10 and later.

3.16.9 UBX-NAV2-SAT (0x29 0x35)

3.16.9.1 Satellite information

Message	**UBX-NAV2-SAT** **Satellite information**				
Type	Periodic/polled				
Comment	This message displays information about SVs that are either known to be visible or currently tracked by the receiver. All signal related information corresponds to the subset of signals specified in Signal Identifiers.				
Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x29	0x35	8 + numSvs·12	see below	CK_A CK_B

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*

0	U4	`iTOW`	-	ms	GPS time of week of the navigation epoch. See section iTOW timestamps in the integration manual for details.
4	U1	`version`	-	-	Message version (0x01 for this version)
5	U1	`numSvs`	-	-	Number of satellites
6	U1[2]	`reserved0`	-	-	Reserved
Start of repeated group (`numSvs` times)					
8 + n·12	U1	`gnssId`	-	-	GNSS identifier (see Satellite Numbering) for assignment
9 + n·12	U1	`svId`	-	-	Satellite identifier (see Satellite Numbering) for assignment
10 + n·12	U1	`cno`	-	dBHz	Carrier to noise ratio (signal strength)
11 + n·12	I1	`elev`	-	deg	Elevation (range: +/-90), unknown if out of range
12 + n·12	I2	`azim`	-	deg	Azimuth (range 0-360), unknown if elevation is out of range
14 + n·12	I2	`prRes`	0.1	m	Pseudorange residual
16 + n·12	X4	`flags`	-	-	Bitmask
bits 2...0	U:3	`qualityInd`	-	-	Signal quality indicator: • 0 = no signal • 1 = searching signal • 2 = signal acquired • 3 = signal detected but unusable • 4 = code locked and time synchronized • 5, 6, 7 = code and carrier locked and time synchronized
bit 3	U:1	`svUsed`	-	-	1 = Signal in the subset specified in Signal Identifiers is currently being used for navigation
bits 5...4	U:2	`health`	-	-	Signal health flag: • 0 = unknown • 1 = healthy • 2 = unhealthy
bit 6	U:1	`diffCorr`	-	-	1 = differential correction data is available for this SV
bit 7	U:1	`smoothed`	-	-	1 = carrier smoothed pseudorange used
bits 10...8	U:3	`orbitSource`	-	-	Orbit source: • 0 = no orbit information is available for this SV • 1 = ephemeris is used • 2 = almanac is used • 3 = AssistNow Offline orbit is used • 4 = AssistNow Autonomous orbit is used • 5, 6, 7 = other orbit information is used
bit 11	U:1	`ephAvail`	-	-	1 = ephemeris is available for this SV
bit 12	U:1	`almAvail`	-	-	1 = almanac is available for this SV
bit 13	U:1	`anoAvail`	-	-	1 = AssistNow Offline data is available for this SV
bit 14	U:1	`aopAvail`	-	-	1 = AssistNow Autonomous data is available for this SV
bit 16	U:1	`sbasCorrUsed`	-	-	1 = SBAS corrections have been used for a signal in the subset specified in Signal Identifiers
bit 17	U:1	`rtcmCorrUsed`	-	-	1 = RTCM corrections have been used for a signal in the subset specified in Signal Identifiers

bit 18	U:1	`slasCorrUsed`	-	-	1 = QZSS SLAS corrections have been used for a signal in the subset specified in Signal Identifiers
bit 19	U:1	`spartnCorrUsed`	-	-	1 = SPARTN corrections have been used for a signal in the subset specified in Signal Identifiers
bit 20	U:1	`prCorrUsed`	-	-	1 = Pseudorange corrections have been used for a signal in the subset specified in Signal Identifiers
bit 21	U:1	`crCorrUsed`	-	-	1 = Carrier range corrections have been used for a signal in the subset specified in Signal Identifiers
bit 22	U:1	`doCorrUsed`	-	-	1 = Range rate (Doppler) corrections have been used for a signal in the subset specified in Signal Identifiers
bit 23	U:1	`clasCorrUsed`	-	-	1 = CLAS corrections have been used for a signal in the subset specified in Signal Identifiers
End of repeated group (`numSvs` times)					

3.16.10 UBX-NAV2-SBAS (0x29 0x32)

3.16.10.1 SBAS status data

Message	**UBX-NAV2-SBAS** **SBAS status data**					
Type	Periodic/polled					
Comment	This message outputs the status of the SBAS sub system					
Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x29	0x32	12 + cnt·12	*see below*	CK_A CK_B

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	U4	`iTOW`	-	ms	GPS time of week of the navigation epoch. See the description of iTOW for details.
4	U1	`geo`	-	-	PRN Number of the GEO where correction and integrity data is used from
5	U1	`mode`	-	-	SBAS Mode • 0 Disabled • 1 Enabled integrity • 3 Enabled test mode
6	I1	`sys`	-	-	SBAS System (WAAS/EGNOS/...) • -1 Unknown • 0 WAAS • 1 EGNOS • 2 MSAS • 3 GAGAN • 16 GPS
7	X1	`service`	-	-	SBAS Services available
bit 0	U:1	`Ranging`	-	-	GEO may be used as ranging source
bit 1	U:1	`Corrections`	-	-	GEO is providing correction data
bit 2	U:1	`Integrity`	-	-	GEO is providing integrity
bit 3	U:1	`Testmode`	-	-	GEO is in test mode
bit 4	U:1	`Bad`	-	-	Problem with signal or broadcast data indicated
8	U1	`cnt`	-	-	Number of SV data following
9	X1	`statusFlags`	-	-	SBAS status flags

bits 1…0	U:2	`integrityUsed`	-	-	SBAS integrity used • 0 = Unknown • 1 = Integrity information is not available or SBAS integrity is not enabled • 2 = Receiver uses only GPS satellites for which integrity information is available
10	U1[2]	`reserved0`	-	-	Reserved
Start of repeated group (`cnt` times)					
12 + n·12	U1	`svid`	-	-	SV ID
13 + n·12	U1	`reserved1`	-	-	Reserved
14 + n·12	U1	`udre`	-	-	Monitoring status
15 + n·12	U1	`svSys`	-	-	System (WAAS/EGNOS/...) same as SYS
16 + n·12	U1	`svService`	-	-	Services available same as SERVICE
17 + n·12	U1	`reserved2`	-	-	Reserved
18 + n·12	I2	`prc`	-	cm	Pseudo Range correction in [cm]
20 + n·12	U1[2]	`reserved3`	-	-	Reserved
22 + n·12	I2	`ic`	-	cm	Ionosphere correction in [cm]
End of repeated group (`cnt` times)					

3.16.11 UBX-NAV2-SIG (0x29 0x43)

3.16.11.1 Signal information

Message	**UBX-NAV2-SIG** **Signal information**				
Type	Periodic/polled				
Comment	This message displays information about signals currently tracked by the receiver. On the F9 platform the maximum number of signals is 120.				
Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x29	0x43	8 + numSigs·16	*see below*	CK_A CK_B

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	U4	`iTOW`	-	ms	GPS time of week of the navigation epoch. See section iTOW timestamps in the integration manual for details.
4	U1	`version`	-	-	Message version (0x00 for this version)
5	U1	`numSigs`	-	-	Number of signals
6	U1[2]	`reserved0`	-	-	Reserved
Start of repeated group (`numSigs` times)					
8 + n·16	U1	`gnssId`	-	-	GNSS identifier (see Satellite Numbering) for assignment
9 + n·16	U1	`svId`	-	-	Satellite identifier (see Satellite Numbering) for assignment
10 + n·16	U1	`sigId`	-	-	New style signal identifier (see Signal Identifiers)

11 + n·16	U1	`freqId`	-	-	Only used for GLONASS: This is the frequency slot + 7 (range from 0 to 13)
12 + n·16	I2	`prRes`	0.1	m	Pseudorange residual
14 + n·16	U1	`cno`	-	dBHz	Carrier-to-noise density ratio (signal strength)
15 + n·16	U1	`qualityInd`	-	-	Signal quality indicator: • 0 = no signal • 1 = searching signal • 2 = signal acquired • 3 = signal detected but unusable • 4 = code locked and time synchronized • 5, 6, 7 = code and carrier locked and time synchronized
16 + n·16	U1	`corrSource`	-	-	Correction source: • 0 = no corrections • 1 = SBAS corrections • 2 = BeiDou corrections • 3 = RTCM2 corrections • 4 = RTCM3 OSR corrections • 5 = RTCM3 SSR corrections • 6 = QZSS SLAS corrections • 7 = SPARTN corrections • 8 = CLAS corrections
17 + n·16	U1	`ionoModel`	-	-	Ionospheric model used: • 0 = no model • 1 = Klobuchar model transmitted by GPS • 2 = SBAS model • 3 = Klobuchar model transmitted by BeiDou • 8 = Iono delay derived from dual frequency observations
18 + n·16	X2	`sigFlags`	-	-	Signal related flags
bits 1…0	U:2	`health`	-	-	Signal health flag: • 0 = unknown • 1 = healthy • 2 = unhealthy
bit 2	U:1	`prSmoothed`	-	-	1 = Pseudorange has been smoothed
bit 3	U:1	`prUsed`	-	-	1 = Pseudorange has been used for this signal
bit 4	U:1	`crUsed`	-	-	1 = Carrier range has been used for this signal
bit 5	U:1	`doUsed`	-	-	1 = Range rate (Doppler) has been used for this signal
bit 6	U:1	`prCorrUsed`	-	-	1 = Pseudorange corrections have been used for this signal
bit 7	U:1	`crCorrUsed`	-	-	1 = Carrier range corrections have been used for this signal
bit 8	U:1	`doCorrUsed`	-	-	1 = Range rate (Doppler) corrections have been used for this signal
20 + n·16	U1[4]	`reserved1`	-	-	Reserved
End of repeated group (`numSigs` times)					

3.16.12 UBX-NAV2-SLAS (0x29 0x42)

3.16.12.1 QZSS L1S SLAS status data

Message	**UBX-NAV2-SLAS** **QZSS L1S SLAS status data**						
Type	Periodic/polled						
Comment	This message outputs the status of the QZSS L1S SLAS sub system						
Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*		*Payload*	*Checksum*
	0xb5 0x62	0x29	0x42	20 + cnt·8		see below	CK_A CK_B

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	U4	`iTOW`	-	ms	GPS time of week of the navigation epoch. See the description of iTOW for details.
4	U1	`version`	-	-	Message version (0x00 for this version)
5	U1[3]	`reserved0`	-	-	Reserved
8	I4	`gmsLon`	1e-3	deg	Longitude of the used ground monitoring station
12	I4	`gmsLat`	1e-3	deg	Latitude of the used ground monitoring station
16	U1	`gmsCode`	-	-	Code of the used ground monitoring station according to the QZSS SLAS Interface Specification, available from qzss.go.jp/en/
17	U1	`qzssSvId`	-	-	Satellite identifier of the QZS/GEO whose correction data is used (see Satellite Numbering)
18	X1	`serviceFlags`	-	-	Flags regarding SLAS service
bit 0	U:1	`gmsAvailable`	-	-	1 = Ground monitoring station available
bit 1	U:1	`qzssSvAvailable`	-	-	1 = Correction providing QZSS SV available
bit 2	U:1	`testMode`	-	-	1 = Currently used QZSS SV in test mode
19	U1	`cnt`	-	-	Number of pseudorange corrections following
Start of repeated group (`cnt` times)					
20 + n·8	U1	`gnssId`	-	-	GNSS identifier (see Satellite Numbering)
21 + n·8	U1	`svId`	-	-	Satellite identifier (see Satellite Numbering)
22 + n·8	U1	`reserved1`	-	-	Reserved
23 + n·8	U1[3]	`reserved2`	-	-	Reserved
26 + n·8	I2	`prc`	-	cm	Pseudorange correction
End of repeated group (`cnt` times)					

3.16.13 UBX-NAV2-STATUS (0x29 0x03)

3.16.13.1 Receiver navigation status

Message	**UBX-NAV2-STATUS** **Receiver navigation status**						
Type	Periodic/polled						
Comment	See important comments concerning validity of position given in section Navigation output filters in the integration manual.						
Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*		*Payload*	*Checksum*
	0xb5 0x62	0x29	0x03	16		see below	CK_A CK_B

Payload description:

Byte offset	Type	Name	Scale	Unit	Description
0	U4	iTOW	-	ms	GPS time of week of the navigation epoch. See section iTOW timestamps in the integration manual for details.
4	U1	gpsFix	-	-	GPSfix Type, this value does **not** qualify a fix as valid and within the limits. See note on flag gpsFixOk below. • 0x00 = no fix • 0x01 = dead reckoning only • 0x02 = 2D-fix • 0x03 = 3D-fix • 0x04 = GPS + dead reckoning combined • 0x05 = Time only fix • 0x06..0xff = reserved
5	X1	flags	-	-	Navigation Status Flags
bit 0	U:1	gpsFixOk	-	-	1 = position and velocity valid and within DOP and ACC Masks.
bit 1	U:1	diffSoln	-	-	1 = differential corrections were applied
bit 2	U:1	wknSet	-	-	1 = Week Number valid (see section Time validity in the integration manual for details)
bit 3	U:1	towSet	-	-	1 = Time of Week valid (see section Time validity in the integration manual for details)
6	X1	fixStat	-	-	Fix Status Information
bit 0	U:1	diffCorr	-	-	1 = differential corrections available
bit 1	U:1	carrSolnValid	-	-	1 = valid carrSoln
bits 7…6	U:2	mapMatching	-	-	map matching status: • 00: none • 01: valid but not used, i.e. map matching data was received, but was too old • 10: valid and used, map matching data was applied • 11: valid and used, map matching data was applied. In case of sensor unavailability map matching data enables dead reckoning. This requires map matched latitude/longitude or heading data.
7	X1	flags2	-	-	further information about navigation output
bits 1…0	U:2	psmState	-	-	power save mode state (not supported for protocol versions less than 13.01) • 0 = ACQUISITION [or when psm disabled] • 1 = TRACKING • 2 = POWER OPTIMIZED TRACKING • 3 = INACTIVE
bits 4…3	U:2	spoofDetState	-	-	Spoofing detection state (not supported for protocol versions less than 18.00) • 0: Unknown or deactivated • 1: No spoofing indicated • 2: Spoofing indicated • 3: Multiple spoofing indications Note that the spoofing state value only reflects the detector state for the current navigation epoch. As spoofing can be detected most easily at the transition from real signal to spoofing signal, this is also where the detector is triggered the most. I.e. a value of *1 - No spoofing indicated* does not mean that the receiver is

					not spoofed, it simply states that the detector was not triggered in this epoch.
bits 7…6	U:2	carrSoln	-	-	Carrier phase range solution status: • 0 = no carrier phase range solution • 1 = carrier phase range solution with floating ambiguities • 2 = carrier phase range solution with fixed ambiguities
8	U4	ttff	-	ms	Time to first fix (millisecond time tag)
12	U4	msss	-	ms	Milliseconds since Startup / Reset

3.16.14 UBX-NAV2-SVIN (0x29 0x3b)

3.16.14.1 Survey-in data

Message	**UBX-NAV2-SVIN** **Survey-in data**				
Type	Periodic/polled				
Comment	This message contains information about survey-in parameters.				
Message structure	*Header*	*Class* *ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x29 0x3b	40	*see below*	CK_A CK_B

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	U1	version	-	-	Message version (0x00 for this version)
1	U1[3]	reserved0	-	-	Reserved
4	U4	iTOW	-	ms	GPS time of week of the navigation epoch. See the description of iTOW for details.
8	U4	dur	-	s	Passed survey-in observation time
12	I4	meanX	-	cm	Current survey-in mean position ECEF X coordinate
16	I4	meanY	-	cm	Current survey-in mean position ECEF Y coordinate
20	I4	meanZ	-	cm	Current survey-in mean position ECEF Z coordinate
24	I1	meanXHP	-	0.1_mm	Current high-precision survey-in mean position ECEF X coordinate. Must be in the range -99..+99. The current survey-in mean position ECEF X coordinate, in units of cm, is given by meanX + (0.01 * meanXHP)
25	I1	meanYHP	-	0.1_mm	Current high-precision survey-in mean position ECEF Y coordinate. Must be in the range -99..+99. The current survey-in mean position ECEF Y coordinate, in units of cm, is given by meanY + (0.01 * meanYHP)
26	I1	meanZHP	-	0.1_mm	Current high-precision survey-in mean position ECEF Z coordinate. Must be in the range -99..+99. The current survey-in mean position ECEF Z coordinate, in units of cm, is given by meanZ + (0.01 * meanZHP)
27	U1	reserved1	-	-	Reserved
28	U4	meanAcc	-	0.1_mm	Current survey-in mean position accuracy
32	U4	obs	-	-	Number of position observations used during survey-in

36	U1	`valid`	-	-	Survey-in position validity flag, 1 = valid, otherwise 0
37	U1	`active`	-	-	Survey-in in progress flag, 1 = in-progress, otherwise 0
38	U1[2]	`reserved2`	-	-	Reserved

3.16.15 UBX-NAV2-TIMEBDS (0x29 0x24)

3.16.15.1 BeiDou time solution

Message	**UBX-NAV2-TIMEBDS** **BeiDou time solution**					
Type	Periodic/polled					
Comment	This message reports the precise BDS time of the most recent navigation solution including validity flags and an accuracy estimate.					
Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x29	0x24	20	*see below*	CK_A CK_B

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	U4	`iTOW`	-	ms	GPS time of week of the navigation epoch. See section iTOW timestamps in the integration manual for details.
4	U4	`SOW`	-	s	BDS time of week (rounded to seconds)
8	I4	`fSOW`	-	ns	Fractional part of SOW (range: +/-500000000). The precise BDS time of week in seconds is: `SOW + fSOW * 1e-9`
12	I2	`week`	-	-	BDS week number of the navigation epoch
14	I1	`leapS`	-	s	BDS leap seconds (BDS-UTC)
15	X1	`valid`	-	-	Validity Flags
bit 0	U:1	`sowValid`	-	-	1 = Valid SOW and fSOW (see section Time validity in the integration manual for details)
bit 1	U:1	`weekValid`	-	-	1 = Valid week (see section Time validity in the integration manual for details)
bit 2	U:1	`leapSValid`	-	-	1 = Valid leap second
16	U4	`tAcc`	-	ns	Time Accuracy Estimate

3.16.16 UBX-NAV2-TIMEGAL (0x29 0x25)

3.16.16.1 Galileo time solution

Message	**UBX-NAV2-TIMEGAL** **Galileo time solution**					
Type	Periodic/polled					
Comment	This message reports the precise Galileo time of the most recent navigation solution including validity flags and an accuracy estimate.					
Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x29	0x25	20	*see below*	CK_A CK_B

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*

0	U4	`iTOW`	-	ms	GPS time of week of the navigation epoch. See section iTOW timestamps in the integration manual for details.
4	U4	`galTow`	-	s	Galileo time of week (rounded to seconds)
8	I4	`fGalTow`	-	ns	Fractional part of the Galileo time of week (range: +/-500000000). The precise Galileo time of week in seconds is: `galTow + fGalTow * 1e-9`
12	I2	`galWno`	-	-	Galileo week number
14	I1	`leapS`	-	s	Galileo leap seconds (Galileo-UTC)
15	X1	`valid`	-	-	Validity Flags
bit 0	U:1	`galTowValid`	-	-	1 = Valid galTow and fGalTow (see section Time validity in the integration manual for details)
bit 1	U:1	`galWnoValid`	-	-	1 = Valid galWno (see section Time validity in the integration manual for details)
bit 2	U:1	`leapSValid`	-	-	1 = Valid leapS
16	U4	`tAcc`	-	ns	Time Accuracy Estimate

3.16.17 UBX-NAV2-TIMEGLO (0x29 0x23)

3.16.17.1 GLONASS time solution

Message	**UBX-NAV2-TIMEGLO** **GLONASS time solution**					
Type	Periodic/polled					
Comment	This message reports the precise GLO time of the most recent navigation solution including validity flags and an accuracy estimate.					
Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x29	0x23	20	*see below*	CK_A CK_B

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	U4	`iTOW`	-	ms	GPS time of week of the navigation epoch. See section iTOW timestamps in the integration manual for details.
4	U4	`TOD`	-	s	GLONASS time of day (rounded to integer seconds)
8	I4	`fTOD`	-	ns	Fractional part of TOD (range: +/-500000000). The precise GLONASS time of day in seconds is: `TOD + fTOD * 1e-9`
12	U2	`Nt`	-	days	Current date (range: 1-1461), starting at 1 from the 1st Jan of the year indicated by N4 and ending at 1461 at the 31st Dec of the third year after that indicated by N4
14	U1	`N4`	-	-	Four-year interval number starting from 1996 (1=1996, 2=2000, 3=2004...)
15	X1	`valid`	-	-	Validity flags
bit 0	U:1	`todValid`	-	-	1 = Valid TOD and fTOD (see section Time validity in the integration manual for details)
bit 1	U:1	`dateValid`	-	-	1 = Valid N4 and Nt (see section Time validity in the integration manual for details)

16	U4	tAcc	-	ns	Time Accuracy Estimate

3.16.18 UBX-NAV2-TIMEGPS (0x29 0x20)

3.16.18.1 GPS time solution

Message	**UBX-NAV2-TIMEGPS** **GPS time solution**				
Type	Periodic/polled				
Comment	This message reports the precise GPS time of the most recent navigation solution including validity flags and an accuracy estimate.				
Message structure	*Header* 0xb5 0x62	*Class* 0x29	*ID* 0x20	*Length (Bytes)* 16	*Payload* *see below*	*Checksum* CK_A CK_B

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	U4	iTOW	-	ms	GPS time of week of the navigation epoch. See section iTOW timestamps in the integration manual for details.
4	I4	fTOW	-	ns	Fractional part of iTOW (range: +/-500000). The precise GPS time of week in seconds is: `(iTOW * 1e-3) + (fTOW * 1e-9)`
8	I2	week	-	-	GPS week number of the navigation epoch
10	I1	leapS	-	s	GPS leap seconds (GPS-UTC)
11	X1	valid	-	-	Validity Flags
bit 0	U:1	towValid	-	-	1 = Valid GPS time of week (iTOW & fTOW, (see section Time validity in the integration manual for details)
bit 1	U:1	weekValid	-	-	1 = Valid GPS week number (see section Time validity in the integration manual for details)
bit 2	U:1	leapSValid	-	-	1 = Valid GPS leap seconds
12	U4	tAcc	-	ns	Time Accuracy Estimate

3.16.19 UBX-NAV2-TIMELS (0x29 0x26)

3.16.19.1 Leap second event information

Message	**UBX-NAV2-TIMELS** **Leap second event information**				
Type	Periodic/polled				
Comment	Information about the upcoming leap second event if one is scheduled.				
Message structure	*Header* 0xb5 0x62	*Class* 0x29	*ID* 0x26	*Length (Bytes)* 24	*Payload* *see below*	*Checksum* CK_A CK_B

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	U4	iTOW	-	ms	GPS time of week of the navigation epoch. See section iTOW timestamps in the integration manual for details.
4	U1	version	-	-	Message version (0x00 for this version)
5	U1[3]	reserved0	-	-	Reserved

8	U1	`srcOfCurrLs`	-	-	Information source for the current number of leap seconds. • 0 = Default (hardcoded in the firmware, can be outdated) • 1 = Derived from time difference between GPS and GLONASS time • 2 = GPS • 3 = SBAS • 4 = BeiDou • 5 = Galileo • 6 = Aided data • 7 = Configured • 8 = NavIC • 255 = Unknown
9	I1	`currLs`	-	s	Current number of leap seconds since start of GPS time (Jan 6, 1980). It reflects how much GPS time is ahead of UTC time. Galileo number of leap seconds is the same as GPS. BeiDou number of leap seconds is 14 less than GPS. GLONASS follows UTC time, so no leap seconds.
10	U1	`srcOfLsChange`	-	-	Information source for the future leap second event. • 0 = No source • 2 = GPS • 3 = SBAS • 4 = BeiDou • 5 = Galileo • 6 = GLONASS • 7 = NavIC
11	I1	`lsChange`	-	s	Future leap second change if one is scheduled. +1 = positive leap second, -1 = negative leap second, 0 = no future leap second event scheduled or no information available.
12	I4	`timeToLsEvent`	-	s	Number of seconds until the next leap second event, or from the last leap second event if no future event scheduled. If > 0 event is in the future, = 0 event is now, < 0 event is in the past. Valid only if validTimeToLsEvent = 1.
16	U2	`dateOfLsGpsWn`	-	-	GPS week number (WN) of the next leap second event or the last one if no future event scheduled. Valid only if validTimeToLsEvent = 1.
18	U2	`dateOfLsGpsDn`	-	-	GPS day of week number (DN) for the next leap second event or the last one if no future event scheduled. Valid only if validTimeToLsEvent = 1. (GPS and Galileo DN: from 1 = Sun to 7 = Sat. BeiDou DN: from 0 = Sun to 6 = Sat.)
20	U1[3]	`reserved1`	-	-	Reserved
23	X1	`valid`	-	-	Validity flags
bit 0	U:1	`validCurrLs`	-	-	1 = Valid current number of leap seconds value.
bit 1	U:1	`validTimeToLsEvent`	-	-	1 = Valid time to next leap second event or from the last leap second event if no future event scheduled.

3.16.20 UBX-NAV2-TIMEQZSS (0x29 0x27)

3.16.20.1 QZSS time solution

Message	**UBX-NAV2-TIMEQZSS** **QZSS time solution**						
Type	Periodic/polled						
Comment	This message reports the precise QZSS time of the most recent navigation solution including validity flags and an accuracy estimate. See the Clocks and time section in the integration manual for details.						
Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*	
	0xb5 0x62	0x29	0x27	20	*see below*	CK_A CK_B	

Payload description:

Byte offset		*Type*	*Name*	*Scale*	*Unit*	*Description*
0		U4	`iTOW`	-	ms	GPS time of week of the navigation epoch.
4		U4	`qzssTow`	-	s	QZSS time of week (rounded to seconds)
8		I4	`fQzssTow`	-	ns	Fractional part of QZSS time of week (range: +/-500000000). The precise QZSS time of week in seconds is: `qzssTow + (fQzssTow * 1e-9)`
12		I2	`qzssWno`	-	-	QZSS week number of the navigation epoch
14		I1	`leapS`	-	s	QZSS leap seconds (QZSS-UTC)
15		X1	`valid`	-	-	Validity Flags
	bit 0	U:1	`qzssTowValid`	-	-	1 = Valid QZSS time of week (qzssTow and fQzssTow)
	bit 1	U:1	`qzssWnoValid`	-	-	1 = Valid QZSS week number
	bit 2	U:1	`leapSValid`	-	-	1 = Valid QZSS leap seconds
16		U4	`tAcc`	-	ns	Time Accuracy Estimate

3.16.21 UBX-NAV2-TIMEUTC (0x29 0x21)

3.16.21.1 UTC time solution

Message	**UBX-NAV2-TIMEUTC** **UTC time solution**					
Type	Periodic/polled					
Comment	Note that during a leap second there may be more or less than 60 seconds in a minute. See the description of leap seconds in the integration manual for details.					
Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x29	0x21	20	*see below*	CK_A CK_B

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	U4	`iTOW`	-	ms	GPS time of week of the navigation epoch. See section iTOW timestamps in the integration manual for details.
4	U4	`tAcc`	-	ns	Time accuracy estimate (UTC)
8	I4	`nano`	-	ns	Fraction of second, range -1e9 .. 1e9 (UTC)
12	U2	`year`	-	y	Year, range 1999..2099 (UTC)
14	U1	`month`	-	month	Month, range 1..12 (UTC)
15	U1	`day`	-	d	Day of month, range 1..31 (UTC)

16	U1	hour	-	h	Hour of day, range 0..23 (UTC)
17	U1	min	-	min	Minute of hour, range 0..59 (UTC)
18	U1	sec	-	s	Seconds of minute, range 0..60 (UTC)
19	X1	valid	-	-	Validity Flags
bit 0	U:1	validTOW	-	-	1 = Valid Time of Week (see section Time validity in the integration manual for details)
bit 1	U:1	validWKN	-	-	1 = Valid Week Number (see section Time validity in the integration manual for details)
bit 2	U:1	validUTC	-	-	1 = Valid UTC Time
bits 7...4	U:4	utcStandard	-	-	UTC standard identifier. (Not supported for protocol versions less than 15.00) • 0 = Information not available • 1 = Communications Research Labratory (CRL), Tokyo, Japan • 2 = National Institute of Standards and Technology (NIST) • 3 = U.S. Naval Observatory (USNO) • 4 = International Bureau of Weights and Measures (BIPM) • 5 = European laboratories • 6 = Former Soviet Union (SU) • 7 = National Time Service Center (NTSC), China • 8 = National Physics Laboratory India (NPLI) • 15 = Unknown

3.16.22 UBX-NAV2-VELECEF (0x29 0x11)

3.16.22.1 Velocity solution in ECEF

Message	**UBX-NAV2-VELECEF** **Velocity solution in ECEF**				
Type	Periodic/polled				
Comment	See important comments concerning validity of position given in section Navigation output filters in the integration manual.				
Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x29	0x11	20	*see below*	CK_A CK_B

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	U4	iTOW	-	ms	GPS time of week of the navigation epoch. See section iTOW timestamps in the integration manual for details.
4	I4	ecefVX	-	cm/s	ECEF X velocity
8	I4	ecefVY	-	cm/s	ECEF Y velocity
12	I4	ecefVZ	-	cm/s	ECEF Z velocity
16	U4	sAcc	-	cm/s	Speed accuracy estimate

3.16.23 UBX-NAV2-VELNED (0x29 0x12)

3.16.23.1 Velocity solution in NED frame

Message	**UBX-NAV2-VELNED**					
	Velocity solution in NED frame					
Type	Periodic/polled					
Comment	See important comments concerning validity of position given in section Navigation output filters in the integration manual.					
Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x29	0x12	36	*see below*	CK_A CK_B

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	U4	`iTOW`	-	ms	GPS time of week of the navigation epoch. See section iTOW timestamps in the integration manual for details.
4	I4	`velN`	-	cm/s	North velocity component
8	I4	`velE`	-	cm/s	East velocity component
12	I4	`velD`	-	cm/s	Down velocity component
16	U4	`speed`	-	cm/s	Speed (3-D)
20	U4	`gSpeed`	-	cm/s	Ground speed (2-D)
24	I4	`heading`	1e-5	deg	Heading of motion 2-D
28	U4	`sAcc`	-	cm/s	Speed accuracy Estimate
32	U4	`cAcc`	1e-5	deg	Course / Heading accuracy estimate

3.17 UBX-RXM (0x02)

The messages in the UBX-RXM class are used to output status and result data from the receiver manager as well as sending commands to the receiver manager.

3.17.1 UBX-RXM-COR (0x02 0x34)

3.17.1.1 Differential correction input status

Message	**UBX-RXM-COR**					
	Differential correction input status					
Type	Output					
Comment	This message shows information on received differential correction input messages. It is output upon successful parsing of a differential correction input message, irrespective of whether the parsed message is supported/used by the receiver.					
Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x02	0x34	12	*see below*	CK_A CK_B

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	U1	`version`	-	-	Message version (0x01 for this version)
1	U1	`ebno`	2^-3	dB	Energy per bit to noise power spectral density ratio (Eb/N0). 0: unknown. Reported only for protocol UBX-RXM-PMP (SPARTN) to monitor signal quality.
2	U1[2]	`reserved0`	-	-	Reserved
4	X4	`statusInfo`	-	-	Message input status information
bits 4…0	U:5	`protocol`	-	-	Input correction data protocol:

					• 0: Unknown • 1: RTCM3 • 2: SPARTN (Secure Position Augmentation for Real Time Navigation) • 29: UBX-RXM-PMP (SPARTN) • 30: UBX-RXM-QZSSL6
bits 6…5	U:2	`errStatus`	-	-	Error status of the received correction message content based on possibly available error codes or checksums: • 0: Unknown • 1: Error-free • 2: Erroneous
bits 8…7	U:2	`msgUsed`	-	-	Status of receiver using the input message: • 0: Unknown • 1: Not used • 2: Used
bits 24…9	U:16	`correctionId`	-	-	Identifier for the correction stream: • For RTCM 3: Reference station ID (DF003) of the received RTCM input message. Valid range 0-4095. Reported only for the standard RTCM messages that include the DF003 field and for the u-blox proprietary RTCM messages 4072.x. For all other messages, reports 0xFFFF. • For other correction protocols 0xFFFF.
bit 25	U:1	`msgTypeValid`	-	-	Validity of the msgType field. Set to False e.g. if the protocol does not define msgType.
bit 26	U:1	`msgSubType Valid`	-	-	Validity of the msgSubType field. Set to False e.g. if the protocol does not define subtype for the msgType.
bit 27	U:1	`msgInputHandle`	-	-	Input handling support of the input message: • 0: Receiver does not have input handling support for this message • 1: Receiver has input handling support for this message. Input handling support does not necessarily mean that message is supported/ used by the receiver.
bits 29…28	U:2	`msgEncrypted`	-	-	Encryption status of the input message: • 0: Unknown • 1: Not encrypted • 2: Encrypted
bits 31…30	U:2	`msgDecrypted`	-	-	Decryption status of the input message: • 0: Unknown • 1: Not decrypted • 2: Successfully decrypted
8	U2	`msgType`	-	-	Message type
10	U2	`msgSubType`	-	-	Message subtype

3.17.2 UBX-RXM-MEASX (0x02 0x14)

3.17.2.1 Satellite measurements for RRLP

Message	**UBX-RXM-MEASX** **Satellite measurements for RRLP**
Type	Periodic/polled
Comment	The message payload data is, where possible and appropriate, according to the Radio Resource LCS (Location Services) Protocol (RRLP) [1]. One exception is the satellite and GNSS IDs, which here are given according to

the Satellite Numbering scheme. The correct satellites have to be selected and their satellite ID translated accordingly [1, tab. A.10.14] for use in a RRLP Measure Position Response Component. Similarly, the measurement reference time of week has to be forwarded correctly (modulo 14400000 for the 24 LSB GPS measurements variant, modulo 3600000 for the 22 LSB Galileo and Additional Navigation Satelllite Systems (GANSS) measurements variant) of the RRLP measure position response to the SMLC.

Reference: [1] ETSI TS 144 031 V11.0.0 (2012-10), Digital cellular telecommunications system (Phase 2+), Location Services (LCS), Mobile Station (MS) - Serving Mobile Location Centre (SMLC), Radio Resource LCS Protocol (RRLP), (3GPP TS 44.031 version 11.0.0 Release 11).

Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x02	0x14	44 + numSV·24	*see below*	CK_A CK_B

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	U1	`version`	-	-	Message version, currently 0x01
1	U1[3]	`reserved0`	-	-	Reserved
4	U4	`gpsTOW`	-	ms	GPS measurement reference time
8	U4	`gloTOW`	-	ms	GLONASS measurement reference time
12	U4	`bdsTOW`	-	ms	BeiDou measurement reference time
16	U1[4]	`reserved1`	-	-	Reserved
20	U4	`qzssTOW`	-	ms	QZSS measurement reference time
24	U2	`gpsTOWacc`	2^-4	ms	GPS measurement reference time accuracy (0xffff = > 4s)
26	U2	`gloTOWacc`	2^-4	ms	GLONASS measurement reference time accuracy (0xffff = > 4s)
28	U2	`bdsTOWacc`	2^-4	ms	BeiDou measurement reference time accuracy (0xffff = > 4s)
30	U1[2]	`reserved2`	-	-	Reserved
32	U2	`qzssTOWacc`	2^-4	ms	QZSS measurement reference time accuracy (0xffff = > 4s)
34	U1	`numSV`	-	-	Number of satellites in repeated block
35	U1	`flags`	-	-	Flags
bits 1…0	U:2	`towSet`	-	-	TOW set (0 = no, 1 or 2 = yes)
36	U1[8]	`reserved3`	-	-	Reserved
Start of repeated group (`numSV` times)					
44 + n·24	U1	`gnssId`	-	-	GNSS ID (see Satellite Numbering)
45 + n·24	U1	`svId`	-	-	Satellite ID (see Satellite Numbering)
46 + n·24	U1	`cNo`	-	-	carrier noise ratio (0..63)
47 + n·24	U1	`mpathIndic`	-	-	multipath index (according to [1]) (0 = not measured, 1 = low, 2 = medium, 3 = high)
48 + n·24	I4	`dopplerMS`	0.04	m/s	Doppler measurement
52 + n·24	I4	`dopplerHz`	0.2	Hz	Doppler measurement
56 + n·24	U2	`wholeChips`	-	-	whole value of the code phase measurement (0..1022 for GPS)
58 + n·24	U2	`fracChips`	-	-	fractional value of the code phase measurement (0..1023)
60 + n·24	U4	`codePhase`	2^-21	ms	Code phase
64 + n·24	U1	`intCodePhase`	-	ms	Integer (part of the) code phase

65 + n·24	U1	`pseuRangeRMSErr`	-	-	pseudorange RMS error index (according to [1]) (0..63)
66 + n·24	U1[2]	`reserved4`	-	-	Reserved
End of repeated group (`numSV` times)					

3.17.3 UBX-RXM-PMP (0x02 0x72)

3.17.3.1 PMP (LBAND) message

Message	**UBX-RXM-PMP** **PMP (LBAND) message**					
Type	Input					
Comment	Point to Multipoint (LBAND) input message					
Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x02	0x72	24 + [0..n]	*see below*	CK_A CK_B

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	U1	`version`	-	-	Message version (0x01 for this version)
1	U1	`reserved0`	-	-	Reserved
2	U2	`numBytesUserData`	-	-	Number of bytes the userData block has in this frame (0...504)
4	U4	`timeTag`	-	ms	Time since startup when frame started - if max value of type is reached the counter will be reset
8	U4[2]	`uniqueWord`	-	-	Received unique words
16	U2	`serviceIdentifier`	-	-	Received service identifier
18	U1	`spare`	-	-	Received spare data
19	U1	`uniqueWordBitErrors`	-	-	Number of bit errors in both unique words
20	U2	`fecBits`	-	-	Number of bits corrected by FEC (forward error correction)
22	U1	`ebno`	2^-3	dB	Energy per bit to noise power spectral density ratio
23	U1	`reserved1`	-	-	Reserved
Start of repeated group (N times)					
24 + n	U1	`userData`	-	-	Received user data, which is variable (=numBytesUserData)
End of repeated group (N times)					

3.17.4 UBX-RXM-PMREQ (0x02 0x41)

3.17.4.1 Power management request

Message	**UBX-RXM-PMREQ** **Power management request**					
Type	Command					
Comment	This message requests a power management related task of the receiver.					
Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x02	0x41	8	*see below*	CK_A CK_B

Payload description:					
Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	U4	`duration`	-	ms	Duration of the requested task, set to zero for infinite duration. The maximum supported time is 12 days.
4	X4	`flags`	-	-	task flags
bit 1	U:1	`backup`	-	-	The receiver goes into backup mode for a time period defined by duration, provided that it is not connected to USB

3.17.4.2 Power management request

Message	**UBX-RXM-PMREQ** **Power management request**				
Type	Command				
Comment	This message requests a power management related task of the receiver.				
Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload* / *Checksum*
	0xb5 0x62	0x02	0x41	16	see below / CK_A CK_B

Payload description:					
Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	U1	`version`	-	-	Message version (0x00 for this version)
1	U1[3]	`reserved0`	-	-	Reserved
4	U4	`duration`	-	ms	Duration of the requested task, set to zero for infinite duration. The maximum supported time is 12 days.
8	X4	`flags`	-	-	task flags
bit 1	U:1	`backup`	-	-	The receiver goes into backup mode for a time period defined by duration, provided that it is not connected to USB
bit 2	U:1	`force`	-	-	Force receiver backup while USB is connected. USB interface will be disabled.
12	X4	`wakeupSources`	-	-	Configure pins to wake up the receiver. The receiver wakes up if there is either a falling or a rising edge on one of the configured pins.
bit 3	U:1	`uartrx`	-	-	Wake up the receiver if there is an edge on the UART RX pin
bit 5	U:1	`extint0`	-	-	Wake up the receiver if there is an edge on the EXTINT0 pin
bit 6	U:1	`extint1`	-	-	Wake up the receiver if there is an edge on the EXTINT1 pin
bit 7	U:1	`spics`	-	-	Wake up the receiver if there is an edge on the SPI CS pin

3.17.5 UBX-RXM-QZSSL6 (0x02 0x73)

3.17.5.1 QZSS L6 message

Message	**UBX-RXM-QZSSL6** **QZSS L6 message**
Type	Input
Comment	QZSS L6 message input, as defined in 'Quasi Zenith Satellite System Interface Specification Centimeter Level Augmentation Service (IS-QZSS-L6-001)'.

Message structure	Header	Class	ID	Length (Bytes)	Payload	Checksum
	0xb5 0x62	0x02	0x73	264	see below	CK_A CK_B

Payload description:

Byte offset	Type	Name	Scale	Unit	Description
0	U1	version	-	-	Message version (0x01 for this version)
1	U1	svId	-	-	Satellite identifier (see Satellite Numbering)
2	U2	cno	2^-8	dBHz	Mean C/N0
4	U4	timeTag	-	ms	Local time tag corresponding to the beginning of a received QZSS L6 message
8	U1	groupDelay	-	ns	L6 group delay w.r.t. L2 on channel
9	U1	bitErrCorr	-	-	Number of bit errors corrected by Reed-Solomon decoder
10	X2	chInfo	-	-	Information about receiver channel associated with a received QZSS L6 message
bits 9…8	U:2	chn	-	-	Receiver channel (0, 1)
bit 10	U:1	msgName	-	-	Message name, 0=L6D, 1=L6E
bits 13…12	U:2	errStatus	-	-	Error status of the received QZSS L6 message: 0=unknown, 1=error-free, 2=erroneous
bits 15…14	U:2	chName	-	-	Channel name, 0=channel A, 1=channel B
12	U1[2]	reserved0	-	-	Reserved
14	U1[250]	msgBytes	-	-	Bytes in a QZSS L6 message

3.17.6 UBX-RXM-RAWX (0x02 0x15)

3.17.6.1 Multi-GNSS raw measurements

Message	**UBX-RXM-RAWX** **Multi-GNSS raw measurements**
Type	Periodic/polled
Comment	This message contains the information needed to be able to generate a RINEX 3 multi-GNSS observation file (see ftp://ftp.igs.org/pub/data/format/). This message contains pseudorange, Doppler, carrier phase, phase lock and signal quality information for GNSS satellites once signals have been synchronized. This message supports all active GNSS. The only difference between this version of the message and the previous version (**UBX-RXM-RAWX-DATA0**) is the addition of the version field.

Message structure	Header	Class	ID	Length (Bytes)	Payload	Checksum
	0xb5 0x62	0x02	0x15	16 + numMeas·32	see below	CK_A CK_B

Payload description:

Byte offset	Type	Name	Scale	Unit	Description
0	R8	rcvTow	-	s	Measurement time of week in receiver local time approximately aligned to the GPS time system. The receiver local time of week, week number and leap second information can be used to translate the time to other time systems. More information about the difference in time systems can be found in the RINEX 3 format documentation. For a receiver operating in GLONASS only mode, UTC time can be determined by subtracting the leapS field from GPS time regardless of whether the GPS leap seconds are valid.
8	U2	week	-	weeks	GPS week number in receiver local time.

10	I1	leapS	-	s	GPS leap seconds (GPS-UTC). This field represents the receiver's best knowledge of the leap seconds offset. A flag is given in the recStat bitfield to indicate if the leap seconds are known.
11	U1	numMeas	-	-	Number of measurements to follow
12	X1	recStat	-	-	Receiver tracking status bitfield
bit 0	U:1	leapSec	-	-	Leap seconds have been determined
bit 1	U:1	clkReset	-	-	Clock reset applied. Typically the receiver clock is changed in increments of integer milliseconds.
13	U1	version	-	-	Message version (0x01 for this version)
14	U1[2]	reserved0	-	-	Reserved
Start of repeated group (numMeas times)					
16 + n·32	R8	prMes	-	m	Pseudorange measurement [m]. GLONASS inter frequency channel delays are compensated with an internal calibration table.
24 + n·32	R8	cpMes	-	cycles	Carrier phase measurement [cycles]. The carrier phase initial ambiguity is initialized using an approximate value to make the magnitude of the phase close to the pseudorange measurement. Clock resets are applied to both phase and code measurements in accordance with the RINEX specification.
32 + n·32	R4	doMes	-	Hz	Doppler measurement (positive sign for approaching satellites) [Hz]
36 + n·32	U1	gnssId	-	-	GNSS identifier (see Satellite Numbering for a list of identifiers)
37 + n·32	U1	svId	-	-	Satellite identifier (see Satellite Numbering)
38 + n·32	U1	sigId	-	-	New style signal identifier (see Signal Identifiers).(not supported for protocol versions less than 27.00)
39 + n·32	U1	freqId	-	-	Only used for GLONASS: This is the frequency slot + 7 (range from 0 to 13)
40 + n·32	U2	locktime	-	ms	Carrier phase locktime counter (maximum 64500ms)
42 + n·32	U1	cno	-	dBHz	Carrier-to-noise density ratio (signal strength) [dB-Hz]
43 + n·32	X1	prStdev	0.01*2^n	m	Estimated pseudorange measurement standard deviation
bits 3...0	U:4	prStd	-	-	Estimated pseudorange standard deviation
44 + n·32	X1	cpStdev	0.004	cycles	Estimated carrier phase measurement standard deviation (note a raw value of 0x0F indicates the value is invalid)
bits 3...0	U:4	cpStd	-	-	Estimated carrier phase standard deviation
45 + n·32	X1	doStdev	0.002*2^n	Hz	Estimated Doppler measurement standard deviation.
bits 3...0	U:4	doStd	-	-	Estimated Doppler standard deviation
46 + n·32	X1	trkStat	-	-	Tracking status bitfield
bit 0	U:1	prValid	-	-	Pseudorange valid
bit 1	U:1	cpValid	-	-	Carrier phase valid
bit 2	U:1	halfCyc	-	-	Half cycle valid
bit 3	U:1	subHalfCyc	-	-	Half cycle subtracted from phase
47 + n·32	U1	reserved1	-	-	Reserved

End of repeated group (numMeas times)

3.17.7 UBX-RXM-RLM (0x02 0x59)

3.17.7.1 Galileo SAR short-RLM report

Message	**UBX-RXM-RLM** **Galileo SAR short-RLM report**				
Type	Output				
Comment	This message contains the contents of any Galileo Search and Rescue (SAR) Short Return Link Message detected by the receiver.				
Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x02	0x59	16	*see below*	CK_A CK_B

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	U1	`version`	-	-	Message version (0x00 for this version)
1	U1	`type`	-	-	Message type (0x01 for Short-RLM)
2	U1	`svId`	-	-	Identifier of transmitting satellite (see Satellite Numbering)
3	U1	`reserved0`	-	-	Reserved
4	U1[8]	`beacon`	-	-	Beacon identifier (60 bits), with bytes ordered by earliest transmitted (most significant) first. Top four bits of first byte are zero.
12	U1	`message`	-	-	Message code (4 bits)
13	U1[2]	`params`	-	-	Parameters (16 bits), with bytes ordered by earliest transmitted (most significant) first.
15	U1	`reserved1`	-	-	Reserved

3.17.7.2 Galileo SAR long-RLM report

Message	**UBX-RXM-RLM** **Galileo SAR long-RLM report**				
Type	Output				
Comment	This message contains the contents of any Galileo Search and Rescue (SAR) Long Return Link Message detected by the receiver.				
Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x02	0x59	28	*see below*	CK_A CK_B

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	U1	`version`	-	-	Message version (0x00 for this version)
1	U1	`type`	-	-	Message type (0x02 for Long-RLM)
2	U1	`svId`	-	-	Identifier of transmitting satellite (see Satellite Numbering)
3	U1	`reserved0`	-	-	Reserved
4	U1[8]	`beacon`	-	-	Beacon identifier (60 bits), with bytes ordered by earliest transmitted (most significant) first. Top four bits of first byte are zero.
12	U1	`message`	-	-	Message code (4 bits)

13	U1[12]	params	-	-	Parameters (96 bits), with bytes ordered by earliest transmitted (most significant) first.
25	U1[3]	reserved1	-	-	Reserved

3.17.8 UBX-RXM-RTCM (0x02 0x32)

3.17.8.1 RTCM input status

Message	**UBX-RXM-RTCM** **RTCM input status**					
Type	Output					
Comment	This message shows info on a received RTCM input message. It is output upon successful parsing of an RTCM input message, irrespective of whether the RTCM message is supported or not by the receiver.					
Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x02	0x32	8	*see below*	CK_A CK_B

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	U1	version	-	-	Message version (0x02 for this version)
1	X1	flags	-	-	RTCM input status flags
bit 0	U:1	crcFailed	-	-	0 when RTCM message received and passed CRC check, 1 when failed, in which case refStation and msgType might be corrupted and misleading
bits 2…1	U:2	msgUsed	-	-	2 = RTCM message used successfully by the receiver, 1 = not used, 0 = do not know
2	U2	subType	-	-	Message subtype, only applicable to u-blox proprietary RTCM message 4072 (not available on all products)
4	U2	refStation	-	-	Reference station ID: • For RTCM 2.3: Reference station ID of the received RTCM 2 input message. Valid range 0-1023. • For RTCM 3.3: Reference station ID (DF003) of the received RTCM input message. Valid range 0-4095. Reported only for the standard RTCM messages that include the DF003 field and for the u-blox proprietary RTCM messages 4072.x. For all other messages, reports 0xFFFF.
6	U2	msgType	-	-	Message type

3.17.9 UBX-RXM-SFRBX (0x02 0x13)

3.17.9.1 Broadcast navigation data subframe

Message	**UBX-RXM-SFRBX** **Broadcast navigation data subframe**					
Type	Output					
Comment	This message reports a complete subframe of broadcast navigation data decoded from a single signal. The number of data words reported in each message depends on the nature of the signal.					
Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x02	0x13	8 + numWords·4	*see below*	CK_A CK_B

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	U1	gnssId	-	-	GNSS identifier (see Satellite Numbering)

1	U1	`svId`	-	-	Satellite identifier (see Satellite Numbering)
2	U1	`sigId`	-	-	Signal identifier (see Signal Identifiers)
3	U1	`freqId`	-	-	Only used for GLONASS: This is the frequency slot + 7 (range from 0 to 13)
4	U1	`numWords`	-	-	The number of data words contained in this message (up to 10, for currently supported signals)
5	U1	`chn`	-	-	The tracking channel number the message was received on
6	U1	`version`	-	-	Message version, (0x02 for this version)
7	U1	`reserved0`	-	-	Reserved
Start of repeated group (`numWords` times)					
8 + n·4	U4	`dwrd`	-	-	The data words
End of repeated group (`numWords` times)					

3.17.10 UBX-RXM-SPARTN (0x02 0x33)

3.17.10.1 SPARTN input status

Message	**UBX-RXM-SPARTN** **SPARTN input status**					
Type	Output					
Comment	This message shows info on a received SPARTN input message. It is output upon successful parsing of a SPARTN input message, irrespective of whether the SPARTN message is supported or not by the receiver.					
Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x02	0x33	8	*see below*	CK_A CK_B

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	U1	`version`	-	-	Message version (0x01 for this version)
1	X1	`flags`	-	-	SPARTN input status flags
bits 2…1	U:2	`msgUsed`	-	-	2 = SPARTN message used successfully by the receiver, 1 = not used, 0 = do not know
2	U2	`subType`	-	-	Message subtype
4	U1[2]	`reserved0`	-	-	Reserved
6	U2	`msgType`	-	-	Message type

3.17.11 UBX-RXM-SPARTNKEY (0x02 0x36)

3.17.11.1 Poll installed keys

Message	**UBX-RXM-SPARTNKEY** **Poll installed keys**					
Type	Poll request					
Comment	Depending on the number of active keys, the receiver shall send a UBX-RXM-SPARTNKEY message describing the keys. If there are no active keys then a UBX-RXM-SPARTNKEY shall be sent, with field numKeys set to zero.					
Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x02	0x36	0	*see below*	CK_A CK_B
Payload	This message has no payload.					

3.17.11.2 Transfer dynamic SPARTN keys

Message	**UBX-RXM-SPARTNKEY** **Transfer dynamic SPARTN keys**
Type	Input/output
Comment	This message is used to load keys to the receiver. The receiver has provision to store up to two (2) keys. By definition, the one currently used is named 'current' and the one that shall be used as soon as 'current' expires is named 'next'. Depending on how many active keys the receiver has at the time of receiving the message, one of the following shall occur: • If the receiver has no active keys, then the first key transferred shall become 'current'. If the message contains a second key, this shall become 'next'. • If the receiver has one (1) active key (current), the transferred key shall be stored as 'next'. If the message contains a second key, that key shall be dropped. • If the receiver has two (2) active keys (current and next), the transferred key(s) shall be dropped. The above implies that the correct procedure to load keys is the following: • When the receiver boots, the host should send 'current' and 'next' keys in one message. • Every time the 'current' key is expired, 'next' takes its place. Therefore the host should then retrieve the new 'next' key and send only that. To query the receiver's keys state (including the keys themselves), send a UBX-RXM-SPARTNKEY poll request.

Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x02	0x36	4 + numKeys·8 + [0..n]	*see below*	CK_A CK_B

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	U1	`version`	-	-	Message version (0x01 for this version)
1	U1	`numKeys`	-	-	Number of keys the message contains (can be 0, 1 or 2). In case of 0 the remaining fields will not be transmitted.
2	U1[2]	`reserved0`	-	-	Reserved
Start of repeated group (`numKeys` times)					
4 + n·8	U1	`reserved1`	-	-	Reserved
5 + n·8	U1	`keyLengthBytes`	-	-	Key length in bytes
6 + n·8	U2	`validFromWno`	-	week	GPS week number the key is valid from
8 + n·8	U4	`validFromTow`	-	sec	GPS time of week the key is valid from
End of repeated group (`numKeys` times)					
Start of repeated group (N times)					
4 + numKeys·8 + n	U1	`key`	-	-	Key(s) payload. This is a concatenation of all keys as raw bytes. The number of keys is defined in 'numKeys' field. Each key length is defined in its 'keyLengthBytes' field.
End of repeated group (N times)					

3.18 UBX-SEC (0x27)

The messages in the UBX-SEC class are used for security features of the receiver.

3.18.1 UBX-SEC-UNIQID (0x27 0x03)

3.18.1.1 Unique chip ID

Message	**UBX-SEC-UNIQID** **Unique chip ID**					
Type	Output					
Comment	This message is used to retrieve a unique chip identifier (40 bits, 5 bytes).					
Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x27	0x03	9	*see below*	CK_A CK_B

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	U1	`version`	-	-	Message version (0x01 for this version)
1	U1[3]	`reserved0`	-	-	Reserved
4	U1[5]	`uniqueId`	-	-	Unique chip ID

3.19 UBX-TIM (0x0d)

The messages in the UBX-TIM class are used to output timing information from the receiver, such as time pulse and time mark measurements.

3.19.1 UBX-TIM-TM2 (0x0d 0x03)

3.19.1.1 Time mark data

Message	**UBX-TIM-TM2** **Time mark data**					
Type	Periodic/polled					
Comment	This message contains information for high precision time stamping / pulse counting. The delay figures and timebase given in UBX-CFG-TP5 are also applied to the time results output in this message.					
Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x0d	0x03	28	*see below*	CK_A CK_B

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	U1	`ch`	-	-	Channel (i.e. EXTINT) upon which the pulse was measured
1	X1	`flags`	-	-	Bitmask
bit 0	U:1	`mode`	-	-	• 0=single • 1=running
bit 1	U:1	`run`	-	-	• 0=armed • 1=stopped
bit 2	U:1	`newFallingEdge`	-	-	New falling edge detected
bits 4…3	U:2	`timeBase`	-	-	• 0=Time base is Receiver time • 1=Time base is GNSS time (the system according to the configuration in UBX-CFG-TP5 for tpIdx=0) • 2=Time base is UTC (the variant according to the configuration in UBX-CFG-NAV5)
bit 5	U:1	`utc`	-	-	• 0=UTC not available • 1=UTC available
bit 6	U:1	`time`	-	-	• 0=Time is not valid • 1=Time is valid (Valid GNSS fix)

bit 7	U:1	newRisingEdge	-	-	New rising edge detected
2	U2	count	-	-	Rising edge counter
4	U2	wnR	-	-	Week number of last rising edge
6	U2	wnF	-	-	Week number of last falling edge
8	U4	towMsR	-	ms	Tow of rising edge
12	U4	towSubMsR	-	ns	Millisecond fraction of tow of rising edge in nanoseconds
16	U4	towMsF	-	ms	Tow of falling edge
20	U4	towSubMsF	-	ns	Millisecond fraction of tow of falling edge in nanoseconds
24	U4	accEst	-	ns	Accuracy estimate

3.19.2 UBX-TIM-TP (0x0d 0x01)

3.19.2.1 Time pulse time data

Message	**UBX-TIM-TP** **Time pulse time data**
Type	Periodic/polled
Comment	This message contains information on the timing of the next pulse at the TIMEPULSE0 output. The recommended configuration when using this message is to set both the measurement rate (CFG-RATE) and the timepulse frequency (CFG-TP) to 1 Hz.

Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x0d	0x01	16	*see below*	CK_A CK_B

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	U4	towMS	-	ms	Time pulse time of week according to time base
4	U4	towSubMS	2^-32	ms	Submillisecond part of towMS
8	I4	qErr	-	ps	Quantization error of time pulse
12	U2	week	-	weeks	Time pulse week number according to time base
14	X1	flags	-	-	Flags
bit 0	U:1	timeBase	-	-	• 0 = Time base is GNSS • 1 = Time base is UTC
bit 1	U:1	utc	-	-	• 0 = UTC not available • 1 = UTC available
bits 3…2	U:2	raim	-	-	(T)RAIM information • 0 = Information not available • 1 = Not active • 2 = Active
bit 4	U:1	qErrInvalid	-	-	• 0 = Quantization error valid • 1 = Quantization error invalid
15	X1	refInfo	-	-	Time reference information
bits 3…0	U:4	timeRefGnss	-	-	GNSS reference information. Only valid if time base is GNSS (timeBase=0). • 0 = GPS • 1 = GLONASS • 2 = BeiDou • 3 = Galileo • 4 = NavIC

					• 15 = Unknown
bits 7...4	U:4	utcStandard	-	-	UTC standard identifier. Only valid if time base is UTC (timeBase=1). • 0 = Information not available • 1 = Communications Research Laboratory (CRL), Tokyo, Japan • 2 = National Institute of Standards and Technology (NIST) • 3 = U.S. Naval Observatory (USNO) • 4 = International Bureau of Weights and Measures (BIPM) • 5 = European laboratories • 6 = Former Soviet Union (SU) • 7 = National Time Service Center (NTSC), China • 8 = National Physics Laboratory India (NPLI) • 15 = Unknown

3.19.3 UBX-TIM-VRFY (0x0d 0x06)

3.19.3.1 Sourced time verification

Message	**UBX-TIM-VRFY** **Sourced time verification**					
Type	Periodic/polled					
Comment	This message contains verification information about previous time received via assistance data or from RTC.					
Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x0d	0x06	20	*see below*	CK_A CK_B

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	I4	itow	-	ms	integer millisecond tow received by source
4	I4	frac	-	ns	sub-millisecond part of tow
8	I4	deltaMs	-	ms	integer milliseconds of delta time (current time minus sourced time)
12	I4	deltaNs	-	ns	Sub-millisecond part of delta time
16	U2	wno	-	week	Week number
18	X1	flags	-	-	Flags
bits 2...0	U:3	src	-	-	Aiding time source • 0 = no time aiding done • 2 = source was RTC • 3 = source was assistance data
19	U1	reserved0	-	-	Reserved

3.20 UBX-UPD (0x09)

The messages in the UBX-UPD class are used to download a firmware to the receiver and to update the firmware on the flash.

3.20.1 UBX-UPD-SOS (0x09 0x14)

3.20.1.1 Poll backup restore status

Message	**UBX-UPD-SOS** **Poll backup restore status**					
Type	Poll request					
Comment	Sending this (empty) message to the receiver results in the receiver returning a *System restored from backup* message as defined below.					
Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x09	0x14	0	*see below*	CK_A CK_B
Payload	This message has no payload.					

3.20.1.2 Create backup in flash

Message	**UBX-UPD-SOS** **Create backup in flash**					
Type	Command					
Comment	The host can send this message in order to save part of the battery-backed memory (BBR) in a file in the flash file system. The feature is designed in order to emulate the presence of the backup battery even if it is not present; the host can issue the save on shutdown command before switching off the device supply. It is recommended to issue a GNSS stop command using UBX-CFG-RST before in order to keep the BBR memory content consistent.					
Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x09	0x14	4	*see below*	CK_A CK_B

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	U1	`cmd`	-	-	Command (must be 0)
1	U1[3]	`reserved0`	-	-	Reserved

3.20.1.3 Clear backup in flash

Message	**UBX-UPD-SOS** **Clear backup in flash**					
Type	Command					
Comment	The host can send this message in order to erase the backup file present in flash. It is recommended that the clear operation is issued after the host has received the notification that the memory has been restored after a reset. Alternatively the host can parse the startup string *Restored data saved on shutdown* or poll the UBX-UPD-SOS message for obtaining the status.					
Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x09	0x14	4	*see below*	CK_A CK_B

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	U1	`cmd`	-	-	Command (must be 1)
1	U1[3]	`reserved0`	-	-	Reserved

3.20.1.4 Backup creation acknowledge

Message	**UBX-UPD-SOS** **Backup creation acknowledge**
Type	Output
Comment	The message is sent from the device as confirmation of creation of a backup file in flash. The host can safely shut down the device after having received this message.

Message structure	Header	Class	ID	Length (Bytes)	Payload	Checksum
	0xb5 0x62	0x09	0x14	8	see below	CK_A CK_B

Payload description:

Byte offset	Type	Name	Scale	Unit	Description
0	U1	`cmd`	-	-	Command (must be 2)
1	U1[3]	`reserved0`	-	-	Reserved
4	U1	`response`	-	-	• 0 = Not acknowledged • 1 = Acknowledged
5	U1[3]	`reserved1`	-	-	Reserved

3.20.1.5 System restored from backup

Message	**UBX-UPD-SOS** **System restored from backup**
Type	Output
Comment	The message is sent from the device to notify the host the BBR has been restored from a backup file in the flash file sysetem. The host should clear the backup file after receiving this message. If the UBX-UPD-SOS message is polled, this message will be resent.

Message structure	Header	Class	ID	Length (Bytes)	Payload	Checksum
	0xb5 0x62	0x09	0x14	8	see below	CK_A CK_B

Payload description:

Byte offset	Type	Name	Scale	Unit	Description
0	U1	`cmd`	-	-	Command (must be 3)
1	U1[3]	`reserved0`	-	-	Reserved
4	U1	`response`	-	-	• 0 = Unknown • 1 = Failed restoring from backup • 2 = Restored from backup • 3 = Not restored (no backup)
5	U1[3]	`reserved1`	-	-	Reserved

4 RTCM protocol

4.1 RTCM introduction

The RTCM (Radio Technical Commission for Maritime Services) protocols are used to supply the GNSS receiver with real-time differential correction data. The RTCM protocol specifications are available from http://www.rtcm.org.

The RTCM 3.x support is implemented according to *RTCM Standard 10403.3 Recommended Standards for Differential GNSS (Global Navigation Satellite Systems) Service, Version 3*.

4.2 RTCM 3.x configuration

The configuration of RTCM 3.x input or RTCM 3.x output (if available) is further detailed in the integration manual for typical applications.

The RTCM 3.x protocol can be disabled/enabled on communication interfaces using the Configuration interface, for example configuration item CFG-UART1INPROT-RTCM3X.

4.3 RTCM messages overview

Message	*Class/ID*	*Description (Type)*
RTCM-3X – RTCM 3.3 messages		
RTCM-3X-TYPE1001	0xf5 0x01	Message type 1001 • L1-only GPS RTK observables (Input)
RTCM-3X-TYPE1002	0xf5 0x02	Message type 1002 • Extended L1-only GPS RTK observables (Input)
RTCM-3X-TYPE1003	0xf5 0x03	Message type 1003 • L1/L2 GPS RTK observables (Input)
RTCM-3X-TYPE1004	0xf5 0x04	Message type 1004 • Extended L1/L2 GPS RTK observables (Input)
RTCM-3X-TYPE1005	0xf5 0x05	Message type 1005 • Stationary RTK reference station ARP (Input/output)
RTCM-3X-TYPE1006	0xf5 0x06	Message type 1006 • Stationary RTK reference station ARP with antenna height (Input)
RTCM-3X-TYPE1007	0xf5 0x07	Message type 1007 • Antenna descriptor (Input)
RTCM-3X-TYPE1009	0xf5 0x09	Message type 1009 • L1-only GLONASS RTK observables (Input)
RTCM-3X-TYPE1010	0xf5 0x0a	Message type 1010 • Extended L1-Only GLONASS RTK observables (Input)
RTCM-3X-TYPE1011	0xf5 0xa1	Message type 1011 • L1&L2 GLONASS RTK observables (Input)
RTCM-3X-TYPE1012	0xf5 0xa2	Message type 1012 • Extended L1&L2 GLONASS RTK observables (Input)
RTCM-3X-TYPE1033	0xf5 0x21	Message type 1033 • Receiver and antenna descriptors (Input)
RTCM-3X-TYPE1074	0xf5 0x4a	Message type 1074 • GPS MSM4 (Input/output)
RTCM-3X-TYPE1075	0xf5 0x4b	Message type 1075 • GPS MSM5 (Input)

Message	Class/ID	Description (Type)
RTCM-3X-TYPE1077	0xf5 0x4d	Message type 1077 • GPS MSM7 (Input/output)
RTCM-3X-TYPE1084	0xf5 0x54	Message type 1084 • GLONASS MSM4 (Input/output)
RTCM-3X-TYPE1085	0xf5 0x55	Message type 1085 • GLONASS MSM5 (Input)
RTCM-3X-TYPE1087	0xf5 0x57	Message type 1087 • GLONASS MSM7 (Input/output)
RTCM-3X-TYPE1094	0xf5 0x5e	Message type 1094 • Galileo MSM4 (Input/output)
RTCM-3X-TYPE1095	0xf5 0x5f	Message type 1095 • Galileo MSM5 (Input)
RTCM-3X-TYPE1097	0xf5 0x61	Message type 1097 • Galileo MSM7 (Input/output)
RTCM-3X-TYPE1124	0xf5 0x7c	Message type 1124 • BeiDou MSM4 (Input/output)
RTCM-3X-TYPE1125	0xf5 0x7d	Message type 1125 • BeiDou MSM5 (Input)
RTCM-3X-TYPE1127	0xf5 0x7f	Message type 1127 • BeiDou MSM7 (Input/output)
RTCM-3X-TYPE1230	0xf5 0xe6	Message type 1230 • GLONASS L1 and L2 code-phase biases (Input/output)
RTCM-3X-TYPE4072_0	0xf5 0xfe	Message type 4072, sub-type 0 • Reference station PVT (u-blox proprietary) (Input/output)
RTCM-3X-TYPE4072_1	0xf5 0xfd	Message type 4072, sub-type 1 • Additional reference station information (u-blox proprietary) (Output)

4.4 RTCM 3.3 messages

For details see RTCM protocol and the RTCM Standard 10403.3 Recommended Standards for Differential GNSS (Global Navigation Satellite Systems) Service, Version 3 available from http://www.rtcm.org.

4.4.1 Message type 1001

4.4.1.1 L1-only GPS RTK observables

Message	**RTCM-3X-TYPE1001** **L1-only GPS RTK observables**
Type	Input
Comment	See RTCM Standard 10403.3 Recommended Standards for Differential GNSS (Global Navigation Satellite Systems) Service, Version 3 for a detailed message specification.
Information	*Class/ID:* 0xf5 0x01, *Message Type:* 1001 (0x3e9), *Message Size:* 6 + nData

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	X1	`rtcmByte0`	-	-	RTCM frame byte 0
bits 7...0	U:8	`preamble`	-	-	Preamble (0xd3)
1	X1	`rtcmByte1`	-	-	RTCM frame byte 1
bits 1...0	U:2	`nDataMSB`	-	-	Payload length (2 MSB)

bits 7…2	U:6	`res1`	-	-	Reserved, all zero
2	X1	`rtcmByte2`	-	-	RTCM frame byte 2
bits 7…0	U:8	`nData`	-	-	Payload length (8 LSB)
Start of repeated group (`nData` times)					
3 + n	U1	`data`	-	-	Message payload data. Payload data length defined by combining nDataMSB and nData to form a 10-bit value.
End of repeated group (`nData` times)					
3 + nData	U1[3]	`crc`	-	-	Checksum

4.4.2 Message type 1002

4.4.2.1 Extended L1-only GPS RTK observables

Message	**RTCM-3X-TYPE1002** **Extended L1-only GPS RTK observables**
Type	Input
Comment	See RTCM Standard 10403.3 Recommended Standards for Differential GNSS (Global Navigation Satellite Systems) Service, Version 3 for a detailed message specification.
Information	*Class/ID:* 0xf5 0x02, *Message Type:* 1002 (0x3ea), *Message Size:* 6 + nData

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	X1	`rtcmByte0`	-	-	RTCM frame byte 0
bits 7…0	U:8	`preamble`	-	-	Preamble (0xd3)
1	X1	`rtcmByte1`	-	-	RTCM frame byte 1
bits 1…0	U:2	`nDataMSB`	-	-	Payload length (2 MSB)
bits 7…2	U:6	`res1`	-	-	Reserved, all zero
2	X1	`rtcmByte2`	-	-	RTCM frame byte 2
bits 7…0	U:8	`nData`	-	-	Payload length (8 LSB)
Start of repeated group (`nData` times)					
3 + n	U1	`data`	-	-	Message payload data. Payload data length defined by combining nDataMSB and nData to form a 10-bit value.
End of repeated group (`nData` times)					
3 + nData	U1[3]	`crc`	-	-	Checksum

4.4.3 Message type 1003

4.4.3.1 L1/L2 GPS RTK observables

Message	**RTCM-3X-TYPE1003** **L1/L2 GPS RTK observables**
Type	Input
Comment	See RTCM Standard 10403.3 Recommended Standards for Differential GNSS (Global Navigation Satellite Systems) Service, Version 3 for a detailed message specification.
Information	*Class/ID:* 0xf5 0x03, *Message Type:* 1003 (0x3eb), *Message Size:* 6 + nData

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*

0	X1	rtcmByte0	-	-	RTCM frame byte 0
bits 7...0	U:8	preamble	-	-	Preamble (0xd3)
1	X1	rtcmByte1	-	-	RTCM frame byte 1
bits 1...0	U:2	nDataMSB	-	-	Payload length (2 MSB)
bits 7...2	U:6	res1	-	-	Reserved, all zero
2	X1	rtcmByte2	-	-	RTCM frame byte 2
bits 7...0	U:8	nData	-	-	Payload length (8 LSB)
Start of repeated group (nData times)					
3 + n	U1	data	-	-	Message payload data. Payload data length defined by combining nDataMSB and nData to form a 10-bit value.
End of repeated group (nData times)					
3 + nData	U1[3]	crc	-	-	Checksum

4.4.4 Message type 1004

4.4.4.1 Extended L1/L2 GPS RTK observables

Message	**RTCM-3X-TYPE1004** **Extended L1/L2 GPS RTK observables**
Type	Input
Comment	See RTCM Standard 10403.3 Recommended Standards for Differential GNSS (Global Navigation Satellite Systems) Service, Version 3 for a detailed message specification.
Information	*Class/ID:* 0xf5 0x04, *Message Type:* 1004 (0x3ec), *Message Size:* 6 + nData

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	X1	rtcmByte0	-	-	RTCM frame byte 0
bits 7...0	U:8	preamble	-	-	Preamble (0xd3)
1	X1	rtcmByte1	-	-	RTCM frame byte 1
bits 1...0	U:2	nDataMSB	-	-	Payload length (2 MSB)
bits 7...2	U:6	res1	-	-	Reserved, all zero
2	X1	rtcmByte2	-	-	RTCM frame byte 2
bits 7...0	U:8	nData	-	-	Payload length (8 LSB)
Start of repeated group (nData times)					
3 + n	U1	data	-	-	Message payload data. Payload data length defined by combining nDataMSB and nData to form a 10-bit value.
End of repeated group (nData times)					
3 + nData	U1[3]	crc	-	-	Checksum

4.4.5 Message type 1005

4.4.5.1 Stationary RTK reference station ARP

Message	**RTCM-3X-TYPE1005** **Stationary RTK reference station ARP**
Type	Input/output

Comment	See RTCM Standard 10403.3 Recommended Standards for Differential GNSS (Global Navigation Satellite Systems) Service, Version 3 for a detailed message specification.				
Information	*Class/ID:* 0xf5 0x05, *Message Type:* 1005 (0x3ed), *Message Size:* 6 + nData				
Payload description:					
Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	X1	`rtcmByte0`	-	-	RTCM frame byte 0
bits 7...0	U:8	`preamble`	-	-	Preamble (0xd3)
1	X1	`rtcmByte1`	-	-	RTCM frame byte 1
bits 1...0	U:2	`nDataMSB`	-	-	Payload length (2 MSB)
bits 7...2	U:6	`res1`	-	-	Reserved, all zero
2	X1	`rtcmByte2`	-	-	RTCM frame byte 2
bits 7...0	U:8	`nData`	-	-	Payload length (8 LSB)
Start of repeated group (`nData` times)					
3 + n	U1	`data`	-	-	Message payload data. Payload data length defined by combining nDataMSB and nData to form a 10-bit value.
End of repeated group (`nData` times)					
3 + nData	U1[3]	`crc`	-	-	Checksum

4.4.6 Message type 1006

4.4.6.1 Stationary RTK reference station ARP with antenna height

Message	**RTCM-3X-TYPE1006** **Stationary RTK reference station ARP with antenna height**				
Type	Input				
Comment	See RTCM Standard 10403.3 Recommended Standards for Differential GNSS (Global Navigation Satellite Systems) Service, Version 3 for a detailed message specification.				
Information	*Class/ID:* 0xf5 0x06, *Message Type:* 1006 (0x3ee), *Message Size:* 6 + nData				
Payload description:					
Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	X1	`rtcmByte0`	-	-	RTCM frame byte 0
bits 7...0	U:8	`preamble`	-	-	Preamble (0xd3)
1	X1	`rtcmByte1`	-	-	RTCM frame byte 1
bits 1...0	U:2	`nDataMSB`	-	-	Payload length (2 MSB)
bits 7...2	U:6	`res1`	-	-	Reserved, all zero
2	X1	`rtcmByte2`	-	-	RTCM frame byte 2
bits 7...0	U:8	`nData`	-	-	Payload length (8 LSB)
Start of repeated group (`nData` times)					
3 + n	U1	`data`	-	-	Message payload data. Payload data length defined by combining nDataMSB and nData to form a 10-bit value.
End of repeated group (`nData` times)					
3 + nData	U1[3]	`crc`	-	-	Checksum

4.4.7 Message type 1007

4.4.7.1 Antenna descriptor

Message	**RTCM-3X-TYPE1007** **Antenna descriptor**				
Type	Input				
Comment	See RTCM Standard 10403.3 Recommended Standards for Differential GNSS (Global Navigation Satellite Systems) Service, Version 3 for a detailed message specification.				
Information	*Class/ID:* 0xf5 0x07, *Message Type:* 1007 (0x3ef), *Message Size:* 6 + nData				
Payload description:					
Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	X1	`rtcmByte0`	-	-	RTCM frame byte 0
bits 7…0	U:8	`preamble`	-	-	Preamble (0xd3)
1	X1	`rtcmByte1`	-	-	RTCM frame byte 1
bits 1…0	U:2	`nDataMSB`	-	-	Payload length (2 MSB)
bits 7…2	U:6	`res1`	-	-	Reserved, all zero
2	X1	`rtcmByte2`	-	-	RTCM frame byte 2
bits 7…0	U:8	`nData`	-	-	Payload length (8 LSB)
Start of repeated group (`nData` times)					
3 + n	U1	`data`	-	-	Message payload data. Payload data length defined by combining nDataMSB and nData to form a 10-bit value.
End of repeated group (`nData` times)					
3 + nData	U1[3]	`crc`	-	-	Checksum

4.4.8 Message type 1009

4.4.8.1 L1-only GLONASS RTK observables

Message	**RTCM-3X-TYPE1009** **L1-only GLONASS RTK observables**				
Type	Input				
Comment	See RTCM Standard 10403.3 Recommended Standards for Differential GNSS (Global Navigation Satellite Systems) Service, Version 3 for a detailed message specification.				
Information	*Class/ID:* 0xf5 0x09, *Message Type:* 1009 (0x3f1), *Message Size:* 6 + nData				
Payload description:					
Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	X1	`rtcmByte0`	-	-	RTCM frame byte 0
bits 7…0	U:8	`preamble`	-	-	Preamble (0xd3)
1	X1	`rtcmByte1`	-	-	RTCM frame byte 1
bits 1…0	U:2	`nDataMSB`	-	-	Payload length (2 MSB)
bits 7…2	U:6	`res1`	-	-	Reserved, all zero
2	X1	`rtcmByte2`	-	-	RTCM frame byte 2
bits 7…0	U:8	`nData`	-	-	Payload length (8 LSB)
Start of repeated group (`nData` times)					
3 + n	U1	`data`	-	-	Message payload data. Payload data length defined by combining nDataMSB and nData to form a 10-bit value.

End of repeated group (`nData` times)					
3 + nData	U1[3]	`crc`	-	-	Checksum

4.4.9 Message type 1010

4.4.9.1 Extended L1-Only GLONASS RTK observables

Message	**RTCM-3X-TYPE1010** **Extended L1-Only GLONASS RTK observables**
Type	Input
Comment	See RTCM Standard 10403.3 Recommended Standards for Differential GNSS (Global Navigation Satellite Systems) Service, Version 3 for a detailed message specification.
Information	*Class/ID:* 0xf5 0x0a, *Message Type:* 1010 (0x3f2), *Message Size:* 6 + nData

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	X1	`rtcmByte0`	-	-	RTCM frame byte 0
bits 7...0	U:8	`preamble`	-	-	Preamble (0xd3)
1	X1	`rtcmByte1`	-	-	RTCM frame byte 1
bits 1...0	U:2	`nDataMSB`	-	-	Payload length (2 MSB)
bits 7...2	U:6	`res1`	-	-	Reserved, all zero
2	X1	`rtcmByte2`	-	-	RTCM frame byte 2
bits 7...0	U:8	`nData`	-	-	Payload length (8 LSB)
Start of repeated group (`nData` times)					
3 + n	U1	`data`	-	-	Message payload data. Payload data length defined by combining nDataMSB and nData to form a 10-bit value.
End of repeated group (`nData` times)					
3 + nData	U1[3]	`crc`	-	-	Checksum

4.4.10 Message type 1011

4.4.10.1 L1&L2 GLONASS RTK observables

Message	**RTCM-3X-TYPE1011** **L1&L2 GLONASS RTK observables**
Type	Input
Comment	See RTCM Standard 10403.3 Recommended Standards for Differential GNSS (Global Navigation Satellite Systems) Service, Version 3 for a detailed message specification.
Information	*Class/ID:* 0xf5 0xa1, *Message Type:* 1011 (0x3f3), *Message Size:* 6 + nData

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	X1	`rtcmByte0`	-	-	RTCM frame byte 0
bits 7...0	U:8	`preamble`	-	-	Preamble (0xd3)
1	X1	`rtcmByte1`	-	-	RTCM frame byte 1
bits 1...0	U:2	`nDataMSB`	-	-	Payload length (2 MSB)
bits 7...2	U:6	`res1`	-	-	Reserved, all zero
2	X1	`rtcmByte2`	-	-	RTCM frame byte 2

bits 7...0	U:8	`nData`	-	-	Payload length (8 LSB)
Start of repeated group (`nData` times)					
3 + n	U1	`data`	-	-	Message payload data. Payload data length defined by combining nDataMSB and nData to form a 10-bit value.
End of repeated group (`nData` times)					
3 + nData	U1[3]	`crc`	-	-	Checksum

4.4.11 Message type 1012

4.4.11.1 Extended L1&L2 GLONASS RTK observables

Message	**RTCM-3X-TYPE1012** **Extended L1&L2 GLONASS RTK observables**
Type	Input
Comment	See RTCM Standard 10403.3 Recommended Standards for Differential GNSS (Global Navigation Satellite Systems) Service, Version 3 for a detailed message specification.
Information	*Class/ID:* 0xf5 0xa2, *Message Type:* 1012 (0x3f4), *Message Size:* 6 + nData

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	X1	`rtcmByte0`	-	-	RTCM frame byte 0
bits 7...0	U:8	`preamble`	-	-	Preamble (0xd3)
1	X1	`rtcmByte1`	-	-	RTCM frame byte 1
bits 1...0	U:2	`nDataMSB`	-	-	Payload length (2 MSB)
bits 7...2	U:6	`res1`	-	-	Reserved, all zero
2	X1	`rtcmByte2`	-	-	RTCM frame byte 2
bits 7...0	U:8	`nData`	-	-	Payload length (8 LSB)
Start of repeated group (`nData` times)					
3 + n	U1	`data`	-	-	Message payload data. Payload data length defined by combining nDataMSB and nData to form a 10-bit value.
End of repeated group (`nData` times)					
3 + nData	U1[3]	`crc`	-	-	Checksum

4.4.12 Message type 1033

4.4.12.1 Receiver and antenna descriptors

Message	**RTCM-3X-TYPE1033** **Receiver and antenna descriptors**
Type	Input
Comment	See RTCM Standard 10403.3 Recommended Standards for Differential GNSS (Global Navigation Satellite Systems) Service, Version 3 for a detailed message specification.
Information	*Class/ID:* 0xf5 0x21, *Message Type:* 1033 (0x409), *Message Size:* 6 + nData

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	X1	`rtcmByte0`	-	-	RTCM frame byte 0
bits 7...0	U:8	`preamble`	-	-	Preamble (0xd3)

1	X1	rtcmByte1	-	-	RTCM frame byte 1
bits 1...0	U:2	nDataMSB	-	-	Payload length (2 MSB)
bits 7...2	U:6	res1	-	-	Reserved, all zero
2	X1	rtcmByte2	-	-	RTCM frame byte 2
bits 7...0	U:8	nData	-	-	Payload length (8 LSB)
Start of repeated group (nData times)					
3 + n	U1	data	-	-	Message payload data. Payload data length defined by combining nDataMSB and nData to form a 10-bit value.
End of repeated group (nData times)					
3 + nData	U1[3]	crc	-	-	Checksum

4.4.13 Message type 1074

4.4.13.1 GPS MSM4

Message	**RTCM-3X-TYPE1074** **GPS MSM4**
Type	Input/output
Comment	Full GPS Pseudoranges and PhaseRanges plus CNR See RTCM Standard 10403.3 Recommended Standards for Differential GNSS (Global Navigation Satellite Systems) Service, Version 3 for a detailed message specification.
Information	*Class/ID:* 0xf5 0x4a, *Message Type:* 1074 (0x432), *Message Size:* 6 + nData

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	X1	rtcmByte0	-	-	RTCM frame byte 0
bits 7...0	U:8	preamble	-	-	Preamble (0xd3)
1	X1	rtcmByte1	-	-	RTCM frame byte 1
bits 1...0	U:2	nDataMSB	-	-	Payload length (2 MSB)
bits 7...2	U:6	res1	-	-	Reserved, all zero
2	X1	rtcmByte2	-	-	RTCM frame byte 2
bits 7...0	U:8	nData	-	-	Payload length (8 LSB)
Start of repeated group (nData times)					
3 + n	U1	data	-	-	Message payload data. Payload data length defined by combining nDataMSB and nData to form a 10-bit value.
End of repeated group (nData times)					
3 + nData	U1[3]	crc	-	-	Checksum

4.4.14 Message type 1075

4.4.14.1 GPS MSM5

Message	**RTCM-3X-TYPE1075** **GPS MSM5**
Type	Input
Comment	Full GPS Pseudoranges, PhaseRanges, PhaseRangeRate and CNR

	See RTCM Standard 10403.3 Recommended Standards for Differential GNSS (Global Navigation Satellite Systems) Service, Version 3 for a detailed message specification.
Information	*Class/ID:* 0xf5 0x4b, *Message Type:* 1075 (0x433), *Message Size:* 6 + nData

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	X1	`rtcmByte0`	-	-	RTCM frame byte 0
bits 7…0	U:8	`preamble`	-	-	Preamble (0xd3)
1	X1	`rtcmByte1`	-	-	RTCM frame byte 1
bits 1…0	U:2	`nDataMSB`	-	-	Payload length (2 MSB)
bits 7…2	U:6	`res1`	-	-	Reserved, all zero
2	X1	`rtcmByte2`	-	-	RTCM frame byte 2
bits 7…0	U:8	`nData`	-	-	Payload length (8 LSB)
Start of repeated group (`nData` times)					
3 + n	U1	`data`	-	-	Message payload data. Payload data length defined by combining nDataMSB and nData to form a 10-bit value.
End of repeated group (`nData` times)					
3 + nData	U1[3]	`crc`	-	-	Checksum

4.4.15 Message type 1077

4.4.15.1 GPS MSM7

Message	**RTCM-3X-TYPE1077** **GPS MSM7**
Type	Input/output
Comment	Full GPS Pseudoranges, PhaseRanges, PhaseRangeRate and CNR (high resolution) See RTCM Standard 10403.3 Recommended Standards for Differential GNSS (Global Navigation Satellite Systems) Service, Version 3 for a detailed message specification.
Information	*Class/ID:* 0xf5 0x4d, *Message Type:* 1077 (0x435), *Message Size:* 6 + nData

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	X1	`rtcmByte0`	-	-	RTCM frame byte 0
bits 7…0	U:8	`preamble`	-	-	Preamble (0xd3)
1	X1	`rtcmByte1`	-	-	RTCM frame byte 1
bits 1…0	U:2	`nDataMSB`	-	-	Payload length (2 MSB)
bits 7…2	U:6	`res1`	-	-	Reserved, all zero
2	X1	`rtcmByte2`	-	-	RTCM frame byte 2
bits 7…0	U:8	`nData`	-	-	Payload length (8 LSB)
Start of repeated group (`nData` times)					
3 + n	U1	`data`	-	-	Message payload data. Payload data length defined by combining nDataMSB and nData to form a 10-bit value.
End of repeated group (`nData` times)					

3 + nData	U1[3]	`crc`	-	-	Checksum

4.4.16 Message type 1084

4.4.16.1 GLONASS MSM4

Message	**RTCM-3X-TYPE1084** **GLONASS MSM4**
Type	Input/output
Comment	Full GLONASS Pseudoranges and PhaseRanges plus CNR See RTCM Standard 10403.3 Recommended Standards for Differential GNSS (Global Navigation Satellite Systems) Service, Version 3 for a detailed message specification.
Information	*Class/ID:* 0xf5 0x54, *Message Type:* 1084 (0x43c), *Message Size:* 6 + nData

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	X1	`rtcmByte0`	-	-	RTCM frame byte 0
bits 7...0	U:8	`preamble`	-	-	Preamble (0xd3)
1	X1	`rtcmByte1`	-	-	RTCM frame byte 1
bits 1...0	U:2	`nDataMSB`	-	-	Payload length (2 MSB)
bits 7...2	U:6	`res1`	-	-	Reserved, all zero
2	X1	`rtcmByte2`	-	-	RTCM frame byte 2
bits 7...0	U:8	`nData`	-	-	Payload length (8 LSB)
Start of repeated group (`nData` times)					
3 + n	U1	`data`	-	-	Message payload data. Payload data length defined by combining nDataMSB and nData to form a 10-bit value.
End of repeated group (`nData` times)					
3 + nData	U1[3]	`crc`	-	-	Checksum

4.4.17 Message type 1085

4.4.17.1 GLONASS MSM5

Message	**RTCM-3X-TYPE1085** **GLONASS MSM5**
Type	Input
Comment	Full GLONASS Pseudoranges, PhaseRanges, PhaseRangeRate and CNR See RTCM Standard 10403.3 Recommended Standards for Differential GNSS (Global Navigation Satellite Systems) Service, Version 3 for a detailed message specification.
Information	*Class/ID:* 0xf5 0x55, *Message Type:* 1085 (0x43d), *Message Size:* 6 + nData

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	X1	`rtcmByte0`	-	-	RTCM frame byte 0
bits 7...0	U:8	`preamble`	-	-	Preamble (0xd3)
1	X1	`rtcmByte1`	-	-	RTCM frame byte 1
bits 1...0	U:2	`nDataMSB`	-	-	Payload length (2 MSB)
bits 7...2	U:6	`res1`	-	-	Reserved, all zero

2	X1	`rtcmByte2`	-	-	RTCM frame byte 2
bits 7...0	U:8	`nData`	-	-	Payload length (8 LSB)
Start of repeated group (`nData` times)					
3 + n	U1	`data`	-	-	Message payload data. Payload data length defined by combining nDataMSB and nData to form a 10-bit value.
End of repeated group (`nData` times)					
3 + nData	U1[3]	`crc`	-	-	Checksum

4.4.18 Message type 1087

4.4.18.1 GLONASS MSM7

Message	**RTCM-3X-TYPE1087** **GLONASS MSM7**
Type	Input/output
Comment	Full GLONASS Pseudoranges, PhaseRanges, PhaseRangeRate and CNR (high resolution) See RTCM Standard 10403.3 Recommended Standards for Differential GNSS (Global Navigation Satellite Systems) Service, Version 3 for a detailed message specification.
Information	*Class/ID:* 0xf5 0x57, *Message Type:* 1087 (0x43f), *Message Size:* 6 + nData

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	X1	`rtcmByte0`	-	-	RTCM frame byte 0
bits 7...0	U:8	`preamble`	-	-	Preamble (0xd3)
1	X1	`rtcmByte1`	-	-	RTCM frame byte 1
bits 1...0	U:2	`nDataMSB`	-	-	Payload length (2 MSB)
bits 7...2	U:6	`res1`	-	-	Reserved, all zero
2	X1	`rtcmByte2`	-	-	RTCM frame byte 2
bits 7...0	U:8	`nData`	-	-	Payload length (8 LSB)
Start of repeated group (`nData` times)					
3 + n	U1	`data`	-	-	Message payload data. Payload data length defined by combining nDataMSB and nData to form a 10-bit value.
End of repeated group (`nData` times)					
3 + nData	U1[3]	`crc`	-	-	Checksum

4.4.19 Message type 1094

4.4.19.1 Galileo MSM4

Message	**RTCM-3X-TYPE1094** **Galileo MSM4**
Type	Input/output
Comment	Full Galileo Pseudoranges and PhaseRanges plus CNR See RTCM Standard 10403.3 Recommended Standards for Differential GNSS (Global Navigation Satellite Systems) Service, Version 3 for a detailed message specification.
Information	*Class/ID:* 0xf5 0x5e, *Message Type:* 1094 (0x446), *Message Size:* 6 + nData

Payload description:

Byte offset	Type	Name	Scale	Unit	Description
0	X1	rtcmByte0	-	-	RTCM frame byte 0
bits 7…0	U:8	preamble	-	-	Preamble (0xd3)
1	X1	rtcmByte1	-	-	RTCM frame byte 1
bits 1…0	U:2	nDataMSB	-	-	Payload length (2 MSB)
bits 7…2	U:6	res1	-	-	Reserved, all zero
2	X1	rtcmByte2	-	-	RTCM frame byte 2
bits 7…0	U:8	nData	-	-	Payload length (8 LSB)
Start of repeated group (`nData` times)					
3 + n	U1	data	-	-	Message payload data. Payload data length defined by combining nDataMSB and nData to form a 10-bit value.
End of repeated group (`nData` times)					
3 + nData	U1[3]	crc	-	-	Checksum

4.4.20 Message type 1095

4.4.20.1 Galileo MSM5

Message	**RTCM-3X-TYPE1095** **Galileo MSM5**
Type	Input
Comment	Full Galileo Pseudoranges, PhaseRanges, PhaseRangeRate and CNR See RTCM Standard 10403.3 Recommended Standards for Differential GNSS (Global Navigation Satellite Systems) Service, Version 3 for a detailed message specification.
Information	*Class/ID:* 0xf5 0x5f, *Message Type:* 1095 (0x447), *Message Size:* 6 + nData

Payload description:

Byte offset	Type	Name	Scale	Unit	Description
0	X1	rtcmByte0	-	-	RTCM frame byte 0
bits 7…0	U:8	preamble	-	-	Preamble (0xd3)
1	X1	rtcmByte1	-	-	RTCM frame byte 1
bits 1…0	U:2	nDataMSB	-	-	Payload length (2 MSB)
bits 7…2	U:6	res1	-	-	Reserved, all zero
2	X1	rtcmByte2	-	-	RTCM frame byte 2
bits 7…0	U:8	nData	-	-	Payload length (8 LSB)
Start of repeated group (`nData` times)					
3 + n	U1	data	-	-	Message payload data. Payload data length defined by combining nDataMSB and nData to form a 10-bit value.
End of repeated group (`nData` times)					
3 + nData	U1[3]	crc	-	-	Checksum

4.4.21 Message type 1097

4.4.21.1 Galileo MSM7

Message	**RTCM-3X-TYPE1097** **Galileo MSM7**				
Type	Input/output				
Comment	Full Galileo Pseudoranges, PhaseRanges, PhaseRangeRate and CNR (high resolution) See RTCM Standard 10403.3 Recommended Standards for Differential GNSS (Global Navigation Satellite Systems) Service, Version 3 for a detailed message specification.				
Information	*Class/ID:* 0xf5 0x61, *Message Type:* 1097 (0x449), *Message Size:* 6 + nData				
Payload description:					
Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	X1	`rtcmByte0`	-	-	RTCM frame byte 0
bits 7…0	U:8	`preamble`	-	-	Preamble (0xd3)
1	X1	`rtcmByte1`	-	-	RTCM frame byte 1
bits 1…0	U:2	`nDataMSB`	-	-	Payload length (2 MSB)
bits 7…2	U:6	`res1`	-	-	Reserved, all zero
2	X1	`rtcmByte2`	-	-	RTCM frame byte 2
bits 7…0	U:8	`nData`	-	-	Payload length (8 LSB)
Start of repeated group (`nData` times)					
3 + n	U1	`data`	-	-	Message payload data. Payload data length defined by combining nDataMSB and nData to form a 10-bit value.
End of repeated group (`nData` times)					
3 + nData	U1[3]	`crc`	-	-	Checksum

4.4.22 Message type 1124

4.4.22.1 BeiDou MSM4

Message	**RTCM-3X-TYPE1124** **BeiDou MSM4**				
Type	Input/output				
Comment	Full BeiDou Pseudoranges and PhaseRanges plus CNR See RTCM Standard 10403.3 Recommended Standards for Differential GNSS (Global Navigation Satellite Systems) Service, Version 3 for a detailed message specification.				
Information	*Class/ID:* 0xf5 0x7c, *Message Type:* 1124 (0x464), *Message Size:* 6 + nData				
Payload description:					
Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	X1	`rtcmByte0`	-	-	RTCM frame byte 0
bits 7…0	U:8	`preamble`	-	-	Preamble (0xd3)
1	X1	`rtcmByte1`	-	-	RTCM frame byte 1
bits 1…0	U:2	`nDataMSB`	-	-	Payload length (2 MSB)
bits 7…2	U:6	`res1`	-	-	Reserved, all zero
2	X1	`rtcmByte2`	-	-	RTCM frame byte 2
bits 7…0	U:8	`nData`	-	-	Payload length (8 LSB)
Start of repeated group (`nData` times)					

3 + n	U1	data	-	-	Message payload data. Payload data length defined by combining nDataMSB and nData to form a 10-bit value.
End of repeated group (nData times)					
3 + nData	U1[3]	crc	-	-	Checksum

4.4.23 Message type 1125

4.4.23.1 BeiDou MSM5

Message	**RTCM-3X-TYPE1125** **BeiDou MSM5**
Type	Input
Comment	Full BeiDou Pseudoranges, PhaseRanges, PhaseRangeRate and CNR See RTCM Standard 10403.3 Recommended Standards for Differential GNSS (Global Navigation Satellite Systems) Service, Version 3 for a detailed message specification.
Information	*Class/ID:* 0xf5 0x7d, *Message Type:* 1125 (0x465), *Message Size:* 6 + nData

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	X1	rtcmByte0	-	-	RTCM frame byte 0
bits 7…0	U:8	preamble	-	-	Preamble (0xd3)
1	X1	rtcmByte1	-	-	RTCM frame byte 1
bits 1…0	U:2	nDataMSB	-	-	Payload length (2 MSB)
bits 7…2	U:6	res1	-	-	Reserved, all zero
2	X1	rtcmByte2	-	-	RTCM frame byte 2
bits 7…0	U:8	nData	-	-	Payload length (8 LSB)
Start of repeated group (nData times)					
3 + n	U1	data	-	-	Message payload data. Payload data length defined by combining nDataMSB and nData to form a 10-bit value.
End of repeated group (nData times)					
3 + nData	U1[3]	crc	-	-	Checksum

4.4.24 Message type 1127

4.4.24.1 BeiDou MSM7

Message	**RTCM-3X-TYPE1127** **BeiDou MSM7**
Type	Input/output
Comment	Full BeiDou pseudoranges, PhaseRanges, PhaseRangeRate and CNR (high resolution) See RTCM Standard 10403.3 Recommended Standards for Differential GNSS (Global Navigation Satellite Systems) Service, Version 3 for a detailed message specification.
Information	*Class/ID:* 0xf5 0x7f, *Message Type:* 1127 (0x467), *Message Size:* 6 + nData

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	X1	rtcmByte0	-	-	RTCM frame byte 0
bits 7…0	U:8	preamble	-	-	Preamble (0xd3)

1	X1	`rtcmByte1`	-	-	RTCM frame byte 1
bits 1…0	U:2	`nDataMSB`	-	-	Payload length (2 MSB)
bits 7…2	U:6	`res1`	-	-	Reserved, all zero
2	X1	`rtcmByte2`	-	-	RTCM frame byte 2
bits 7…0	U:8	`nData`	-	-	Payload length (8 LSB)
Start of repeated group (`nData` times)					
3 + n	U1	`data`	-	-	Message payload data. Payload data length defined by combining nDataMSB and nData to form a 10-bit value.
End of repeated group (`nData` times)					
3 + nData	U1[3]	`crc`	-	-	Checksum

4.4.25 Message type 1230

4.4.25.1 GLONASS L1 and L2 code-phase biases

Message	**RTCM-3X-TYPE1230** **GLONASS L1 and L2 code-phase biases**				
Type	Input/output				
Comment	See RTCM Standard 10403.3 Recommended Standards for Differential GNSS (Global Navigation Satellite Systems) Service, Version 3 for a detailed message specification.				
Information	*Class/ID:* 0xf5 0xe6, *Message Type:* 1230 (0x4ce), *Message Size:* 6 + nData				
Payload description:					
Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	X1	`rtcmByte0`	-	-	RTCM frame byte 0
bits 7…0	U:8	`preamble`	-	-	Preamble (0xd3)
1	X1	`rtcmByte1`	-	-	RTCM frame byte 1
bits 1…0	U:2	`nDataMSB`	-	-	Payload length (2 MSB)
bits 7…2	U:6	`res1`	-	-	Reserved, all zero
2	X1	`rtcmByte2`	-	-	RTCM frame byte 2
bits 7…0	U:8	`nData`	-	-	Payload length (8 LSB)
Start of repeated group (`nData` times)					
3 + n	U1	`data`	-	-	Message payload data. Payload data length defined by combining nDataMSB and nData to form a 10-bit value.
End of repeated group (`nData` times)					
3 + nData	U1[3]	`crc`	-	-	Checksum

4.4.26 Message type 4072, sub-type 0

4.4.26.1 Reference station PVT (u-blox proprietary)

Message	**RTCM-3X-TYPE4072_0** **Reference station PVT (u-blox proprietary)**
Type	Input/output
Comment	The `payload` starts with the following RTCM data fields: • uint12 (12 bits unsigned, RTCM data field type D002): message type (0xfe8 for this message) • uint12 (12 bits unsigned, RTCM data field type D002): message sub-type (0x000 for this message)

Information	*Class/ID:* 0xf5 0xfe, *Message Type:* 4072 (0xfe8), *Sub-type:* 0 (0x000), *Message Size:* 6 + nData

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	X1	`rtcmByte0`	-	-	RTCM frame byte 0
bits 7...0	U:8	`preamble`	-	-	Preamble (0xd3)
1	X1	`rtcmByte1`	-	-	RTCM frame byte 1
bits 1...0	U:2	`nDataMSB`	-	-	Payload length (2 MSB)
bits 7...2	U:6	`res1`	-	-	Reserved, all zero
2	X1	`rtcmByte2`	-	-	RTCM frame byte 2
bits 7...0	U:8	`nData`	-	-	Payload length (8 LSB)
Start of repeated group (`nData` times)					
3 + n	U1	`data`	-	-	Message payload data. Payload data length defined by combining nDataMSB and nData to form a 10-bit value.
End of repeated group (`nData` times)					
3 + nData	U1[3]	`crc`	-	-	Checksum

4.4.27 Message type 4072, sub-type 1

4.4.27.1 Additional reference station information (u-blox proprietary)

Message	**RTCM-3X-TYPE4072_1** **Additional reference station information (u-blox proprietary)**
Type	Output
Comment	The `payload` starts with the following RTCM data fields: • uint12 (12 bits unsigned, RTCM data field type D002): message type (0xfe8 for this message) • uint12 (12 bits unsigned, RTCM data field type D002): message sub-type (0x001 for this message)
Information	*Class/ID:* 0xf5 0xfd, *Message Type:* 4072 (0xfe8), *Sub-type:* 1 (0x001), *Message Size:* 6 + nData

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	X1	`rtcmByte0`	-	-	RTCM frame byte 0
bits 7...0	U:8	`preamble`	-	-	Preamble (0xd3)
1	X1	`rtcmByte1`	-	-	RTCM frame byte 1
bits 1...0	U:2	`nDataMSB`	-	-	Payload length (2 MSB)
bits 7...2	U:6	`res1`	-	-	Reserved, all zero
2	X1	`rtcmByte2`	-	-	RTCM frame byte 2
bits 7...0	U:8	`nData`	-	-	Payload length (8 LSB)
Start of repeated group (`nData` times)					
3 + n	U1	`data`	-	-	Message payload data. Payload data length defined by combining nDataMSB and nData to form a 10-bit value.
End of repeated group (`nData` times)					
3 + nData	U1[3]	`crc`	-	-	Checksum

5 SPARTN protocol

5.1 SPARTN introduction

The SPARTN (Secure Position Augmentation for Real-Time Navigation) protocol are used to supply the GNSS receiver with real-time correction data. The SPARTN protocol specifications are available in spartnformat.org.

The SPARTN 2.0 support is implemented according to *Secure Position Augmentation for Real-Time Navigation (SPARTN) Interface Control Document, Version 2.0.1, September 2021*.

5.2 SPARTN configuration

The configuration of SPARTN input is further detailed in the integration manual for typical applications.

The SPARTN protocol can be disabled/enabled on communication interfaces using the Configuration interface, for example configuration item CFG-UART1INPROT-SPARTN.

5.3 SPARTN messages overview

Message	*Class/ID*	*Description (Type)*
SPARTN-1X – SPARTN messages		
SPARTN-1X-OCB_GPS	0xf6 0x01	Message type 0, sub-type 0 • GPS orbit, clock, bias (OCB) (Input)
SPARTN-1X-OCB_GLO	0xf6 0x02	Message type 0, sub-type 1 • GLONASS orbit, clock, bias (OCB) (Input)
SPARTN-1X-OCB_GAL	0xf6 0x03	Message type 0, sub-type 2 • Galileo orbit, clock, bias (OCB) (Input)
SPARTN-1X-HPAC_GPS	0xf6 0x0a	Message type 1, sub-type 0 • GPS high-precision atmosphere correction (HPAC) (Input)
SPARTN-1X-HPAC_GLO	0xf6 0x0b	Message type 1, sub-type 1 • GLONASS high-precision atmosphere correction (HPAC) (Input)
SPARTN-1X-HPAC_GAL	0xf6 0x0c	Message type 1, sub-type 2 • Galileo high-precision atmosphere correction (HPAC) (Input)
SPARTN-1X-GAD	0xf6 0x13	Message type 2, sub-type 0 • Geographic area definition (GAD) (Input)

5.4 SPARTN messages

For details see SPARTN protocol and the Secure Position Augmentation for Real-Time Navigation (SPARTN) Interface Control Document, Version 2.0.1, September 2021 available from https://www.sapcorda.com/.

5.4.1 Message type 0, sub-type 0

5.4.1.1 GPS orbit, clock, bias (OCB)

Message	**SPARTN-1X-OCB_GPS** **GPS orbit, clock, bias (OCB)**
Type	Input
Comment	This message carries the data for GPS satellite orbits, clocks, biases and other auxiliary information.

	See Secure Position Augmentation for Real-Time Navigation (SPARTN) Interface Control Document, Version 1.8.0, January 2020 or Secure Position Augmentation for Real-Time Navigation (SPARTN) Interface Control Document, Version 2.0.1, September 2021 for a detailed message specification.
Information	*Class/ID:* 0xf6 0x01, *Message Type:* 0 (0x00), *Sub-type:* 0 (0x0), *Message Size:* 5 + nData + crcType

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	X1	`spartnByte0`	-	-	SPARTN frame byte 0
bits 7…0	U:8	`preamble`	-	-	Preamble (0x73, 's')
1	X1	`spartnByte1`	-	-	SPARTN frame byte 1
bit 0	U:1	`nDataMSB`	-	-	Payload length (MSB)
bits 7…1	U:7	`msgType`	-	-	Message type
2	X1	`spartnByte2`	-	-	SPARTN frame byte 2
bits 7…0	U:8	`nData`	-	-	Payload length (middle 8 bits)
3	X1	`spartnByte3`	-	-	SPARTN frame byte 3
bits 3…0	U:4	`frameCrc`	-	-	Frame CRC
bits 5…4	U:2	`crcType`	-	-	Message CRC type
bit 6	U:1	`eaf`	-	-	Encryption and/or authentication flag
bit 7	U:1	`nDataLSB`	-	-	Payload length (LSB)
Start of repeated group (`nData` times)					
4 + n	U1	`data`	-	-	Message payload data. Payload data length defined by combining nDataMSB, nData and nDataLSB to form a 10-bit value.
End of repeated group (`nData` times)					
4 + nData	U1	`crc0`	-	-	Message CRC 1st byte
Start of repeated group (`crcType` times)					
5 + nData + n	U1	`crcN`	-	-	Message CRC additional bytes
End of repeated group (`crcType` times)					

5.4.2 Message type 0, sub-type 1

5.4.2.1 GLONASS orbit, clock, bias (OCB)

Message	**SPARTN-1X-OCB_GLO** **GLONASS orbit, clock, bias (OCB)**
Type	Input
Comment	This message carries the data for GLONASS satellite orbits, clocks, biases and other auxiliary information. See Secure Position Augmentation for Real-Time Navigation (SPARTN) Interface Control Document, Version 1.8.0, January 2020 or Secure Position Augmentation for Real-Time Navigation (SPARTN) Interface Control Document, Version 2.0.1, September 2021 for a detailed message specification.
Information	*Class/ID:* 0xf6 0x02, *Message Type:* 0 (0x00), *Sub-type:* 1 (0x1), *Message Size:* 5 + nData + crcType

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	X1	`spartnByte0`	-	-	SPARTN frame byte 0
bits 7…0	U:8	`preamble`	-	-	Preamble (0x73, 's')
1	X1	`spartnByte1`	-	-	SPARTN frame byte 1
bit 0	U:1	`nDataMSB`	-	-	Payload length (MSB)

bits 7...1	U:7	msgType	-	-	Message type
2	X1	spartnByte2	-	-	SPARTN frame byte 2
bits 7...0	U:8	nData	-	-	Payload length (middle 8 bits)
3	X1	spartnByte3	-	-	SPARTN frame byte 3
bits 3...0	U:4	frameCrc	-	-	Frame CRC
bits 5...4	U:2	crcType	-	-	Message CRC type
bit 6	U:1	eaf	-	-	Encryption and/or authentication flag
bit 7	U:1	nDataLSB	-	-	Payload length (LSB)
Start of repeated group (nData times)					
4 + n	U1	data	-	-	Message payload data. Payload data length defined by combining nDataMSB, nData and nDataLSB to form a 10-bit value.
End of repeated group (nData times)					
4 + nData	U1	crc0	-	-	Message CRC 1st byte
Start of repeated group (crcType times)					
5 + nData + n	U1	crcN	-	-	Message CRC additional bytes
End of repeated group (crcType times)					

5.4.3 Message type 0, sub-type 2

5.4.3.1 Galileo orbit, clock, bias (OCB)

Message	**SPARTN-1X-OCB_GAL** **Galileo orbit, clock, bias (OCB)**
Type	Input
Comment	This message carries the data for Galileo satellite orbits, clocks, biases and other auxiliary information. See Secure Position Augmentation for Real-Time Navigation (SPARTN) Interface Control Document, Version 1.8.0, January 2020 or Secure Position Augmentation for Real-Time Navigation (SPARTN) Interface Control Document, Version 2.0.1, September 2021 for a detailed message specification.
Information	*Class/ID:* 0xf6 0x03, *Message Type:* 0 (0x00), *Sub-type:* 2 (0x2), *Message Size:* 5 + nData + crcType

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	X1	spartnByte0	-	-	SPARTN frame byte 0
bits 7...0	U:8	preamble	-	-	Preamble (0x73, 's')
1	X1	spartnByte1	-	-	SPARTN frame byte 1
bit 0	U:1	nDataMSB	-	-	Payload length (MSB)
bits 7...1	U:7	msgType	-	-	Message type
2	X1	spartnByte2	-	-	SPARTN frame byte 2
bits 7...0	U:8	nData	-	-	Payload length (middle 8 bits)
3	X1	spartnByte3	-	-	SPARTN frame byte 3
bits 3...0	U:4	frameCrc	-	-	Frame CRC
bits 5...4	U:2	crcType	-	-	Message CRC type
bit 6	U:1	eaf	-	-	Encryption and/or authentication flag
bit 7	U:1	nDataLSB	-	-	Payload length (LSB)
Start of repeated group (nData times)					

4 + n	U1	`data`	-	-	Message payload data. Payload data length defined by combining nDataMSB, nData and nDataLSB to form a 10-bit value.
End of repeated group (`nData` times)					
4 + nData	U1	`crc0`	-	-	Message CRC 1st byte
Start of repeated group (`crcType` times)					
5 + nData + n	U1	`crcN`	-	-	Message CRC additional bytes
End of repeated group (`crcType` times)					

5.4.4 Message type 1, sub-type 0

5.4.4.1 GPS high-precision atmosphere correction (HPAC)

Message	**SPARTN-1X-HPAC_GPS** **GPS high-precision atmosphere correction (HPAC)**
Type	Input
Comment	This message contains high-precision atmosphere data for GPS, specifically ionospheric and tropospheric correction data. Both ionosphere and troposphere data are transmitted in the same message. See Secure Position Augmentation for Real-Time Navigation (SPARTN) Interface Control Document, Version 1.8.0, January 2020 or Secure Position Augmentation for Real-Time Navigation (SPARTN) Interface Control Document, Version 2.0.1, September 2021 for a detailed message specification.
Information	*Class/ID:* 0xf6 0x0a, *Message Type:* 1 (0x01), *Sub-type:* 0 (0x0), *Message Size:* 5 + nData + crcType

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	X1	`spartnByte0`	-	-	SPARTN frame byte 0
bits 7...0	U:8	`preamble`	-	-	Preamble (0x73, 's')
1	X1	`spartnByte1`	-	-	SPARTN frame byte 1
bit 0	U:1	`nDataMSB`	-	-	Payload length (MSB)
bits 7...1	U:7	`msgType`	-	-	Message type
2	X1	`spartnByte2`	-	-	SPARTN frame byte 2
bits 7...0	U:8	`nData`	-	-	Payload length (middle 8 bits)
3	X1	`spartnByte3`	-	-	SPARTN frame byte 3
bits 3...0	U:4	`frameCrc`	-	-	Frame CRC
bits 5...4	U:2	`crcType`	-	-	Message CRC type
bit 6	U:1	`eaf`	-	-	Encryption and/or authentication flag
bit 7	U:1	`nDataLSB`	-	-	Payload length (LSB)
Start of repeated group (`nData` times)					
4 + n	U1	`data`	-	-	Message payload data. Payload data length defined by combining nDataMSB, nData and nDataLSB to form a 10-bit value.
End of repeated group (`nData` times)					
4 + nData	U1	`crc0`	-	-	Message CRC 1st byte
Start of repeated group (`crcType` times)					
5 + nData + n	U1	`crcN`	-	-	Message CRC additional bytes

End of repeated group (crcType times)

5.4.5 Message type 1, sub-type 1

5.4.5.1 GLONASS high-precision atmosphere correction (HPAC)

Message	**SPARTN-1X-HPAC_GLO** **GLONASS high-precision atmosphere correction (HPAC)**
Type	Input
Comment	This message contains high-precision atmosphere data for GLONASS, specifically ionospheric and tropospheric correction data. Both ionosphere and troposphere data are transmitted in the same message. See Secure Position Augmentation for Real-Time Navigation (SPARTN) Interface Control Document, Version 1.8.0, January 2020 or Secure Position Augmentation for Real-Time Navigation (SPARTN) Interface Control Document, Version 2.0.1, September 2021 for a detailed message specification.
Information	*Class/ID:* 0xf6 0x0b, *Message Type:* 1 (0x01), *Sub-type:* 1 (0x1), *Message Size:* 5 + nData + crcType

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	X1	`spartnByte0`	-	-	SPARTN frame byte 0
bits 7…0	U:8	`preamble`	-	-	Preamble (0x73, 's')
1	X1	`spartnByte1`	-	-	SPARTN frame byte 1
bit 0	U:1	`nDataMSB`	-	-	Payload length (MSB)
bits 7…1	U:7	`msgType`	-	-	Message type
2	X1	`spartnByte2`	-	-	SPARTN frame byte 2
bits 7…0	U:8	`nData`	-	-	Payload length (middle 8 bits)
3	X1	`spartnByte3`	-	-	SPARTN frame byte 3
bits 3…0	U:4	`frameCrc`	-	-	Frame CRC
bits 5…4	U:2	`crcType`	-	-	Message CRC type
bit 6	U:1	`eaf`	-	-	Encryption and/or authentication flag
bit 7	U:1	`nDataLSB`	-	-	Payload length (LSB)
Start of repeated group (nData times)					
4 + n	U1	`data`	-	-	Message payload data. Payload data length defined by combining nDataMSB, nData and nDataLSB to form a 10-bit value.
End of repeated group (nData times)					
4 + nData	U1	`crc0`	-	-	Message CRC 1st byte
Start of repeated group (crcType times)					
5 + nData + n	U1	`crcN`	-	-	Message CRC additional bytes
End of repeated group (crcType times)					

5.4.6 Message type 1, sub-type 2

5.4.6.1 Galileo high-precision atmosphere correction (HPAC)

Message	**SPARTN-1X-HPAC_GAL** **Galileo high-precision atmosphere correction (HPAC)**
Type	Input
Comment	This message contains high-precision atmosphere data for Galileo, specifically ionospheric and tropospheric correction data. Both ionosphere and troposphere data are transmitted in the same message.

	See Secure Position Augmentation for Real-Time Navigation (SPARTN) Interface Control Document, Version 1.8.0, January 2020 or Secure Position Augmentation for Real-Time Navigation (SPARTN) Interface Control Document, Version 2.0.1, September 2021 for a detailed message specification.
Information	*Class/ID:* 0xf6 0x0c, *Message Type:* 1 (0x01), *Sub-type:* 2 (0x2), *Message Size:* 5 + nDataa + crcType

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	X1	`spartnByte0`	-	-	SPARTN frame byte 0
bits 7...0	U:8	`preamble`	-	-	Preamble (0x73, 's')
1	X1	`spartnByte1`	-	-	SPARTN frame byte 1
bit 0	U:1	`nDataMSB`	-	-	Payload length (MSB)
bits 7...1	U:7	`msgType`	-	-	Message type
2	X1	`spartnByte2`	-	-	SPARTN frame byte 2
bits 7...0	U:8	`nData`	-	-	Payload length (middle 8 bits)
3	X1	`spartnByte3`	-	-	SPARTN frame byte 3
bits 3...0	U:4	`frameCrc`	-	-	Frame CRC
bits 5...4	U:2	`crcType`	-	-	Message CRC type
bit 6	U:1	`eaf`	-	-	Encryption and/or authentication flag
bit 7	U:1	`nDataLSB`	-	-	Payload length (LSB)
Start of repeated group (`nData` times)					
4 + n	U1	`data`	-	-	Message payload data. Payload data length defined by combining nDataMSB, nData and nDataLSB to form a 10-bit value.
End of repeated group (`nData` times)					
4 + nData	U1	`crc0`	-	-	Message CRC 1st byte
Start of repeated group (`crcType` times)					
5 + nData + n	U1	`crcN`	-	-	Message CRC additional bytes
End of repeated group (`crcType` times)					

5.4.7 Message type 2, sub-type 0

5.4.7.1 Geographic area definition (GAD)

Message	**SPARTN-1X-GAD** **Geographic area definition (GAD)**
Type	Input
Comment	This message is used to define geographic areas of data usage. The use of this message can serve different purposes, including atmospheric data availability and other types of geographical/geometrical aspects of usage of data. See Secure Position Augmentation for Real-Time Navigation (SPARTN) Interface Control Document, Version 1.8.0, January 2020 or Secure Position Augmentation for Real-Time Navigation (SPARTN) Interface Control Document, Version 2.0.1, September 2021 for a detailed message specification.
Information	*Class/ID:* 0xf6 0x13, *Message Type:* 2 (0x02), *Sub-type:* 0 (0x0), *Message Size:* 5 + nData + crcType

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	X1	`spartnByte0`	-	-	SPARTN frame byte 0
bits 7...0	U:8	`preamble`	-	-	Preamble (0x73, 's')
1	X1	`spartnByte1`	-	-	SPARTN frame byte 1

bit 0	U:1	`nDataMSB`	-	-	Payload length (MSB)
bits 7…1	U:7	`msgType`	-	-	Message type
2	X1	`spartnByte2`	-	-	SPARTN frame byte 2
bits 7…0	U:8	`nData`	-	-	Payload length (middle 8 bits)
3	X1	`spartnByte3`	-	-	SPARTN frame byte 3
bits 3…0	U:4	`frameCrc`	-	-	Frame CRC
bits 5…4	U:2	`crcType`	-	-	Message CRC type
bit 6	U:1	`eaf`	-	-	Encryption and/or authentication flag
bit 7	U:1	`nDataLSB`	-	-	Payload length (LSB)
Start of repeated group (`nData` times)					
4 + n	U1	`data`	-	-	Message payload data. Payload data length defined by combining nDataMSB, nData and nDataLSB to form a 10-bit value.
End of repeated group (`nData` times)					
4 + nData	U1	`crc0`	-	-	Message CRC 1st byte
Start of repeated group (`crcType` times)					
5 + nData + n	U1	`crcN`	-	-	Message CRC additional bytes
End of repeated group (`crcType` times)					

6 Configuration interface

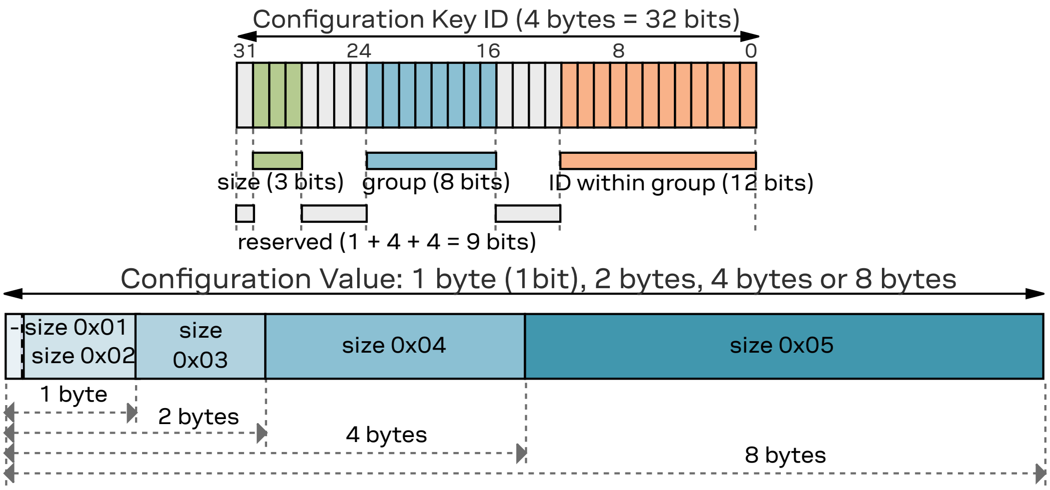
This chapter describes the receiver configuration interface.

6.1 Configuration database

The configuration database in the receiver's RAM holds the current configuration, which is used by the receiver at run-time. It is constructed on startup of the receiver from several sources of configuration. These sources are called *Configuration Layers*. The current configuration is called the *RAM Layer*. Any configuration in any layer is organized as *Configuration Items*, where each Configuration Item is referenced to by a unique *Configuration Key ID* and holds a single *Configuration Value*.

6.2 Configuration items

The following figure shows the structure of a *Configuration Item*, which consists of a *(Configuration) Key ID* and its *(Configuration) Value*:



A Configuration Key ID is a 32-bit integer value, which is split into the following parts:

- Bit 31: Currently unused. Reserved for future use.
- Bits 30…28: Three bits that indicate the storage size of a Configuration Value (range 0x01-0x05, see below)
- Bits 27…24: Currently unused. Reserved for future use.
- Bits 23…16: Eight bits that define a unique group ID (range 0x01-0xfe)
- Bits 15…12: Currently unused. Reserved for future use.
- Bits 11…0: Twelve bits that define a unique item ID within a group (range 0x001-0xffe)

The entire 32-bit value is the unique Key ID, which uniquely identifies a particular item. The numeric representation of the Key ID uses the lower-case hexadecimal format, such as `0x20c400a1`. An easier, more readable text representation uses the form CFG-*GROUP*-*ITEM*. This is also referred to as the *(Configuration) Key Name*.

Supported storage size identifiers (bits 30…28 of the Key ID) are:

- 0x01: one bit (the actual storage used is one byte, but only the least significant bit is used)
- 0x02: one byte
- 0x03: two bytes
- 0x04: four bytes

- 0x05: eight bytes

Each Configuration Item is of a certain type, which defines the interpretation of the raw binary data (see also UBX data types):

- U1, U2, U4, U8: unsigned little-endian integers of 8-, 16-, 32- and 64-bit widths
- I1, I2, I4, I8: signed little-endian, two's complement integers of 8-, 16-, 32- and 64-bit widths
- R4, R8: IEEE 754 single (32-bit) and double (64-bit) precision floats
- E1, E2, E4: unsigned little-endian enumeration of 8-, 16-, and 32-bit widths
- X1, X2, X4, X8: unsigned little-endian integers of 8-, 16-, 32- and 64-bit widths for bitfields and other binary data, such as strings
- L: single-bit boolean (true = 1, false = 0), stored as U1

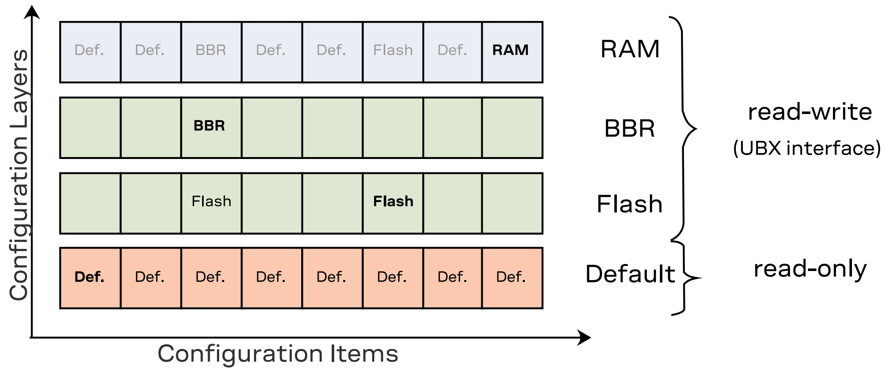
6.3 Configuration layers

Several *Configuration Layers* exist. They are separate sources of Configuration Items. Some of the layers are read-only and others are modifiable. Layers are organized in terms of priority. Values in a high-priority layer will replace values stored in low-priority layer. On startup of the receiver all configuration layers are read and the items within each layer are stacked up in order to create the *Current Configuration*, which is used by the receiver at run-time.

The following configuration layers are available (in order of priority, highest priority first):

- **RAM**: This layer contains items stored in volatile RAM. This is the Current Configuration. The value of any item can be set by the user at run-time (see UBX protocol interface) and it will become effective immediately.
- **BBR**: This layer contains items stored in the battery-backed RAM. The contents in this layer are preserved as long as a battery backup supply is provided during off periods. The value of any item can be set by the user at run-time (see UBX protocol interface) and it will become effective upon a restart of the receiver.
- **Flash**: This layer contains items stored permanently in the external flash memory. This layer is only available if there is a usable external flash memory. The value of any item can be set by the user at run-time (see UBX protocol interface) and it will become effective upon a restart of the receiver.
- **Default:** This layer contains all items known to the running receiver software and their hard-coded default values. Data in this layer is not writable.

The stacking of the configuration items from the different layers (sources) in order to construct the Current Configuration in the RAM Layer is depicted in the following figure. For each defined item, i.e. for each item in the Default Layer, the receiver software goes through the layers above and stacks all the found items on top. Some items may not be present in every layer. The result is the RAM Layer filled with all configuration items given Configuration Values coming from the highest priority layer the corresponding item was present. In the example figure below bold text indicates the source of the value in the Current Configuration (the RAM Layer). Empty boxes mean that the layer can hold the item but that it is not currently stored there. Boxes with text mean that an item is currently stored in the layer.



In the example figure above several items (e.g. the first item) are only set in the Default Layer and hence the default value ends up in Current Configuration in the RAM Layer. The third item is present in the Default, Flash and BBR Layers. The value from the BBR Layer has the highest priority and therefore it ends up in the RAM Layer. On the other hand, the default value of the sixth item is changed by the value in the Flash Layer. The value of the last item is changed in the RAM Layer only, i.e. upon startup the value in the RAM Layer was the value from the Default Layer, but the user has changed the value in the RAM Layer at run-time.

6.4 Configuration interface access

The following sections describe the existing interfaces to access the Configuration Database.

6.4.1 UBX protocol interface

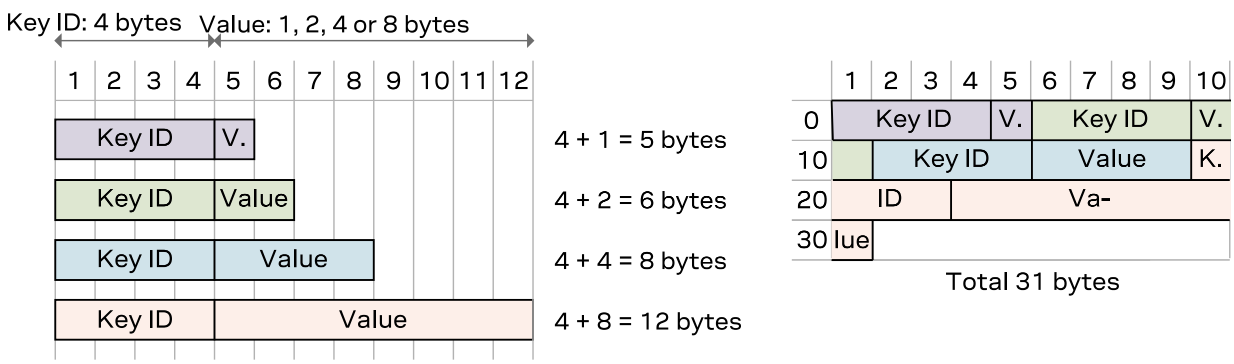
The following UBX protocol messages are available to access the Configuration Database:

- UBX-CFG-VALGET to read configuration items from the database
- UBX-CFG-VALSET to set configuration items in the database
- UBX-CFG-VALDEL to delete configuration items from the database

6.5 Configuration data

Configuration data is the binary representation of a list of Key ID and Value pairs. It is formed by concatenating keys (U4 values) and values (variable type) without any padding. This format is used in the UBX-CFG-VALSET and UBX-CFG-VALGET messages.

The figure below shows an example. The four Items (Key ID - Value pairs) on the left use the four fundamental storage sizes: one byte (L, U1, I1, E1 and X1 types), 2 bytes (U2, I2, E2 and X2 types), four byte (U4, I4, E4, X4 and R4 types) and eight bytes (U8, I8, X8 and R8 types). When concatenated (right) the Key IDs and Values are not aligned and there is no padding.



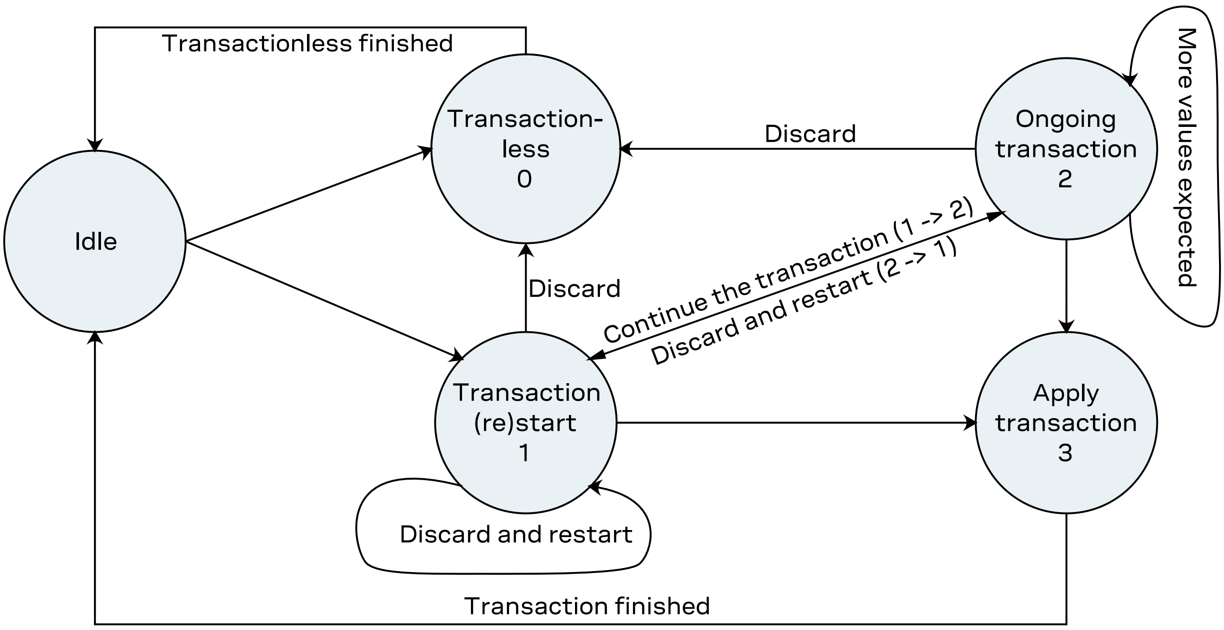
Note that this is an arbitrary example and any number of items of any value storage size can be concatenated the same way.

6.6 Configuration transactions

The configuration interface supports two mechanisms of configuration: the first is a transactionless mechanism where sent configuration changes are applied immediately to the configuration layer(s) requested. The second mechanism is a configuration transaction.

A transaction offers a way of queuing multiple configuration changes. It is particularly useful where different configuration keys depend on each other in such a way that sending one before the other can cause the configuration to be rejected. The queued configuration change requests are stored then checked collectively before being applied to the receiver.

A transaction can have the following states described in the figure below.



When starting a transaction, the user must specify the layer(s) the changes will be applied to. This list of configuration layer(s) must be observed throughout the transaction states. Modifying the configuration layer(s) mid-transaction will cause the transaction to be aborted and no queued changes will be applied.

In the start transaction state, the receiver will lock the configuration database so that changes from another entity or message cannot be applied. It is possible to send a configuration key-value pairs with the start transaction state. These will be queued waiting to be applied.

In the ongoing state, a configuration key and value must be sent. The receiver will abort the transaction and not apply any changes if this condition is violated. Key-value pairs sent in the ongoing state will be queued waiting to be applied.

In the apply state, the queued changes will be collectively checked and applied to the requested configuration layer(s). Note that any additional key-value pairs sent within the apply state will be ignored.

Note that a transaction can only come from a single source, a UBX-CFG-VALSET message or a UBX-CFG-VALDEL message. This means that in any given transaction it is not possible to mix a delete and a save request. Starting a transaction from a different source will abort the current transaction and no queued changes would be applied.

Refer to UBX-CFG-VALSET and UBX-CFG-VALDEL messages for a detailed description of how to set up a configuration transaction, its limitations and conditions that would cause the transaction to be rejected.

6.7 Configuration reset behavior

The RAM layer is always rebuilt from the layers below when the chip's processor comes out from reset. When using UBX-CFG-RST the processor goes through a reset cycle with these reset types (`resetMode` field):

- `0x00` hardware reset (watchdog) immediately
- `0x01` controlled software reset
- `0x04` hardware reset (watchdog) after shutdown

See section Forcing a receiver reset in the integration manual.

6.8 Configuration overview

Group	Description
CFG-BDS	BeiDou system configuration
CFG-GEOFENCE	Geofencing configuration
CFG-HW	Hardware configuration
CFG-I2C	Configuration of the I2C interface
CFG-I2CINPROT	Input protocol configuration of the I2C interface
CFG-I2COUTPROT	Output protocol configuration of the I2C interface
CFG-INFMSG	Information message configuration
CFG-ITFM	Jamming and interference monitor configuration
CFG-LOGFILTER	Data logger configuration
CFG-MOT	Motion detector configuration
CFG-MSGOUT	Message output configuration
CFG-NAV2	Secondary output configuration
CFG-NAVHPG	High precision navigation configuration
CFG-NAVSPG	Standard precision navigation configuration
CFG-NMEA	NMEA protocol configuration
CFG-ODO	Odometer and low-speed course over ground filter configuration
CFG-QZSS	QZSS system configuration
CFG-RATE	Navigation and measurement rate configuration
CFG-RINV	Remote inventory
CFG-RTCM	RTCM protocol configuration
CFG-SBAS	SBAS configuration
CFG-SEC	Security configuration
CFG-SIGNAL	Satellite systems (GNSS) signal configuration
CFG-SPARTN	SPARTN configuration
CFG-SPI	Configuration of the SPI interface

Group	Description
CFG-SPIINPROT	Input protocol configuration of the SPI interface
CFG-SPIOUTPROT	Output protocol configuration of the SPI interface
CFG-TMODE	Time mode configuration
CFG-TP	Timepulse configuration
CFG-TXREADY	TX ready configuration
CFG-UART1	Configuration of the UART1 interface
CFG-UART1INPROT	Input protocol configuration of the UART1 interface
CFG-UART1OUTPROT	Output protocol configuration of the UART1 interface
CFG-UART2	Configuration of the UART2 interface
CFG-UART2INPROT	Input protocol configuration of the UART2 interface
CFG-UART2OUTPROT	Output protocol configuration of the UART2 interface
CFG-USB	Configuration of the USB interface
CFG-USBINPROT	Input protocol configuration of the USB interface
CFG-USBOUTPROT	Output protocol configuration of the USB interface

6.9 Configuration reference

6.9.1 CFG-BDS: BeiDou system configuration

Note that enabling and disabling of individual GNSS is done via the CFG-SIGNAL configuration group.

Configuration item	Key ID	Type	Scale	Unit	Description
CFG-BDS-USE_GEO_PRN	`0x10340014`	L	-	-	Use BeiDou geostationary satellites (PRN 1-5 and 59-63)

Table 1: CFG-BDS configuration items

6.9.2 CFG-GEOFENCE: Geofencing configuration

Configuration for the geofencing feature. See section Geofencing in the integration manual for feature details.

If the receiver is sent a valid new configuration, it will respond with a UBX-ACK-ACK message and immediately change to the new configuration. Otherwise the receiver will reject the request, by issuing a UBX-ACK-NAK and continuing operation with the previous configuration.

Note that the acknowledge message does not indicate whether the PIO configuration has been successfully applied (pin assigned), it only indicates the successful configuration of the feature. The configured PIO must be previously unoccupied for successful assignment.

Configuration item	Key ID	Type	Scale	Unit	Description
CFG-GEOFENCE-CONFLVL	`0x20240011`	E1	-	-	Required confidence level for state evaluation
This value times the position's standard deviation (sigma) defines the confidence band. See Table 3 below for a list of possible constants for this item.					
CFG-GEOFENCE-USE_PIO	`0x10240012`	L	-	-	Use PIO combined fence state output
CFG-GEOFENCE-PINPOL	`0x20240013`	E1	-	-	PIO pin polarity
See Table 4 below for a list of possible constants for this item.					
CFG-GEOFENCE-PIN	`0x20240014`	U1	-	-	PIO pin number

Configuration item	Key ID	Type	Scale	Unit	Description
CFG-GEOFENCE-USE_FENCE1	0x10240020	L	-	-	Use first geofence
CFG-GEOFENCE-FENCE1_LAT	0x40240021	I4	1e-7	deg	Latitude of the first geofence circle center
CFG-GEOFENCE-FENCE1_LON	0x40240022	I4	1e-7	deg	Longitude of the first geofence circle center
CFG-GEOFENCE-FENCE1_RAD	0x40240023	U4	0.01	m	Radius of the first geofence circle
CFG-GEOFENCE-USE_FENCE2	0x10240030	L	-	-	Use second geofence
CFG-GEOFENCE-FENCE2_LAT	0x40240031	I4	1e-7	deg	Latitude of the second geofence circle center
CFG-GEOFENCE-FENCE2_LON	0x40240032	I4	1e-7	deg	Longitude of the second geofence circle center
CFG-GEOFENCE-FENCE2_RAD	0x40240033	U4	0.01	m	Radius of the second geofence circle
CFG-GEOFENCE-USE_FENCE3	0x10240040	L	-	-	Use third geofence
CFG-GEOFENCE-FENCE3_LAT	0x40240041	I4	1e-7	deg	Latitude of the third geofence circle center
CFG-GEOFENCE-FENCE3_LON	0x40240042	I4	1e-7	deg	Longitude of the third geofence circle center
CFG-GEOFENCE-FENCE3_RAD	0x40240043	U4	0.01	m	Radius of the third geofence circle
CFG-GEOFENCE-USE_FENCE4	0x10240050	L	-	-	Use fourth geofence
CFG-GEOFENCE-FENCE4_LAT	0x40240051	I4	1e-7	deg	Latitude of the fourth geofence circle center
CFG-GEOFENCE-FENCE4_LON	0x40240052	I4	1e-7	deg	Longitude of the fourth geofence circle center
CFG-GEOFENCE-FENCE4_RAD	0x40240053	U4	0.01	m	Radius of the fourth geofence circle

Table 2: CFG-GEOFENCE configuration items

Constant	Value	Description
L000	0	No confidence
L680	1	68%
L950	2	95%
L997	3	99.7%
L9999	4	99.99%
L999999	5	99.9999%

Table 3: Constants for CFG-GEOFENCE-CONFLVL

Constant	Value	Description
LOW_IN	0	PIO low means inside geofence
LOW_OUT	1	PIO low means outside geofence

Table 4: Constants for CFG-GEOFENCE-PINPOL

6.9.3 CFG-HW: Hardware configuration

Hardware configuration settings.

Configuration item	Key ID	Type	Scale	Unit	Description
CFG-HW-ANT_CFG_VOLTCTRL	0x10a3002e	L	-	-	Active antenna voltage control flag
Enable active antenna voltage control flag. Used by EXT and MADC engines.					
CFG-HW-ANT_CFG_SHORTDET	0x10a3002f	L	-	-	Short antenna detection flag
Enable short antenna detection flag. Used by EXT and MADC engines.					
CFG-HW-ANT_CFG_SHORTDET_POL	0x10a30030	L	-	-	Short antenna detection polarity
Set to true if polarity of the antenna short detection is active low. Used by EXT engine.					
CFG-HW-ANT_CFG_OPENDET	0x10a30031	L	-	-	Open antenna detection flag
Enable open antenna detection flag. Used by EXT and MADC engines.					

Configuration item	Key ID	Type	Scale	Unit	Description
CFG-HW-ANT_CFG_OPENDET_POL	0x10a30032	L	-	-	Open antenna detection polarity
Set to true if polarity of the antenna open detection is active low. Used by EXT engine.					
CFG-HW-ANT_CFG_PWRDOWN	0x10a30033	L	-	-	Power down antenna flag
Enable power down antenna logic in the event of antenna short circuit. CFG-HW-ANT_CFG_SHORTDET must be enabled to use this feature. Used by EXT and MADC engines.					
CFG-HW-ANT_CFG_PWRDOWN_POL	0x10a30034	L	-	-	Power down antenna logic polarity
Set to true if polarity of the antenna power down logic is active high. Used by EXT and MADC engines.					
CFG-HW-ANT_CFG_RECOVER	0x10a30035	L	-	-	Automatic recovery from short state flag
Enable automatic recovery from short state. Used by EXT and MADC engines.					
CFG-HW-ANT_SUP_SWITCH_PIN	0x20a30036	U1	-	-	ANT1 PIO number
Antenna Switch (ANT1) PIO number. Used by EXT and MADC engines.					
CFG-HW-ANT_SUP_SHORT_PIN	0x20a30037	U1	-	-	ANT0 PIO number
Antenna Short (ANT0) PIO number. Used by EXT engine.					
CFG-HW-ANT_SUP_OPEN_PIN	0x20a30038	U1	-	-	ANT2 PIO number
Antenna Switch (ANT2) PIO number. Used by EXT engine.					
CFG-HW-ANT_SUP_ENGINE	0x20a30054	E1	-	-	Antenna supervisor engine selection
Select the engine used to evaluate antenna state. See Table 6 below for a list of possible constants for this item.					
CFG-HW-ANT_SUP_SHORT_THR	0x20a30055	U1	-	mV	Antenna supervisor MADC engine short detection threshold
Threshold above which antenna short is detected. Used by MADC engine.					
CFG-HW-ANT_SUP_OPEN_THR	0x20a30056	U1	-	mV	Antenna supervisor MADC engine open detection threshold
Threshold below which antenna open/disconnected is detected. Used by MADC engine.					

Table 5: CFG-HW configuration items

Constant	Value	Description
EXT	0	Uses external comparators for current measurement.
MADC	1	Uses built-in ADC and a shunt for current measurement.

Table 6: Constants for CFG-HW-ANT_SUP_ENGINE

6.9.4 CFG-I2C: Configuration of the I2C interface

Settings needed to configure the I2C communication interface.

Configuration item	Key ID	Type	Scale	Unit	Description
CFG-I2C-ADDRESS	0x20510001	U1	-	-	I2C slave address of the receiver (7 bits)
CFG-I2C-EXTENDEDTIMEOUT	0x10510002	L	-	-	Flag to disable timeouting the interface after 1.5 s
CFG-I2C-ENABLED	0x10510003	L	-	-	Flag to indicate if the I2C interface should be enabled

Table 7: CFG-I2C configuration items

6.9.5 CFG-I2CINPROT: Input protocol configuration of the I2C interface

Input protocol enable flags of the I2C interface.

Configuration item	Key ID	Type	Scale	Unit	Description
CFG-I2CINPROT-UBX	0x10710001	L	-	-	Flag to indicate if UBX should be an input protocol on I2C

Configuration item	Key ID	Type	Scale	Unit	Description
CFG-I2CINPROT-NMEA	0x10710002	L	-	-	Flag to indicate if NMEA should be an input protocol on I2C
CFG-I2CINPROT-RTCM3X	0x10710004	L	-	-	Flag to indicate if RTCM3X should be an input protocol on I2C
CFG-I2CINPROT-SPARTN	0x10710005	L	-	-	Flag to indicate if SPARTN should be an input protocol on I2C

Table 8: CFG-I2CINPROT configuration items

6.9.6 CFG-I2COUTPROT: Output protocol configuration of the I2C interface

Output protocol enable flags of the I2C interface.

Configuration item	Key ID	Type	Scale	Unit	Description
CFG-I2COUTPROT-UBX	0x10720001	L	-	-	Flag to indicate if UBX should be an output protocol on I2C
CFG-I2COUTPROT-NMEA	0x10720002	L	-	-	Flag to indicate if NMEA should be an output protocol on I2C
CFG-I2COUTPROT-RTCM3X	0x10720004	L	-	-	Flag to indicate if RTCM3X should be an output protocol on I2C

Table 9: CFG-I2COUTPROT configuration items

6.9.7 CFG-INFMSG: Information message configuration

Information message configuration for the NMEA and UBX protocols.

Configuration item	Key ID	Type	Scale	Unit	Description
CFG-INFMSG-UBX_I2C	0x20920001	X1	-	-	Information message enable flags for the UBX protocol on the I2C interface
See Table 11 below for a list of possible constants for this item.					
CFG-INFMSG-UBX_UART1	0x20920002	X1	-	-	Information message enable flags for the UBX protocol on the UART1 interface
See Table 11 below for a list of possible constants for this item.					
CFG-INFMSG-UBX_UART2	0x20920003	X1	-	-	Information message enable flags for the UBX protocol on the UART2 interface
See Table 11 below for a list of possible constants for this item.					
CFG-INFMSG-UBX_USB	0x20920004	X1	-	-	Information message enable flags for the UBX protocol on the USB interface
See Table 11 below for a list of possible constants for this item.					
CFG-INFMSG-UBX_SPI	0x20920005	X1	-	-	Information message enable flags for the UBX protocol on the SPI interface
See Table 11 below for a list of possible constants for this item.					
CFG-INFMSG-NMEA_I2C	0x20920006	X1	-	-	Information message enable flags for the NMEA protocol on the I2C interface
See Table 11 below for a list of possible constants for this item.					
CFG-INFMSG-NMEA_UART1	0x20920007	X1	-	-	Information message enable flags for the NMEA protocol on the UART1 interface
See Table 11 below for a list of possible constants for this item.					
CFG-INFMSG-NMEA_UART2	0x20920008	X1	-	-	Information message enable flags for the NMEA protocol on the UART2 interface
See Table 11 below for a list of possible constants for this item.					
CFG-INFMSG-NMEA_USB	0x20920009	X1	-	-	Information message enable flags for the NMEA protocol on the USB interface
See Table 11 below for a list of possible constants for this item.					

Configuration item	Key ID	Type	Scale	Unit	Description
CFG-INFMSG-NMEA_SPI	`0x2092000a`	X1	-	-	Information message enable flags for the NMEA protocol on the SPI interface
See Table 11 below for a list of possible constants for this item.					

Table 10: CFG-INFMSG configuration items

Constant	Value	Description
ERROR	`0x01`	Enable ERROR information messages
WARNING	`0x02`	Enable WARNING information messages
NOTICE	`0x04`	Enable NOTICE information messages
TEST	`0x08`	Enable TEST information messages
DEBUG	`0x10`	Enable DEBUG information messages

Table 11: Constants for CFG-INFMSG-UBX_I2C, CFG-INFMSG-UBX_UART1, CFG-INFMSG-UBX_UART2, CFG-INFMSG-UBX_USB, CFG-INFMSG-UBX_SPI, CFG-INFMSG-NMEA_I2C, CFG-INFMSG-NMEA_UART1, CFG-INFMSG-NMEA_UART2, CFG-INFMSG-NMEA_USB, CFG-INFMSG-NMEA_SPI

6.9.8 CFG-ITFM: Jamming and interference monitor configuration

Configuration of jamming and interference monitor.

Configuration item	Key ID	Type	Scale	Unit	Description
CFG-ITFM-BBTHRESHOLD	`0x20410001`	U1	-	-	Broadband jamming detection threshold
CFG-ITFM-CWTHRESHOLD	`0x20410002`	U1	-	-	CW jamming detection threshold
CFG-ITFM-ENABLE	`0x1041000d`	L	-	-	Enable interference detection
CFG-ITFM-ANTSETTING	`0x20410010`	E1	-	-	Antenna setting
See Table 13 below for a list of possible constants for this item.					
CFG-ITFM-ENABLE_AUX	`0x10410013`	L	-	-	Scan auxiliary bands
Set to true to scan auxiliary bands. Supported on u-blox 8 / u-blox M8 only, otherwise ignored.					

Table 12: CFG-ITFM configuration items

Constant	Value	Description
UNKNOWN	`0`	Unknown
PASSIVE	`1`	Passive
ACTIVE	`2`	Active

Table 13: Constants for CFG-ITFM-ANTSETTING

6.9.9 CFG-LOGFILTER: Data logger configuration

This group can be used to configure the data logger, i.e. to enable/disable the log recording and to get/set the position entry filter settings.

Position entries can be filtered based on time difference, position difference or current speed thresholds. Position and speed filtering also have a minimum time interval. A position is logged if any of the thresholds are exceeded. If a threshold is set to zero it is ignored. The maximum rate of position logging is 1 Hz.

The filter settings will be configured to the provided values only if the APPLY_ALL_FILTERS flag is set. This allows the recording to be enabled/disabled independently of configuring the filter settings.

It is possible to configure the data logger in the absence of a logging file. By doing so, once the logging file is created, the data logger configuration will take effect immediately and logging recording and filtering will activate according to the configuration.

Configuration item	Key ID	Type	Scale	Unit	Description
CFG-LOGFILTER-RECORD_ENA	`0x10de0002`	L	-	-	Recording enabled
Set to true when recording enabled.					
CFG-LOGFILTER-ONCE_PER_WAKE_UP_ENA	`0x10de0003`	L	-	-	Once per wake up
Set to true recording only one single position per PSM on/off mode wake-up period is enabled. Note: the value set here does not take effect unless CFG-LOGFILTER-APPLY_ALL_FILTERS is enabled.					
CFG-LOGFILTER-APPLY_ALL_FILTERS	`0x10de0004`	L	-	-	Apply all filter settings
Set to true when all filter settings are to be applied, not just recording enabling/disabling.					
CFG-LOGFILTER-MIN_INTERVAL	`0x30de0005`	U2	-	s	Minimum time interval between logged positions
Minimum time interval between logged positions (0 = not set). **This is only applied in combination with the speed and/or position thresholds.** If both MIN_INTERVAL and TIME_THRS are set, MIN_INTERVAL must be less than or equal to TIME_THRS. Note: the value set here does not take effect unless CFG-LOGFILTER-APPLY_ALL_FILTERS is enabled.					
CFG-LOGFILTER-TIME_THRS	`0x30de0006`	U2	-	s	Time threshold
If the time difference is greater than the threshold then the position is logged (0 = not set). Note: the value set here does not take effect unless CFG-LOGFILTER-APPLY_ALL_FILTERS is enabled.					
CFG-LOGFILTER-SPEED_THRS	`0x30de0007`	U2	-	m/s	Speed threshold
If the current speed is greater than the threshold then the position is logged (0 = not set). MIN_INTERVAL also applies. Note: value set here does not take effect unless CFG-LOGFILTER-APPLY_ALL_FILTERS is enabled.					
CFG-LOGFILTER-POSITION_THRS	`0x40de0008`	U4	-	m	Position threshold
If the 3D position difference is greater than the threshold then the position is logged (0 = not set). MIN_INTERVAL also applies. Note: the value set here does not take effect unless CFG-LOGFILTER-APPLY_ALL_FILTERS is enabled.					

Table 14: CFG-LOGFILTER configuration items

6.9.10 CFG-MOT: Motion detector configuration

The items in this group specify the parameters used for the internal receiver motion detector. The platform motion is assessed by combining the detected motion of different detectors looking at specific data types (i.e. GNSS, gyroscopes, accelerometers, wheel ticks). The decision thresholds of the internal detectors can be specified using the configuration items in this group.

Configuration item	Key ID	Type	Scale	Unit	Description
CFG-MOT-GNSSSPEED_THRS	`0x20250038`	U1	0.01	m/s	GNSS speed threshold below which platform is considered as stationary (a.k.a. static hold threshold)
Set this parameter to 0 for firmware default value or behavior.					
CFG-MOT-GNSSDIST_THRS	`0x3025003b`	U2	-	-	Distance above which GNSS-based stationary motion is exit (a.k.a. static hold distance threshold)
Set this parameter to 0 for firmware default value or behavior.					

Table 15: CFG-MOT configuration items

6.9.11 CFG-MSGOUT: Message output configuration

For each message and port a separate output rate (per second, per epoch) can be configured.

Configuration item	Key ID	Type	Scale	Unit	Description
CFG-MSGOUT-NMEA_ID_DTM_I2C	0x209100a6	U1	-	-	Output rate of the NMEA-GX-DTM message on port I2C
CFG-MSGOUT-NMEA_ID_DTM_SPI	0x209100aa	U1	-	-	Output rate of the NMEA-GX-DTM message on port SPI
CFG-MSGOUT-NMEA_ID_DTM_UART1	0x209100a7	U1	-	-	Output rate of the NMEA-GX-DTM message on port UART1
CFG-MSGOUT-NMEA_ID_DTM_UART2	0x209100a8	U1	-	-	Output rate of the NMEA-GX-DTM message on port UART2
CFG-MSGOUT-NMEA_ID_DTM_USB	0x209100a9	U1	-	-	Output rate of the NMEA-GX-DTM message on port USB
CFG-MSGOUT-NMEA_ID_GBS_I2C	0x209100dd	U1	-	-	Output rate of the NMEA-GX-GBS message on port I2C
CFG-MSGOUT-NMEA_ID_GBS_SPI	0x209100e1	U1	-	-	Output rate of the NMEA-GX-GBS message on port SPI
CFG-MSGOUT-NMEA_ID_GBS_UART1	0x209100de	U1	-	-	Output rate of the NMEA-GX-GBS message on port UART1
CFG-MSGOUT-NMEA_ID_GBS_UART2	0x209100df	U1	-	-	Output rate of the NMEA-GX-GBS message on port UART2
CFG-MSGOUT-NMEA_ID_GBS_USB	0x209100e0	U1	-	-	Output rate of the NMEA-GX-GBS message on port USB
CFG-MSGOUT-NMEA_ID_GGA_I2C	0x209100ba	U1	-	-	Output rate of the NMEA-GX-GGA message on port I2C
CFG-MSGOUT-NMEA_ID_GGA_SPI	0x209100be	U1	-	-	Output rate of the NMEA-GX-GGA message on port SPI
CFG-MSGOUT-NMEA_ID_GGA_UART1	0x209100bb	U1	-	-	Output rate of the NMEA-GX-GGA message on port UART1
CFG-MSGOUT-NMEA_ID_GGA_UART2	0x209100bc	U1	-	-	Output rate of the NMEA-GX-GGA message on port UART2
CFG-MSGOUT-NMEA_ID_GGA_USB	0x209100bd	U1	-	-	Output rate of the NMEA-GX-GGA message on port USB
CFG-MSGOUT-NMEA_ID_GLL_I2C	0x209100c9	U1	-	-	Output rate of the NMEA-GX-GLL message on port I2C
CFG-MSGOUT-NMEA_ID_GLL_SPI	0x209100cd	U1	-	-	Output rate of the NMEA-GX-GLL message on port SPI
CFG-MSGOUT-NMEA_ID_GLL_UART1	0x209100ca	U1	-	-	Output rate of the NMEA-GX-GLL message on port UART1
CFG-MSGOUT-NMEA_ID_GLL_UART2	0x209100cb	U1	-	-	Output rate of the NMEA-GX-GLL message on port UART2
CFG-MSGOUT-NMEA_ID_GLL_USB	0x209100cc	U1	-	-	Output rate of the NMEA-GX-GLL message on port USB
CFG-MSGOUT-NMEA_ID_GNS_I2C	0x209100b5	U1	-	-	Output rate of the NMEA-GX-GNS message on port I2C
CFG-MSGOUT-NMEA_ID_GNS_SPI	0x209100b9	U1	-	-	Output rate of the NMEA-GX-GNS message on port SPI
CFG-MSGOUT-NMEA_ID_GNS_UART1	0x209100b6	U1	-	-	Output rate of the NMEA-GX-GNS message on port UART1
CFG-MSGOUT-NMEA_ID_GNS_UART2	0x209100b7	U1	-	-	Output rate of the NMEA-GX-GNS message on port UART2
CFG-MSGOUT-NMEA_ID_GNS_USB	0x209100b8	U1	-	-	Output rate of the NMEA-GX-GNS message on port USB
CFG-MSGOUT-NMEA_ID_GRS_I2C	0x209100ce	U1	-	-	Output rate of the NMEA-GX-GRS message on port I2C

Configuration item	Key ID	Type	Scale	Unit	Description
CFG-MSGOUT-NMEA_ID_GRS_SPI	0x209100d2	U1	-	-	Output rate of the NMEA-GX-GRS message on port SPI
CFG-MSGOUT-NMEA_ID_GRS_UART1	0x209100cf	U1	-	-	Output rate of the NMEA-GX-GRS message on port UART1
CFG-MSGOUT-NMEA_ID_GRS_UART2	0x209100d0	U1	-	-	Output rate of the NMEA-GX-GRS message on port UART2
CFG-MSGOUT-NMEA_ID_GRS_USB	0x209100d1	U1	-	-	Output rate of the NMEA-GX-GRS message on port USB
CFG-MSGOUT-NMEA_ID_GSA_I2C	0x209100bf	U1	-	-	Output rate of the NMEA-GX-GSA message on port I2C
CFG-MSGOUT-NMEA_ID_GSA_SPI	0x209100c3	U1	-	-	Output rate of the NMEA-GX-GSA message on port SPI
CFG-MSGOUT-NMEA_ID_GSA_UART1	0x209100c0	U1	-	-	Output rate of the NMEA-GX-GSA message on port UART1
CFG-MSGOUT-NMEA_ID_GSA_UART2	0x209100c1	U1	-	-	Output rate of the NMEA-GX-GSA message on port UART2
CFG-MSGOUT-NMEA_ID_GSA_USB	0x209100c2	U1	-	-	Output rate of the NMEA-GX-GSA message on port USB
CFG-MSGOUT-NMEA_ID_GST_I2C	0x209100d3	U1	-	-	Output rate of the NMEA-GX-GST message on port I2C
CFG-MSGOUT-NMEA_ID_GST_SPI	0x209100d7	U1	-	-	Output rate of the NMEA-GX-GST message on port SPI
CFG-MSGOUT-NMEA_ID_GST_UART1	0x209100d4	U1	-	-	Output rate of the NMEA-GX-GST message on port UART1
CFG-MSGOUT-NMEA_ID_GST_UART2	0x209100d5	U1	-	-	Output rate of the NMEA-GX-GST message on port UART2
CFG-MSGOUT-NMEA_ID_GST_USB	0x209100d6	U1	-	-	Output rate of the NMEA-GX-GST message on port USB
CFG-MSGOUT-NMEA_ID_GSV_I2C	0x209100c4	U1	-	-	Output rate of the NMEA-GX-GSV message on port I2C
CFG-MSGOUT-NMEA_ID_GSV_SPI	0x209100c8	U1	-	-	Output rate of the NMEA-GX-GSV message on port SPI
CFG-MSGOUT-NMEA_ID_GSV_UART1	0x209100c5	U1	-	-	Output rate of the NMEA-GX-GSV message on port UART1
CFG-MSGOUT-NMEA_ID_GSV_UART2	0x209100c6	U1	-	-	Output rate of the NMEA-GX-GSV message on port UART2
CFG-MSGOUT-NMEA_ID_GSV_USB	0x209100c7	U1	-	-	Output rate of the NMEA-GX-GSV message on port USB
CFG-MSGOUT-NMEA_ID_RLM_I2C	0x20910400	U1	-	-	Output rate of the NMEA-GX-RLM message on port I2C
CFG-MSGOUT-NMEA_ID_RLM_SPI	0x20910404	U1	-	-	Output rate of the NMEA-GX-RLM message on port SPI
CFG-MSGOUT-NMEA_ID_RLM_UART1	0x20910401	U1	-	-	Output rate of the NMEA-GX-RLM message on port UART1
CFG-MSGOUT-NMEA_ID_RLM_UART2	0x20910402	U1	-	-	Output rate of the NMEA-GX-RLM message on port UART2
CFG-MSGOUT-NMEA_ID_RLM_USB	0x20910403	U1	-	-	Output rate of the NMEA-GX-RLM message on port USB
CFG-MSGOUT-NMEA_ID_RMC_I2C	0x209100ab	U1	-	-	Output rate of the NMEA-GX-RMC message on port I2C
CFG-MSGOUT-NMEA_ID_RMC_SPI	0x209100af	U1	-	-	Output rate of the NMEA-GX-RMC message on port SPI

Configuration item	Key ID	Type	Scale	Unit	Description
CFG-MSGOUT-NMEA_ID_RMC_UART1	`0x209100ac`	U1	-	-	Output rate of the NMEA-GX-RMC message on port UART1
CFG-MSGOUT-NMEA_ID_RMC_UART2	`0x209100ad`	U1	-	-	Output rate of the NMEA-GX-RMC message on port UART2
CFG-MSGOUT-NMEA_ID_RMC_USB	`0x209100ae`	U1	-	-	Output rate of the NMEA-GX-RMC message on port USB
CFG-MSGOUT-NMEA_ID_VLW_I2C	`0x209100e7`	U1	-	-	Output rate of the NMEA-GX-VLW message on port I2C
CFG-MSGOUT-NMEA_ID_VLW_SPI	`0x209100eb`	U1	-	-	Output rate of the NMEA-GX-VLW message on port SPI
CFG-MSGOUT-NMEA_ID_VLW_UART1	`0x209100e8`	U1	-	-	Output rate of the NMEA-GX-VLW message on port UART1
CFG-MSGOUT-NMEA_ID_VLW_UART2	`0x209100e9`	U1	-	-	Output rate of the NMEA-GX-VLW message on port UART2
CFG-MSGOUT-NMEA_ID_VLW_USB	`0x209100ea`	U1	-	-	Output rate of the NMEA-GX-VLW message on port USB
CFG-MSGOUT-NMEA_ID_VTG_I2C	`0x209100b0`	U1	-	-	Output rate of the NMEA-GX-VTG message on port I2C
CFG-MSGOUT-NMEA_ID_VTG_SPI	`0x209100b4`	U1	-	-	Output rate of the NMEA-GX-VTG message on port SPI
CFG-MSGOUT-NMEA_ID_VTG_UART1	`0x209100b1`	U1	-	-	Output rate of the NMEA-GX-VTG message on port UART1
CFG-MSGOUT-NMEA_ID_VTG_UART2	`0x209100b2`	U1	-	-	Output rate of the NMEA-GX-VTG message on port UART2
CFG-MSGOUT-NMEA_ID_VTG_USB	`0x209100b3`	U1	-	-	Output rate of the NMEA-GX-VTG message on port USB
CFG-MSGOUT-NMEA_ID_ZDA_I2C	`0x209100d8`	U1	-	-	Output rate of the NMEA-GX-ZDA message on port I2C
CFG-MSGOUT-NMEA_ID_ZDA_SPI	`0x209100dc`	U1	-	-	Output rate of the NMEA-GX-ZDA message on port SPI
CFG-MSGOUT-NMEA_ID_ZDA_UART1	`0x209100d9`	U1	-	-	Output rate of the NMEA-GX-ZDA message on port UART1
CFG-MSGOUT-NMEA_ID_ZDA_UART2	`0x209100da`	U1	-	-	Output rate of the NMEA-GX-ZDA message on port UART2
CFG-MSGOUT-NMEA_ID_ZDA_USB	`0x209100db`	U1	-	-	Output rate of the NMEA-GX-ZDA message on port USB
CFG-MSGOUT-NMEA_NAV2_ID_GGA_I2C	`0x20910661`	U1	-	-	Output rate of the NMEA-NAV2-GX-GGA message on port I2C
CFG-MSGOUT-NMEA_NAV2_ID_GGA_SPI	`0x20910665`	U1	-	-	Output rate of the NMEA-NAV2-GX-GGA message on port SPI
CFG-MSGOUT-NMEA_NAV2_ID_GGA_UART1	`0x20910662`	U1	-	-	Output rate of the NMEA-NAV2-GX-GGA message on port UART1
CFG-MSGOUT-NMEA_NAV2_ID_GGA_UART2	`0x20910663`	U1	-	-	Output rate of the NMEA-NAV2-GX-GGA message on port UART2
CFG-MSGOUT-NMEA_NAV2_ID_GGA_USB	`0x20910664`	U1	-	-	Output rate of the NMEA-NAV2-GX-GGA message on port USB
CFG-MSGOUT-NMEA_NAV2_ID_GLL_I2C	`0x20910670`	U1	-	-	Output rate of the NMEA-NAV2-GX-GLL message on port I2C
CFG-MSGOUT-NMEA_NAV2_ID_GLL_SPI	`0x20910674`	U1	-	-	Output rate of the NMEA-NAV2-GX-GLL message on port SPI
CFG-MSGOUT-NMEA_NAV2_ID_GLL_UART1	`0x20910671`	U1	-	-	Output rate of the NMEA-NAV2-GX-GLL message on port UART1

Configuration item	Key ID	Type	Scale	Unit	Description
CFG-MSGOUT-NMEA_NAV2_ID_GLL_UART2	`0x20910672`	U1	-	-	Output rate of the NMEA-NAV2-GX-GLL message on port UART2
CFG-MSGOUT-NMEA_NAV2_ID_GLL_USB	`0x20910673`	U1	-	-	Output rate of the NMEA-NAV2-GX-GLL message on port USB
CFG-MSGOUT-NMEA_NAV2_ID_GNS_I2C	`0x2091065c`	U1	-	-	Output rate of the NMEA-NAV2-GX-GNS message on port I2C
CFG-MSGOUT-NMEA_NAV2_ID_GNS_SPI	`0x20910660`	U1	-	-	Output rate of the NMEA-NAV2-GX-GNS message on port SPI
CFG-MSGOUT-NMEA_NAV2_ID_GNS_UART1	`0x2091065d`	U1	-	-	Output rate of the NMEA-NAV2-GX-GNS message on port UART1
CFG-MSGOUT-NMEA_NAV2_ID_GNS_UART2	`0x2091065e`	U1	-	-	Output rate of the NMEA-NAV2-GX-GNS message on port UART2
CFG-MSGOUT-NMEA_NAV2_ID_GNS_USB	`0x2091065f`	U1	-	-	Output rate of the NMEA-NAV2-GX-GNS message on port USB
CFG-MSGOUT-NMEA_NAV2_ID_GSA_I2C	`0x20910666`	U1	-	-	Output rate of the NMEA-NAV2-GX-GSA message on port I2C
CFG-MSGOUT-NMEA_NAV2_ID_GSA_SPI	`0x2091066a`	U1	-	-	Output rate of the NMEA-NAV2-GX-GSA message on port SPI
CFG-MSGOUT-NMEA_NAV2_ID_GSA_UART1	`0x20910667`	U1	-	-	Output rate of the NMEA-NAV2-GX-GSA message on port UART1
CFG-MSGOUT-NMEA_NAV2_ID_GSA_UART2	`0x20910668`	U1	-	-	Output rate of the NMEA-NAV2-GX-GSA message on port UART2
CFG-MSGOUT-NMEA_NAV2_ID_GSA_USB	`0x20910669`	U1	-	-	Output rate of the NMEA-NAV2-GX-GSA message on port USB
CFG-MSGOUT-NMEA_NAV2_ID_RMC_I2C	`0x20910652`	U1	-	-	Output rate of the NMEA-NAV2-GX-RMC message on port I2C
CFG-MSGOUT-NMEA_NAV2_ID_RMC_SPI	`0x20910656`	U1	-	-	Output rate of the NMEA-NAV2-GX-RMC message on port SPI
CFG-MSGOUT-NMEA_NAV2_ID_RMC_UART1	`0x20910653`	U1	-	-	Output rate of the NMEA-NAV2-GX-RMC message on port UART1
CFG-MSGOUT-NMEA_NAV2_ID_RMC_UART2	`0x20910654`	U1	-	-	Output rate of the NMEA-NAV2-GX-RMC message on port UART2
CFG-MSGOUT-NMEA_NAV2_ID_RMC_USB	`0x20910655`	U1	-	-	Output rate of the NMEA-NAV2-GX-RMC message on port USB
CFG-MSGOUT-NMEA_NAV2_ID_VTG_I2C	`0x20910657`	U1	-	-	Output rate of the NMEA-NAV2-GX-VTG message on port I2C
CFG-MSGOUT-NMEA_NAV2_ID_VTG_SPI	`0x2091065b`	U1	-	-	Output rate of the NMEA-NAV2-GX-VTG message on port SPI
CFG-MSGOUT-NMEA_NAV2_ID_VTG_UART1	`0x20910658`	U1	-	-	Output rate of the NMEA-NAV2-GX-VTG message on port UART1
CFG-MSGOUT-NMEA_NAV2_ID_VTG_UART2	`0x20910659`	U1	-	-	Output rate of the NMEA-NAV2-GX-VTG message on port UART2
CFG-MSGOUT-NMEA_NAV2_ID_VTG_USB	`0x2091065a`	U1	-	-	Output rate of the NMEA-NAV2-GX-VTG message on port USB
CFG-MSGOUT-NMEA_NAV2_ID_ZDA_I2C	`0x2091067f`	U1	-	-	Output rate of the NMEA-NAV2-GX-ZDA message on port I2C
CFG-MSGOUT-NMEA_NAV2_ID_ZDA_SPI	`0x20910683`	U1	-	-	Output rate of the NMEA-NAV2-GX-ZDA message on port SPI
CFG-MSGOUT-NMEA_NAV2_ID_ZDA_UART1	`0x20910680`	U1	-	-	Output rate of the NMEA-NAV2-GX-ZDA message on port UART1
CFG-MSGOUT-NMEA_NAV2_ID_ZDA_UART2	`0x20910681`	U1	-	-	Output rate of the NMEA-NAV2-GX-ZDA message on port UART2

Configuration item	Key ID	Type	Scale	Unit	Description
CFG-MSGOUT-NMEA_NAV2_ID_ZDA_USB	0x20910682	U1	-	-	Output rate of the NMEA-NAV2-GX-ZDA message on port USB
CFG-MSGOUT-PUBX_ID_POLYP_I2C	0x209100ec	U1	-	-	Output rate of the NMEA-GX-PUBX00 message on port I2C
CFG-MSGOUT-PUBX_ID_POLYP_SPI	0x209100f0	U1	-	-	Output rate of the NMEA-GX-PUBX00 message on port SPI
CFG-MSGOUT-PUBX_ID_POLYP_UART1	0x209100ed	U1	-	-	Output rate of the NMEA-GX-PUBX00 message on port UART1
CFG-MSGOUT-PUBX_ID_POLYP_UART2	0x209100ee	U1	-	-	Output rate of the NMEA-GX-PUBX00 message on port UART2
CFG-MSGOUT-PUBX_ID_POLYP_USB	0x209100ef	U1	-	-	Output rate of the NMEA-GX-PUBX00 message on port USB
CFG-MSGOUT-PUBX_ID_POLYS_I2C	0x209100f1	U1	-	-	Output rate of the NMEA-GX-PUBX03 message on port I2C
CFG-MSGOUT-PUBX_ID_POLYS_SPI	0x209100f5	U1	-	-	Output rate of the NMEA-GX-PUBX03 message on port SPI
CFG-MSGOUT-PUBX_ID_POLYS_UART1	0x209100f2	U1	-	-	Output rate of the NMEA-GX-PUBX03 message on port UART1
CFG-MSGOUT-PUBX_ID_POLYS_UART2	0x209100f3	U1	-	-	Output rate of the NMEA-GX-PUBX03 message on port UART2
CFG-MSGOUT-PUBX_ID_POLYS_USB	0x209100f4	U1	-	-	Output rate of the NMEA-GX-PUBX03 message on port USB
CFG-MSGOUT-PUBX_ID_POLYT_I2C	0x209100f6	U1	-	-	Output rate of the NMEA-GX-PUBX04 message on port I2C
CFG-MSGOUT-PUBX_ID_POLYT_SPI	0x209100fa	U1	-	-	Output rate of the NMEA-GX-PUBX04 message on port SPI
CFG-MSGOUT-PUBX_ID_POLYT_UART1	0x209100f7	U1	-	-	Output rate of the NMEA-GX-PUBX04 message on port UART1
CFG-MSGOUT-PUBX_ID_POLYT_UART2	0x209100f8	U1	-	-	Output rate of the NMEA-GX-PUBX04 message on port UART2
CFG-MSGOUT-PUBX_ID_POLYT_USB	0x209100f9	U1	-	-	Output rate of the NMEA-GX-PUBX04 message on port USB
CFG-MSGOUT-RTCM_3X_TYPE1005_I2C	0x209102bd	U1	-	-	Output rate of the RTCM-3X-TYPE1005 message on port I2C
CFG-MSGOUT-RTCM_3X_TYPE1005_SPI	0x209102c1	U1	-	-	Output rate of the RTCM-3X-TYPE1005 message on port SPI
CFG-MSGOUT-RTCM_3X_TYPE1005_UART1	0x209102be	U1	-	-	Output rate of the RTCM-3X-TYPE1005 message on port UART1
CFG-MSGOUT-RTCM_3X_TYPE1005_UART2	0x209102bf	U1	-	-	Output rate of the RTCM-3X-TYPE1005 message on port UART2
CFG-MSGOUT-RTCM_3X_TYPE1005_USB	0x209102c0	U1	-	-	Output rate of the RTCM-3X-TYPE1005 message on port USB
CFG-MSGOUT-RTCM_3X_TYPE1074_I2C	0x2091035e	U1	-	-	Output rate of the RTCM-3X-TYPE1074 message on port I2C
CFG-MSGOUT-RTCM_3X_TYPE1074_SPI	0x20910362	U1	-	-	Output rate of the RTCM-3X-TYPE1074 message on port SPI
CFG-MSGOUT-RTCM_3X_TYPE1074_UART1	0x2091035f	U1	-	-	Output rate of the RTCM-3X-TYPE1074 message on port UART1
CFG-MSGOUT-RTCM_3X_TYPE1074_UART2	0x20910360	U1	-	-	Output rate of the RTCM-3X-TYPE1074 message on port UART2
CFG-MSGOUT-RTCM_3X_TYPE1074_USB	0x20910361	U1	-	-	Output rate of the RTCM-3X-TYPE1074 message on port USB

Configuration item	Key ID	Type	Scale	Unit	Description
CFG-MSGOUT-RTCM_3X_TYPE1077_I2C	`0x209102cc`	U1	-	-	Output rate of the RTCM-3X-TYPE1077 message on port I2C
CFG-MSGOUT-RTCM_3X_TYPE1077_SPI	`0x209102d0`	U1	-	-	Output rate of the RTCM-3X-TYPE1077 message on port SPI
CFG-MSGOUT-RTCM_3X_TYPE1077_UART1	`0x209102cd`	U1	-	-	Output rate of the RTCM-3X-TYPE1077 message on port UART1
CFG-MSGOUT-RTCM_3X_TYPE1077_UART2	`0x209102ce`	U1	-	-	Output rate of the RTCM-3X-TYPE1077 message on port UART2
CFG-MSGOUT-RTCM_3X_TYPE1077_USB	`0x209102cf`	U1	-	-	Output rate of the RTCM-3X-TYPE1077 message on port USB
CFG-MSGOUT-RTCM_3X_TYPE1084_I2C	`0x20910363`	U1	-	-	Output rate of the RTCM-3X-TYPE1084 message on port I2C
CFG-MSGOUT-RTCM_3X_TYPE1084_SPI	`0x20910367`	U1	-	-	Output rate of the RTCM-3X-TYPE1084 message on port SPI
CFG-MSGOUT-RTCM_3X_TYPE1084_UART1	`0x20910364`	U1	-	-	Output rate of the RTCM-3X-TYPE1084 message on port UART1
CFG-MSGOUT-RTCM_3X_TYPE1084_UART2	`0x20910365`	U1	-	-	Output rate of the RTCM-3X-TYPE1084 message on port UART2
CFG-MSGOUT-RTCM_3X_TYPE1084_USB	`0x20910366`	U1	-	-	Output rate of the RTCM-3X-TYPE1084 message on port USB
CFG-MSGOUT-RTCM_3X_TYPE1087_I2C	`0x209102d1`	U1	-	-	Output rate of the RTCM-3X-TYPE1087 message on port I2C
CFG-MSGOUT-RTCM_3X_TYPE1087_SPI	`0x209102d5`	U1	-	-	Output rate of the RTCM-3X-TYPE1087 message on port SPI
CFG-MSGOUT-RTCM_3X_TYPE1087_UART1	`0x209102d2`	U1	-	-	Output rate of the RTCM-3X-TYPE1087 message on port UART1
CFG-MSGOUT-RTCM_3X_TYPE1087_UART2	`0x209102d3`	U1	-	-	Output rate of the RTCM-3X-TYPE1087 message on port UART2
CFG-MSGOUT-RTCM_3X_TYPE1087_USB	`0x209102d4`	U1	-	-	Output rate of the RTCM-3X-TYPE1087 message on port USB
CFG-MSGOUT-RTCM_3X_TYPE1094_I2C	`0x20910368`	U1	-	-	Output rate of the RTCM-3X-TYPE1094 message on port I2C
CFG-MSGOUT-RTCM_3X_TYPE1094_SPI	`0x2091036c`	U1	-	-	Output rate of the RTCM-3X-TYPE1094 message on port SPI
CFG-MSGOUT-RTCM_3X_TYPE1094_UART1	`0x20910369`	U1	-	-	Output rate of the RTCM-3X-TYPE1094 message on port UART1
CFG-MSGOUT-RTCM_3X_TYPE1094_UART2	`0x2091036a`	U1	-	-	Output rate of the RTCM-3X-TYPE1094 message on port UART2
CFG-MSGOUT-RTCM_3X_TYPE1094_USB	`0x2091036b`	U1	-	-	Output rate of the RTCM-3X-TYPE1094 message on port USB
CFG-MSGOUT-RTCM_3X_TYPE1097_I2C	`0x20910318`	U1	-	-	Output rate of the RTCM-3X-TYPE1097 message on port I2C
CFG-MSGOUT-RTCM_3X_TYPE1097_SPI	`0x2091031c`	U1	-	-	Output rate of the RTCM-3X-TYPE1097 message on port SPI
CFG-MSGOUT-RTCM_3X_TYPE1097_UART1	`0x20910319`	U1	-	-	Output rate of the RTCM-3X-TYPE1097 message on port UART1
CFG-MSGOUT-RTCM_3X_TYPE1097_UART2	`0x2091031a`	U1	-	-	Output rate of the RTCM-3X-TYPE1097 message on port UART2
CFG-MSGOUT-RTCM_3X_TYPE1097_USB	`0x2091031b`	U1	-	-	Output rate of the RTCM-3X-TYPE1097 message on port USB
CFG-MSGOUT-RTCM_3X_TYPE1124_I2C	`0x2091036d`	U1	-	-	Output rate of the RTCM-3X-TYPE1124 message on port I2C

Configuration item	Key ID	Type	Scale	Unit	Description
CFG-MSGOUT-RTCM_3X_TYPE1124_SPI	`0x20910371`	U1	-	-	Output rate of the RTCM-3X-TYPE1124 message on port SPI
CFG-MSGOUT-RTCM_3X_TYPE1124_UART1	`0x2091036e`	U1	-	-	Output rate of the RTCM-3X-TYPE1124 message on port UART1
CFG-MSGOUT-RTCM_3X_TYPE1124_UART2	`0x2091036f`	U1	-	-	Output rate of the RTCM-3X-TYPE1124 message on port UART2
CFG-MSGOUT-RTCM_3X_TYPE1124_USB	`0x20910370`	U1	-	-	Output rate of the RTCM-3X-TYPE1124 message on port USB
CFG-MSGOUT-RTCM_3X_TYPE1127_I2C	`0x209102d6`	U1	-	-	Output rate of the RTCM-3X-TYPE1127 message on port I2C
CFG-MSGOUT-RTCM_3X_TYPE1127_SPI	`0x209102da`	U1	-	-	Output rate of the RTCM-3X-TYPE1127 message on port SPI
CFG-MSGOUT-RTCM_3X_TYPE1127_UART1	`0x209102d7`	U1	-	-	Output rate of the RTCM-3X-TYPE1127 message on port UART1
CFG-MSGOUT-RTCM_3X_TYPE1127_UART2	`0x209102d8`	U1	-	-	Output rate of the RTCM-3X-TYPE1127 message on port UART2
CFG-MSGOUT-RTCM_3X_TYPE1127_USB	`0x209102d9`	U1	-	-	Output rate of the RTCM-3X-TYPE1127 message on port USB
CFG-MSGOUT-RTCM_3X_TYPE1230_I2C	`0x20910303`	U1	-	-	Output rate of the RTCM-3X-TYPE1230 message on port I2C
CFG-MSGOUT-RTCM_3X_TYPE1230_SPI	`0x20910307`	U1	-	-	Output rate of the RTCM-3X-TYPE1230 message on port SPI
CFG-MSGOUT-RTCM_3X_TYPE1230_UART1	`0x20910304`	U1	-	-	Output rate of the RTCM-3X-TYPE1230 message on port UART1
CFG-MSGOUT-RTCM_3X_TYPE1230_UART2	`0x20910305`	U1	-	-	Output rate of the RTCM-3X-TYPE1230 message on port UART2
CFG-MSGOUT-RTCM_3X_TYPE1230_USB	`0x20910306`	U1	-	-	Output rate of the RTCM-3X-TYPE1230 message on port USB
CFG-MSGOUT-RTCM_3X_TYPE4072_0_I2C	`0x209102fe`	U1	-	-	Output rate of the RTCM-3X-TYPE4072_0 message on port I2C
CFG-MSGOUT-RTCM_3X_TYPE4072_0_SPI	`0x20910302`	U1	-	-	Output rate of the RTCM-3X-TYPE4072_0 message on port SPI
CFG-MSGOUT-RTCM_3X_TYPE4072_0_UART1	`0x209102ff`	U1	-	-	Output rate of the RTCM-3X-TYPE4072_0 message on port UART1
CFG-MSGOUT-RTCM_3X_TYPE4072_0_UART2	`0x20910300`	U1	-	-	Output rate of the RTCM-3X-TYPE4072_0 message on port UART2
CFG-MSGOUT-RTCM_3X_TYPE4072_0_USB	`0x20910301`	U1	-	-	Output rate of the RTCM-3X-TYPE4072_0 message on port USB
CFG-MSGOUT-RTCM_3X_TYPE4072_1_I2C	`0x20910381`	U1	-	-	Output rate of the RTCM-3X-TYPE4072_1 message on port I2C
CFG-MSGOUT-RTCM_3X_TYPE4072_1_SPI	`0x20910385`	U1	-	-	Output rate of the RTCM-3X-TYPE4072_1 message on port SPI
CFG-MSGOUT-RTCM_3X_TYPE4072_1_UART1	`0x20910382`	U1	-	-	Output rate of the RTCM-3X-TYPE4072_1 message on port UART1
CFG-MSGOUT-RTCM_3X_TYPE4072_1_UART2	`0x20910383`	U1	-	-	Output rate of the RTCM-3X-TYPE4072_1 message on port UART2
CFG-MSGOUT-RTCM_3X_TYPE4072_1_USB	`0x20910384`	U1	-	-	Output rate of the RTCM-3X-TYPE4072_1 message on port USB
CFG-MSGOUT-UBX_LOG_INFO_I2C	`0x20910259`	U1	-	-	Output rate of the UBX-LOG-INFO message on port I2C
CFG-MSGOUT-UBX_LOG_INFO_SPI	`0x2091025d`	U1	-	-	Output rate of the UBX-LOG-INFO message on port SPI

Configuration item	Key ID	Type	Scale	Unit	Description
CFG-MSGOUT-UBX_LOG_INFO_UART1	0x2091025a	U1	-	-	Output rate of the UBX-LOG-INFO message on port UART1
CFG-MSGOUT-UBX_LOG_INFO_UART2	0x2091025b	U1	-	-	Output rate of the UBX-LOG-INFO message on port UART2
CFG-MSGOUT-UBX_LOG_INFO_USB	0x2091025c	U1	-	-	Output rate of the UBX-LOG-INFO message on port USB
CFG-MSGOUT-UBX_MON_COMMS_I2C	0x2091034f	U1	-	-	Output rate of the UBX-MON-COMMS message on port I2C
CFG-MSGOUT-UBX_MON_COMMS_SPI	0x20910353	U1	-	-	Output rate of the UBX-MON-COMMS message on port SPI
CFG-MSGOUT-UBX_MON_COMMS_UART1	0x20910350	U1	-	-	Output rate of the UBX-MON-COMMS message on port UART1
CFG-MSGOUT-UBX_MON_COMMS_UART2	0x20910351	U1	-	-	Output rate of the UBX-MON-COMMS message on port UART2
CFG-MSGOUT-UBX_MON_COMMS_USB	0x20910352	U1	-	-	Output rate of the UBX-MON-COMMS message on port USB
CFG-MSGOUT-UBX_MON_HW2_I2C	0x209101b9	U1	-	-	Output rate of the UBX-MON-HW2 message on port I2C
CFG-MSGOUT-UBX_MON_HW2_SPI	0x209101bd	U1	-	-	Output rate of the UBX-MON-HW2 message on port SPI
CFG-MSGOUT-UBX_MON_HW2_UART1	0x209101ba	U1	-	-	Output rate of the UBX-MON-HW2 message on port UART1
CFG-MSGOUT-UBX_MON_HW2_UART2	0x209101bb	U1	-	-	Output rate of the UBX-MON-HW2 message on port UART2
CFG-MSGOUT-UBX_MON_HW2_USB	0x209101bc	U1	-	-	Output rate of the UBX-MON-HW2 message on port USB
CFG-MSGOUT-UBX_MON_HW3_I2C	0x20910354	U1	-	-	Output rate of the UBX-MON-HW3 message on port I2C
CFG-MSGOUT-UBX_MON_HW3_SPI	0x20910358	U1	-	-	Output rate of the UBX-MON-HW3 message on port SPI
CFG-MSGOUT-UBX_MON_HW3_UART1	0x20910355	U1	-	-	Output rate of the UBX-MON-HW3 message on port UART1
CFG-MSGOUT-UBX_MON_HW3_UART2	0x20910356	U1	-	-	Output rate of the UBX-MON-HW3 message on port UART2
CFG-MSGOUT-UBX_MON_HW3_USB	0x20910357	U1	-	-	Output rate of the UBX-MON-HW3 message on port USB
CFG-MSGOUT-UBX_MON_HW_I2C	0x209101b4	U1	-	-	Output rate of the UBX-MON-HW message on port I2C
CFG-MSGOUT-UBX_MON_HW_SPI	0x209101b8	U1	-	-	Output rate of the UBX-MON-HW message on port SPI
CFG-MSGOUT-UBX_MON_HW_UART1	0x209101b5	U1	-	-	Output rate of the UBX-MON-HW message on port UART1
CFG-MSGOUT-UBX_MON_HW_UART2	0x209101b6	U1	-	-	Output rate of the UBX-MON-HW message on port UART2
CFG-MSGOUT-UBX_MON_HW_USB	0x209101b7	U1	-	-	Output rate of the UBX-MON-HW message on port USB
CFG-MSGOUT-UBX_MON_IO_I2C	0x209101a5	U1	-	-	Output rate of the UBX-MON-IO message on port I2C
CFG-MSGOUT-UBX_MON_IO_SPI	0x209101a9	U1	-	-	Output rate of the UBX-MON-IO message on port SPI
CFG-MSGOUT-UBX_MON_IO_UART1	0x209101a6	U1	-	-	Output rate of the UBX-MON-IO message on port UART1

Configuration item	Key ID	Type	Scale	Unit	Description
CFG-MSGOUT-UBX_MON_IO_UART2	`0x209101a7`	U1	-	-	Output rate of the UBX-MON-IO message on port UART2
CFG-MSGOUT-UBX_MON_IO_USB	`0x209101a8`	U1	-	-	Output rate of the UBX-MON-IO message on port USB
CFG-MSGOUT-UBX_MON_MSGPP_I2C	`0x20910196`	U1	-	-	Output rate of the UBX-MON-MSGPP message on port I2C
CFG-MSGOUT-UBX_MON_MSGPP_SPI	`0x2091019a`	U1	-	-	Output rate of the UBX-MON-MSGPP message on port SPI
CFG-MSGOUT-UBX_MON_MSGPP_UART1	`0x20910197`	U1	-	-	Output rate of the UBX-MON-MSGPP message on port UART1
CFG-MSGOUT-UBX_MON_MSGPP_UART2	`0x20910198`	U1	-	-	Output rate of the UBX-MON-MSGPP message on port UART2
CFG-MSGOUT-UBX_MON_MSGPP_USB	`0x20910199`	U1	-	-	Output rate of the UBX-MON-MSGPP message on port USB
CFG-MSGOUT-UBX_MON_RF_I2C	`0x20910359`	U1	-	-	Output rate of the UBX-MON-RF message on port I2C
CFG-MSGOUT-UBX_MON_RF_SPI	`0x2091035d`	U1	-	-	Output rate of the UBX-MON-RF message on port SPI
CFG-MSGOUT-UBX_MON_RF_UART1	`0x2091035a`	U1	-	-	Output rate of the UBX-MON-RF message on port UART1
CFG-MSGOUT-UBX_MON_RF_UART2	`0x2091035b`	U1	-	-	Output rate of the UBX-MON-RF message on port UART2
CFG-MSGOUT-UBX_MON_RF_USB	`0x2091035c`	U1	-	-	Output rate of the UBX-MON-RF message on port USB
CFG-MSGOUT-UBX_MON_RXBUF_I2C	`0x209101a0`	U1	-	-	Output rate of the UBX-MON-RXBUF message on port I2C
CFG-MSGOUT-UBX_MON_RXBUF_SPI	`0x209101a4`	U1	-	-	Output rate of the UBX-MON-RXBUF message on port SPI
CFG-MSGOUT-UBX_MON_RXBUF_UART1	`0x209101a1`	U1	-	-	Output rate of the UBX-MON-RXBUF message on port UART1
CFG-MSGOUT-UBX_MON_RXBUF_UART2	`0x209101a2`	U1	-	-	Output rate of the UBX-MON-RXBUF message on port UART2
CFG-MSGOUT-UBX_MON_RXBUF_USB	`0x209101a3`	U1	-	-	Output rate of the UBX-MON-RXBUF message on port USB
CFG-MSGOUT-UBX_MON_RXR_I2C	`0x20910187`	U1	-	-	Output rate of the UBX-MON-RXR message on port I2C
CFG-MSGOUT-UBX_MON_RXR_SPI	`0x2091018b`	U1	-	-	Output rate of the UBX-MON-RXR message on port SPI
CFG-MSGOUT-UBX_MON_RXR_UART1	`0x20910188`	U1	-	-	Output rate of the UBX-MON-RXR message on port UART1
CFG-MSGOUT-UBX_MON_RXR_UART2	`0x20910189`	U1	-	-	Output rate of the UBX-MON-RXR message on port UART2
CFG-MSGOUT-UBX_MON_RXR_USB	`0x2091018a`	U1	-	-	Output rate of the UBX-MON-RXR message on port USB
CFG-MSGOUT-UBX_MON_SPAN_I2C	`0x2091038b`	U1	-	-	Output rate of the UBX-MON-SPAN message on port I2C
CFG-MSGOUT-UBX_MON_SPAN_SPI	`0x2091038f`	U1	-	-	Output rate of the UBX-MON-SPAN message on port SPI
CFG-MSGOUT-UBX_MON_SPAN_UART1	`0x2091038c`	U1	-	-	Output rate of the UBX-MON-SPAN message on port UART1
CFG-MSGOUT-UBX_MON_SPAN_UART2	`0x2091038d`	U1	-	-	Output rate of the UBX-MON-SPAN message on port UART2

Configuration item	Key ID	Type	Scale	Unit	Description
CFG-MSGOUT-UBX_MON_SPAN_USB	0x2091038e	U1	-	-	Output rate of the UBX-MON-SPAN message on port USB
CFG-MSGOUT-UBX_MON_SYS_I2C	0x2091069d	U1	-	-	Output rate of the UBX-MON-SYS message on port I2C
CFG-MSGOUT-UBX_MON_SYS_SPI	0x209106a1	U1	-	-	Output rate of the UBX-MON-SYS message on port SPI
CFG-MSGOUT-UBX_MON_SYS_UART1	0x2091069e	U1	-	-	Output rate of the UBX-MON-SYS message on port UART1
CFG-MSGOUT-UBX_MON_SYS_UART2	0x2091069f	U1	-	-	Output rate of the UBX-MON-SYS message on port UART2
CFG-MSGOUT-UBX_MON_SYS_USB	0x209106a0	U1	-	-	Output rate of the UBX-MON-SYS message on port USB
CFG-MSGOUT-UBX_MON_TXBUF_I2C	0x2091019b	U1	-	-	Output rate of the UBX-MON-TXBUF message on port I2C
CFG-MSGOUT-UBX_MON_TXBUF_SPI	0x2091019f	U1	-	-	Output rate of the UBX-MON-TXBUF message on port SPI
CFG-MSGOUT-UBX_MON_TXBUF_UART1	0x2091019c	U1	-	-	Output rate of the UBX-MON-TXBUF message on port UART1
CFG-MSGOUT-UBX_MON_TXBUF_UART2	0x2091019d	U1	-	-	Output rate of the UBX-MON-TXBUF message on port UART2
CFG-MSGOUT-UBX_MON_TXBUF_USB	0x2091019e	U1	-	-	Output rate of the UBX-MON-TXBUF message on port USB
CFG-MSGOUT-UBX_NAV2_CLOCK_I2C	0x20910430	U1	-	-	Output rate of the UBX-NAV2-CLOCK message on port I2C
CFG-MSGOUT-UBX_NAV2_CLOCK_SPI	0x20910434	U1	-	-	Output rate of the UBX-NAV2-CLOCK message on port SPI
CFG-MSGOUT-UBX_NAV2_CLOCK_UART1	0x20910431	U1	-	-	Output rate of the UBX-NAV2-CLOCK message on port UART1
CFG-MSGOUT-UBX_NAV2_CLOCK_UART2	0x20910432	U1	-	-	Output rate of the UBX-NAV2-CLOCK message on port UART2
CFG-MSGOUT-UBX_NAV2_CLOCK_USB	0x20910433	U1	-	-	Output rate of the UBX-NAV2-CLOCK message on port USB
CFG-MSGOUT-UBX_NAV2_COV_I2C	0x20910435	U1	-	-	Output rate of the UBX-NAV2-COV message on port I2C
CFG-MSGOUT-UBX_NAV2_COV_SPI	0x20910439	U1	-	-	Output rate of the UBX-NAV2-COV message on port SPI
CFG-MSGOUT-UBX_NAV2_COV_UART1	0x20910436	U1	-	-	Output rate of the UBX-NAV2-COV message on port UART1
CFG-MSGOUT-UBX_NAV2_COV_UART2	0x20910437	U1	-	-	Output rate of the UBX-NAV2-COV message on port UART2
CFG-MSGOUT-UBX_NAV2_COV_USB	0x20910438	U1	-	-	Output rate of the UBX-NAV2-COV message on port USB
CFG-MSGOUT-UBX_NAV2_DOP_I2C	0x20910465	U1	-	-	Output rate of the UBX-NAV2-DOP message on port I2C
CFG-MSGOUT-UBX_NAV2_DOP_SPI	0x20910469	U1	-	-	Output rate of the UBX-NAV2-DOP message on port SPI
CFG-MSGOUT-UBX_NAV2_DOP_UART1	0x20910466	U1	-	-	Output rate of the UBX-NAV2-DOP message on port UART1
CFG-MSGOUT-UBX_NAV2_DOP_UART2	0x20910467	U1	-	-	Output rate of the UBX-NAV2-DOP message on port UART2
CFG-MSGOUT-UBX_NAV2_DOP_USB	0x20910468	U1	-	-	Output rate of the UBX-NAV2-DOP message on port USB

Configuration item	Key ID	Type	Scale	Unit	Description
CFG-MSGOUT-UBX_NAV2_EOE_I2C	0x20910565	U1	-	-	Output rate of the UBX-NAV2-EOE message on port I2C
CFG-MSGOUT-UBX_NAV2_EOE_SPI	0x20910569	U1	-	-	Output rate of the UBX-NAV2-EOE message on port SPI
CFG-MSGOUT-UBX_NAV2_EOE_UART1	0x20910566	U1	-	-	Output rate of the UBX-NAV2-EOE message on port UART1
CFG-MSGOUT-UBX_NAV2_EOE_UART2	0x20910567	U1	-	-	Output rate of the UBX-NAV2-EOE message on port UART2
CFG-MSGOUT-UBX_NAV2_EOE_USB	0x20910568	U1	-	-	Output rate of the UBX-NAV2-EOE message on port USB
CFG-MSGOUT-UBX_NAV2_ODO_I2C	0x20910475	U1	-	-	Output rate of the UBX-NAV2-ODO message on port I2C
CFG-MSGOUT-UBX_NAV2_ODO_SPI	0x20910479	U1	-	-	Output rate of the UBX-NAV2-ODO message on port SPI
CFG-MSGOUT-UBX_NAV2_ODO_UART1	0x20910476	U1	-	-	Output rate of the UBX-NAV2-ODO message on port UART1
CFG-MSGOUT-UBX_NAV2_ODO_UART2	0x20910477	U1	-	-	Output rate of the UBX-NAV2-ODO message on port UART2
CFG-MSGOUT-UBX_NAV2_ODO_USB	0x20910478	U1	-	-	Output rate of the UBX-NAV2-ODO message on port USB
CFG-MSGOUT-UBX_NAV2_POSECEF_I2C	0x20910480	U1	-	-	Output rate of the UBX-NAV2-POSECEF message on port I2C
CFG-MSGOUT-UBX_NAV2_POSECEF_SPI	0x20910484	U1	-	-	Output rate of the UBX-NAV2-POSECEF message on port SPI
CFG-MSGOUT-UBX_NAV2_POSECEF_UART1	0x20910481	U1	-	-	Output rate of the UBX-NAV2-POSECEF message on port UART1
CFG-MSGOUT-UBX_NAV2_POSECEF_UART2	0x20910482	U1	-	-	Output rate of the UBX-NAV2-POSECEF message on port UART2
CFG-MSGOUT-UBX_NAV2_POSECEF_USB	0x20910483	U1	-	-	Output rate of the UBX-NAV2-POSECEF message on port USB
CFG-MSGOUT-UBX_NAV2_POSLLH_I2C	0x20910485	U1	-	-	Output rate of the UBX-NAV2-POSLLH message on port I2C
CFG-MSGOUT-UBX_NAV2_POSLLH_SPI	0x20910489	U1	-	-	Output rate of the UBX-NAV2-POSLLH message on port SPI
CFG-MSGOUT-UBX_NAV2_POSLLH_UART1	0x20910486	U1	-	-	Output rate of the UBX-NAV2-POSLLH message on port UART1
CFG-MSGOUT-UBX_NAV2_POSLLH_UART2	0x20910487	U1	-	-	Output rate of the UBX-NAV2-POSLLH message on port UART2
CFG-MSGOUT-UBX_NAV2_POSLLH_USB	0x20910488	U1	-	-	Output rate of the UBX-NAV2-POSLLH message on port USB
CFG-MSGOUT-UBX_NAV2_PVT_I2C	0x20910490	U1	-	-	Output rate of the UBX-NAV2-PVT message on port I2C
CFG-MSGOUT-UBX_NAV2_PVT_SPI	0x20910494	U1	-	-	Output rate of the UBX-NAV2-PVT message on port SPI
CFG-MSGOUT-UBX_NAV2_PVT_UART1	0x20910491	U1	-	-	Output rate of the UBX-NAV2-PVT message on port UART1
CFG-MSGOUT-UBX_NAV2_PVT_UART2	0x20910492	U1	-	-	Output rate of the UBX-NAV2-PVT message on port UART2
CFG-MSGOUT-UBX_NAV2_PVT_USB	0x20910493	U1	-	-	Output rate of the UBX-NAV2-PVT message on port USB
CFG-MSGOUT-UBX_NAV2_SAT_I2C	0x20910495	U1	-	-	Output rate of the UBX-NAV2-SAT message on port I2C

Configuration item	Key ID	Type	Scale	Unit	Description
CFG-MSGOUT-UBX_NAV2_SAT_SPI	`0x20910499`	U1	-	-	Output rate of the UBX-NAV2-SAT message on port SPI
CFG-MSGOUT-UBX_NAV2_SAT_UART1	`0x20910496`	U1	-	-	Output rate of the UBX-NAV2-SAT message on port UART1
CFG-MSGOUT-UBX_NAV2_SAT_UART2	`0x20910497`	U1	-	-	Output rate of the UBX-NAV2-SAT message on port UART2
CFG-MSGOUT-UBX_NAV2_SAT_USB	`0x20910498`	U1	-	-	Output rate of the UBX-NAV2-SAT message on port USB
CFG-MSGOUT-UBX_NAV2_SBAS_I2C	`0x20910500`	U1	-	-	Output rate of the UBX-NAV2-SBAS message on port I2C
CFG-MSGOUT-UBX_NAV2_SBAS_SPI	`0x20910504`	U1	-	-	Output rate of the UBX-NAV2-SBAS message on port SPI
CFG-MSGOUT-UBX_NAV2_SBAS_UART1	`0x20910501`	U1	-	-	Output rate of the UBX-NAV2-SBAS message on port UART1
CFG-MSGOUT-UBX_NAV2_SBAS_UART2	`0x20910502`	U1	-	-	Output rate of the UBX-NAV2-SBAS message on port UART2
CFG-MSGOUT-UBX_NAV2_SBAS_USB	`0x20910503`	U1	-	-	Output rate of the UBX-NAV2-SBAS message on port USB
CFG-MSGOUT-UBX_NAV2_SIG_I2C	`0x20910505`	U1	-	-	Output rate of the UBX-NAV2-SIG message on port I2C
CFG-MSGOUT-UBX_NAV2_SIG_SPI	`0x20910509`	U1	-	-	Output rate of the UBX-NAV2-SIG message on port SPI
CFG-MSGOUT-UBX_NAV2_SIG_UART1	`0x20910506`	U1	-	-	Output rate of the UBX-NAV2-SIG message on port UART1
CFG-MSGOUT-UBX_NAV2_SIG_UART2	`0x20910507`	U1	-	-	Output rate of the UBX-NAV2-SIG message on port UART2
CFG-MSGOUT-UBX_NAV2_SIG_USB	`0x20910508`	U1	-	-	Output rate of the UBX-NAV2-SIG message on port USB
CFG-MSGOUT-UBX_NAV2_SLAS_I2C	`0x20910510`	U1	-	-	Output rate of the UBX-NAV2-SLAS message on port I2C
CFG-MSGOUT-UBX_NAV2_SLAS_SPI	`0x20910514`	U1	-	-	Output rate of the UBX-NAV2-SLAS message on port SPI
CFG-MSGOUT-UBX_NAV2_SLAS_UART1	`0x20910511`	U1	-	-	Output rate of the UBX-NAV2-SLAS message on port UART1
CFG-MSGOUT-UBX_NAV2_SLAS_UART2	`0x20910512`	U1	-	-	Output rate of the UBX-NAV2-SLAS message on port UART2
CFG-MSGOUT-UBX_NAV2_SLAS_USB	`0x20910513`	U1	-	-	Output rate of the UBX-NAV2-SLAS message on port USB
CFG-MSGOUT-UBX_NAV2_STATUS_I2C	`0x20910515`	U1	-	-	Output rate of the UBX-NAV2-STATUS message on port I2C
CFG-MSGOUT-UBX_NAV2_STATUS_SPI	`0x20910519`	U1	-	-	Output rate of the UBX-NAV2-STATUS message on port SPI
CFG-MSGOUT-UBX_NAV2_STATUS_UART1	`0x20910516`	U1	-	-	Output rate of the UBX-NAV2-STATUS message on port UART1
CFG-MSGOUT-UBX_NAV2_STATUS_UART2	`0x20910517`	U1	-	-	Output rate of the UBX-NAV2-STATUS message on port UART2
CFG-MSGOUT-UBX_NAV2_STATUS_USB	`0x20910518`	U1	-	-	Output rate of the UBX-NAV2-STATUS message on port USB
CFG-MSGOUT-UBX_NAV2_SVIN_I2C	`0x20910520`	U1	-	-	Output rate of the UBX-NAV2-SVIN message on port I2C
CFG-MSGOUT-UBX_NAV2_SVIN_SPI	`0x20910524`	U1	-	-	Output rate of the UBX-NAV2-SVIN message on port SPI

Configuration item	Key ID	Type	Scale	Unit	Description
CFG-MSGOUT-UBX_NAV2_SVIN_UART1	0x20910521	U1	-	-	Output rate of the UBX-NAV2-SVIN message on port UART1
CFG-MSGOUT-UBX_NAV2_SVIN_UART2	0x20910522	U1	-	-	Output rate of the UBX-NAV2-SVIN message on port UART2
CFG-MSGOUT-UBX_NAV2_SVIN_USB	0x20910523	U1	-	-	Output rate of the UBX-NAV2-SVIN message on port USB
CFG-MSGOUT-UBX_NAV2_TIMEBDS_I2C	0x20910525	U1	-	-	Output rate of the UBX-NAV2-TIMEBDS message on port I2C
CFG-MSGOUT-UBX_NAV2_TIMEBDS_SPI	0x20910529	U1	-	-	Output rate of the UBX-NAV2-TIMEBDS message on port SPI
CFG-MSGOUT-UBX_NAV2_TIMEBDS_UART1	0x20910526	U1	-	-	Output rate of the UBX-NAV2-TIMEBDS message on port UART1
CFG-MSGOUT-UBX_NAV2_TIMEBDS_UART2	0x20910527	U1	-	-	Output rate of the UBX-NAV2-TIMEBDS message on port UART2
CFG-MSGOUT-UBX_NAV2_TIMEBDS_USB	0x20910528	U1	-	-	Output rate of the UBX-NAV2-TIMEBDS message on port USB
CFG-MSGOUT-UBX_NAV2_TIMEGAL_I2C	0x20910530	U1	-	-	Output rate of the UBX-NAV2-TIMEGAL message on port I2C
CFG-MSGOUT-UBX_NAV2_TIMEGAL_SPI	0x20910534	U1	-	-	Output rate of the UBX-NAV2-TIMEGAL message on port SPI
CFG-MSGOUT-UBX_NAV2_TIMEGAL_UART1	0x20910531	U1	-	-	Output rate of the UBX-NAV2-TIMEGAL message on port UART1
CFG-MSGOUT-UBX_NAV2_TIMEGAL_UART2	0x20910532	U1	-	-	Output rate of the UBX-NAV2-TIMEGAL message on port UART2
CFG-MSGOUT-UBX_NAV2_TIMEGAL_USB	0x20910533	U1	-	-	Output rate of the UBX-NAV2-TIMEGAL message on port USB
CFG-MSGOUT-UBX_NAV2_TIMEGLO_I2C	0x20910535	U1	-	-	Output rate of the UBX-NAV2-TIMEGLO message on port I2C
CFG-MSGOUT-UBX_NAV2_TIMEGLO_SPI	0x20910539	U1	-	-	Output rate of the UBX-NAV2-TIMEGLO message on port SPI
CFG-MSGOUT-UBX_NAV2_TIMEGLO_UART1	0x20910536	U1	-	-	Output rate of the UBX-NAV2-TIMEGLO message on port UART1
CFG-MSGOUT-UBX_NAV2_TIMEGLO_UART2	0x20910537	U1	-	-	Output rate of the UBX-NAV2-TIMEGLO message on port UART2
CFG-MSGOUT-UBX_NAV2_TIMEGLO_USB	0x20910538	U1	-	-	Output rate of the UBX-NAV2-TIMEGLO message on port USB
CFG-MSGOUT-UBX_NAV2_TIMEGPS_I2C	0x20910540	U1	-	-	Output rate of the UBX-NAV2-TIMEGPS message on port I2C
CFG-MSGOUT-UBX_NAV2_TIMEGPS_SPI	0x20910544	U1	-	-	Output rate of the UBX-NAV2-TIMEGPS message on port SPI
CFG-MSGOUT-UBX_NAV2_TIMEGPS_UART1	0x20910541	U1	-	-	Output rate of the UBX-NAV2-TIMEGPS message on port UART1
CFG-MSGOUT-UBX_NAV2_TIMEGPS_UART2	0x20910542	U1	-	-	Output rate of the UBX-NAV2-TIMEGPS message on port UART2
CFG-MSGOUT-UBX_NAV2_TIMEGPS_USB	0x20910543	U1	-	-	Output rate of the UBX-NAV2-TIMEGPS message on port USB
CFG-MSGOUT-UBX_NAV2_TIMELS_I2C	0x20910545	U1	-	-	Output rate of the UBX-NAV2-TIMELS message on port I2C
CFG-MSGOUT-UBX_NAV2_TIMELS_SPI	0x20910549	U1	-	-	Output rate of the UBX-NAV2-TIMELS message on port SPI
CFG-MSGOUT-UBX_NAV2_TIMELS_UART1	0x20910546	U1	-	-	Output rate of the UBX-NAV2-TIMELS message on port UART1

Configuration item	Key ID	Type	Scale	Unit	Description
CFG-MSGOUT-UBX_NAV2_TIMELS_UART2	0x20910547	U1	-	-	Output rate of the UBX-NAV2-TIMELS message on port UART2
CFG-MSGOUT-UBX_NAV2_TIMELS_USB	0x20910548	U1	-	-	Output rate of the UBX-NAV2-TIMELS message on port USB
CFG-MSGOUT-UBX_NAV2_TIMEQZSS_I2C	0x20910575	U1	-	-	Output rate of the UBX-NAV2-TIMEQZSS message on port I2C
CFG-MSGOUT-UBX_NAV2_TIMEQZSS_SPI	0x20910579	U1	-	-	Output rate of the UBX-NAV2-TIMEQZSS message on port SPI
CFG-MSGOUT-UBX_NAV2_TIMEQZSS_UART1	0x20910576	U1	-	-	Output rate of the UBX-NAV2-TIMEQZSS message on port UART1
CFG-MSGOUT-UBX_NAV2_TIMEQZSS_UART2	0x20910577	U1	-	-	Output rate of the UBX-NAV2-TIMEQZSS message on port UART2
CFG-MSGOUT-UBX_NAV2_TIMEQZSS_USB	0x20910578	U1	-	-	Output rate of the UBX-NAV2-TIMEQZSS message on port USB
CFG-MSGOUT-UBX_NAV2_TIMEUTC_I2C	0x20910550	U1	-	-	Output rate of the UBX-NAV2-TIMEUTC message on port I2C
CFG-MSGOUT-UBX_NAV2_TIMEUTC_SPI	0x20910554	U1	-	-	Output rate of the UBX-NAV2-TIMEUTC message on port SPI
CFG-MSGOUT-UBX_NAV2_TIMEUTC_UART1	0x20910551	U1	-	-	Output rate of the UBX-NAV2-TIMEUTC message on port UART1
CFG-MSGOUT-UBX_NAV2_TIMEUTC_UART2	0x20910552	U1	-	-	Output rate of the UBX-NAV2-TIMEUTC message on port UART2
CFG-MSGOUT-UBX_NAV2_TIMEUTC_USB	0x20910553	U1	-	-	Output rate of the UBX-NAV2-TIMEUTC message on port USB
CFG-MSGOUT-UBX_NAV2_VELECEF_I2C	0x20910555	U1	-	-	Output rate of the UBX-NAV2-VELECEF message on port I2C
CFG-MSGOUT-UBX_NAV2_VELECEF_SPI	0x20910559	U1	-	-	Output rate of the UBX-NAV2-VELECEF message on port SPI
CFG-MSGOUT-UBX_NAV2_VELECEF_UART1	0x20910556	U1	-	-	Output rate of the UBX-NAV2-VELECEF message on port UART1
CFG-MSGOUT-UBX_NAV2_VELECEF_UART2	0x20910557	U1	-	-	Output rate of the UBX-NAV2-VELECEF message on port UART2
CFG-MSGOUT-UBX_NAV2_VELECEF_USB	0x20910558	U1	-	-	Output rate of the UBX-NAV2-VELECEF message on port USB
CFG-MSGOUT-UBX_NAV2_VELNED_I2C	0x20910560	U1	-	-	Output rate of the UBX-NAV2-VELNED message on port I2C
CFG-MSGOUT-UBX_NAV2_VELNED_SPI	0x20910564	U1	-	-	Output rate of the UBX-NAV2-VELNED message on port SPI
CFG-MSGOUT-UBX_NAV2_VELNED_UART1	0x20910561	U1	-	-	Output rate of the UBX-NAV2-VELNED message on port UART1
CFG-MSGOUT-UBX_NAV2_VELNED_UART2	0x20910562	U1	-	-	Output rate of the UBX-NAV2-VELNED message on port UART2
CFG-MSGOUT-UBX_NAV2_VELNED_USB	0x20910563	U1	-	-	Output rate of the UBX-NAV2-VELNED message on port USB
CFG-MSGOUT-UBX_NAV_CLOCK_I2C	0x20910065	U1	-	-	Output rate of the UBX-NAV-CLOCK message on port I2C
CFG-MSGOUT-UBX_NAV_CLOCK_SPI	0x20910069	U1	-	-	Output rate of the UBX-NAV-CLOCK message on port SPI
CFG-MSGOUT-UBX_NAV_CLOCK_UART1	0x20910066	U1	-	-	Output rate of the UBX-NAV-CLOCK message on port UART1
CFG-MSGOUT-UBX_NAV_CLOCK_UART2	0x20910067	U1	-	-	Output rate of the UBX-NAV-CLOCK message on port UART2

Configuration item	Key ID	Type	Scale	Unit	Description
CFG-MSGOUT-UBX_NAV_CLOCK_USB	0x20910068	U1	-	-	Output rate of the UBX-NAV-CLOCK message on port USB
CFG-MSGOUT-UBX_NAV_COV_I2C	0x20910083	U1	-	-	Output rate of the UBX-NAV-COV message on port I2C
CFG-MSGOUT-UBX_NAV_COV_SPI	0x20910087	U1	-	-	Output rate of the UBX-NAV-COV message on port SPI
CFG-MSGOUT-UBX_NAV_COV_UART1	0x20910084	U1	-	-	Output rate of the UBX-NAV-COV message on port UART1
CFG-MSGOUT-UBX_NAV_COV_UART2	0x20910085	U1	-	-	Output rate of the UBX-NAV-COV message on port UART2
CFG-MSGOUT-UBX_NAV_COV_USB	0x20910086	U1	-	-	Output rate of the UBX-NAV-COV message on port USB
CFG-MSGOUT-UBX_NAV_DOP_I2C	0x20910038	U1	-	-	Output rate of the UBX-NAV-DOP message on port I2C
CFG-MSGOUT-UBX_NAV_DOP_SPI	0x2091003c	U1	-	-	Output rate of the UBX-NAV-DOP message on port SPI
CFG-MSGOUT-UBX_NAV_DOP_UART1	0x20910039	U1	-	-	Output rate of the UBX-NAV-DOP message on port UART1
CFG-MSGOUT-UBX_NAV_DOP_UART2	0x2091003a	U1	-	-	Output rate of the UBX-NAV-DOP message on port UART2
CFG-MSGOUT-UBX_NAV_DOP_USB	0x2091003b	U1	-	-	Output rate of the UBX-NAV-DOP message on port USB
CFG-MSGOUT-UBX_NAV_EOE_I2C	0x2091015f	U1	-	-	Output rate of the UBX-NAV-EOE message on port I2C
CFG-MSGOUT-UBX_NAV_EOE_SPI	0x20910163	U1	-	-	Output rate of the UBX-NAV-EOE message on port SPI
CFG-MSGOUT-UBX_NAV_EOE_UART1	0x20910160	U1	-	-	Output rate of the UBX-NAV-EOE message on port UART1
CFG-MSGOUT-UBX_NAV_EOE_UART2	0x20910161	U1	-	-	Output rate of the UBX-NAV-EOE message on port UART2
CFG-MSGOUT-UBX_NAV_EOE_USB	0x20910162	U1	-	-	Output rate of the UBX-NAV-EOE message on port USB
CFG-MSGOUT-UBX_NAV_GEOFENCE_I2C	0x209100a1	U1	-	-	Output rate of the UBX-NAV-GEOFENCE message on port I2C
CFG-MSGOUT-UBX_NAV_GEOFENCE_SPI	0x209100a5	U1	-	-	Output rate of the UBX-NAV-GEOFENCE message on port SPI
CFG-MSGOUT-UBX_NAV_GEOFENCE_UART1	0x209100a2	U1	-	-	Output rate of the UBX-NAV-GEOFENCE message on port UART1
CFG-MSGOUT-UBX_NAV_GEOFENCE_UART2	0x209100a3	U1	-	-	Output rate of the UBX-NAV-GEOFENCE message on port UART2
CFG-MSGOUT-UBX_NAV_GEOFENCE_USB	0x209100a4	U1	-	-	Output rate of the UBX-NAV-GEOFENCE message on port USB
CFG-MSGOUT-UBX_NAV_HPPOSECEF_I2C	0x2091002e	U1	-	-	Output rate of the UBX-NAV-HPPOSECEF message on port I2C
CFG-MSGOUT-UBX_NAV_HPPOSECEF_SPI	0x20910032	U1	-	-	Output rate of the UBX-NAV-HPPOSECEF message on port SPI
CFG-MSGOUT-UBX_NAV_HPPOSECEF_UART1	0x2091002f	U1	-	-	Output rate of the UBX-NAV-HPPOSECEF message on port UART1
CFG-MSGOUT-UBX_NAV_HPPOSECEF_UART2	0x20910030	U1	-	-	Output rate of the UBX-NAV-HPPOSECEF message on port UART2
CFG-MSGOUT-UBX_NAV_HPPOSECEF_USB	0x20910031	U1	-	-	Output rate of the UBX-NAV-HPPOSECEF message on port USB

Configuration item	Key ID	Type	Scale	Unit	Description
CFG-MSGOUT-UBX_NAV_HPPOSLLH_I2C	`0x20910033`	U1	-	-	Output rate of the UBX-NAV-HPPOSLLH message on port I2C
CFG-MSGOUT-UBX_NAV_HPPOSLLH_SPI	`0x20910037`	U1	-	-	Output rate of the UBX-NAV-HPPOSLLH message on port SPI
CFG-MSGOUT-UBX_NAV_HPPOSLLH_UART1	`0x20910034`	U1	-	-	Output rate of the UBX-NAV-HPPOSLLH message on port UART1
CFG-MSGOUT-UBX_NAV_HPPOSLLH_UART2	`0x20910035`	U1	-	-	Output rate of the UBX-NAV-HPPOSLLH message on port UART2
CFG-MSGOUT-UBX_NAV_HPPOSLLH_USB	`0x20910036`	U1	-	-	Output rate of the UBX-NAV-HPPOSLLH message on port USB
CFG-MSGOUT-UBX_NAV_ODO_I2C	`0x2091007e`	U1	-	-	Output rate of the UBX-NAV-ODO message on port I2C
CFG-MSGOUT-UBX_NAV_ODO_SPI	`0x20910082`	U1	-	-	Output rate of the UBX-NAV-ODO message on port SPI
CFG-MSGOUT-UBX_NAV_ODO_UART1	`0x2091007f`	U1	-	-	Output rate of the UBX-NAV-ODO message on port UART1
CFG-MSGOUT-UBX_NAV_ODO_UART2	`0x20910080`	U1	-	-	Output rate of the UBX-NAV-ODO message on port UART2
CFG-MSGOUT-UBX_NAV_ODO_USB	`0x20910081`	U1	-	-	Output rate of the UBX-NAV-ODO message on port USB
CFG-MSGOUT-UBX_NAV_ORB_I2C	`0x20910010`	U1	-	-	Output rate of the UBX-NAV-ORB message on port I2C
CFG-MSGOUT-UBX_NAV_ORB_SPI	`0x20910014`	U1	-	-	Output rate of the UBX-NAV-ORB message on port SPI
CFG-MSGOUT-UBX_NAV_ORB_UART1	`0x20910011`	U1	-	-	Output rate of the UBX-NAV-ORB message on port UART1
CFG-MSGOUT-UBX_NAV_ORB_UART2	`0x20910012`	U1	-	-	Output rate of the UBX-NAV-ORB message on port UART2
CFG-MSGOUT-UBX_NAV_ORB_USB	`0x20910013`	U1	-	-	Output rate of the UBX-NAV-ORB message on port USB
CFG-MSGOUT-UBX_NAV_PL_I2C	`0x20910415`	U1	-	-	Output rate of the UBX-NAV-PL message on port I2C
CFG-MSGOUT-UBX_NAV_PL_SPI	`0x20910419`	U1	-	-	Output rate of the UBX-NAV-PL message on port SPI
CFG-MSGOUT-UBX_NAV_PL_UART1	`0x20910416`	U1	-	-	Output rate of the UBX-NAV-PL message on port UART1
CFG-MSGOUT-UBX_NAV_PL_UART2	`0x20910417`	U1	-	-	Output rate of the UBX-NAV-PL message on port UART2
CFG-MSGOUT-UBX_NAV_PL_USB	`0x20910418`	U1	-	-	Output rate of the UBX-NAV-PL message on port USB
CFG-MSGOUT-UBX_NAV_POSECEF_I2C	`0x20910024`	U1	-	-	Output rate of the UBX-NAV-POSECEF message on port I2C
CFG-MSGOUT-UBX_NAV_POSECEF_SPI	`0x20910028`	U1	-	-	Output rate of the UBX-NAV-POSECEF message on port SPI
CFG-MSGOUT-UBX_NAV_POSECEF_UART1	`0x20910025`	U1	-	-	Output rate of the UBX-NAV-POSECEF message on port UART1
CFG-MSGOUT-UBX_NAV_POSECEF_UART2	`0x20910026`	U1	-	-	Output rate of the UBX-NAV-POSECEF message on port UART2
CFG-MSGOUT-UBX_NAV_POSECEF_USB	`0x20910027`	U1	-	-	Output rate of the UBX-NAV-POSECEF message on port USB
CFG-MSGOUT-UBX_NAV_POSLLH_I2C	`0x20910029`	U1	-	-	Output rate of the UBX-NAV-POSLLH message on port I2C

Configuration item	Key ID	Type	Scale	Unit	Description
CFG-MSGOUT-UBX_NAV_POSLLH_SPI	0x2091002d	U1	-	-	Output rate of the UBX-NAV-POSLLH message on port SPI
CFG-MSGOUT-UBX_NAV_POSLLH_UART1	0x2091002a	U1	-	-	Output rate of the UBX-NAV-POSLLH message on port UART1
CFG-MSGOUT-UBX_NAV_POSLLH_UART2	0x2091002b	U1	-	-	Output rate of the UBX-NAV-POSLLH message on port UART2
CFG-MSGOUT-UBX_NAV_POSLLH_USB	0x2091002c	U1	-	-	Output rate of the UBX-NAV-POSLLH message on port USB
CFG-MSGOUT-UBX_NAV_PVT_I2C	0x20910006	U1	-	-	Output rate of the UBX-NAV-PVT message on port I2C
CFG-MSGOUT-UBX_NAV_PVT_SPI	0x2091000a	U1	-	-	Output rate of the UBX-NAV-PVT message on port SPI
CFG-MSGOUT-UBX_NAV_PVT_UART1	0x20910007	U1	-	-	Output rate of the UBX-NAV-PVT message on port UART1
CFG-MSGOUT-UBX_NAV_PVT_UART2	0x20910008	U1	-	-	Output rate of the UBX-NAV-PVT message on port UART2
CFG-MSGOUT-UBX_NAV_PVT_USB	0x20910009	U1	-	-	Output rate of the UBX-NAV-PVT message on port USB
CFG-MSGOUT-UBX_NAV_RELPOSNED_I2C	0x2091008d	U1	-	-	Output rate of the UBX-NAV-RELPOSNED message on port I2C
CFG-MSGOUT-UBX_NAV_RELPOSNED_SPI	0x20910091	U1	-	-	Output rate of the UBX-NAV-RELPOSNED message on port SPI
CFG-MSGOUT-UBX_NAV_RELPOSNED_UART1	0x2091008e	U1	-	-	Output rate of the UBX-NAV-RELPOSNED message on port UART1
CFG-MSGOUT-UBX_NAV_RELPOSNED_UART2	0x2091008f	U1	-	-	Output rate of the UBX-NAV-RELPOSNED message on port UART2
CFG-MSGOUT-UBX_NAV_RELPOSNED_USB	0x20910090	U1	-	-	Output rate of the UBX-NAV-RELPOSNED message on port USB
CFG-MSGOUT-UBX_NAV_SAT_I2C	0x20910015	U1	-	-	Output rate of the UBX-NAV-SAT message on port I2C
CFG-MSGOUT-UBX_NAV_SAT_SPI	0x20910019	U1	-	-	Output rate of the UBX-NAV-SAT message on port SPI
CFG-MSGOUT-UBX_NAV_SAT_UART1	0x20910016	U1	-	-	Output rate of the UBX-NAV-SAT message on port UART1
CFG-MSGOUT-UBX_NAV_SAT_UART2	0x20910017	U1	-	-	Output rate of the UBX-NAV-SAT message on port UART2
CFG-MSGOUT-UBX_NAV_SAT_USB	0x20910018	U1	-	-	Output rate of the UBX-NAV-SAT message on port USB
CFG-MSGOUT-UBX_NAV_SBAS_I2C	0x2091006a	U1	-	-	Output rate of the UBX-NAV-SBAS message on port I2C
CFG-MSGOUT-UBX_NAV_SBAS_SPI	0x2091006e	U1	-	-	Output rate of the UBX-NAV-SBAS message on port SPI
CFG-MSGOUT-UBX_NAV_SBAS_UART1	0x2091006b	U1	-	-	Output rate of the UBX-NAV-SBAS message on port UART1
CFG-MSGOUT-UBX_NAV_SBAS_UART2	0x2091006c	U1	-	-	Output rate of the UBX-NAV-SBAS message on port UART2
CFG-MSGOUT-UBX_NAV_SBAS_USB	0x2091006d	U1	-	-	Output rate of the UBX-NAV-SBAS message on port USB
CFG-MSGOUT-UBX_NAV_SIG_I2C	0x20910345	U1	-	-	Output rate of the UBX-NAV-SIG message on port I2C
CFG-MSGOUT-UBX_NAV_SIG_SPI	0x20910349	U1	-	-	Output rate of the UBX-NAV-SIG message on port SPI

Configuration item	Key ID	Type	Scale	Unit	Description
CFG-MSGOUT-UBX_NAV_SIG_UART1	0x20910346	U1	-	-	Output rate of the UBX-NAV-SIG message on port UART1
CFG-MSGOUT-UBX_NAV_SIG_UART2	0x20910347	U1	-	-	Output rate of the UBX-NAV-SIG message on port UART2
CFG-MSGOUT-UBX_NAV_SIG_USB	0x20910348	U1	-	-	Output rate of the UBX-NAV-SIG message on port USB
CFG-MSGOUT-UBX_NAV_SLAS_I2C	0x20910336	U1	-	-	Output rate of the UBX-NAV-SLAS message on port I2C
CFG-MSGOUT-UBX_NAV_SLAS_SPI	0x2091033a	U1	-	-	Output rate of the UBX-NAV-SLAS message on port SPI
CFG-MSGOUT-UBX_NAV_SLAS_UART1	0x20910337	U1	-	-	Output rate of the UBX-NAV-SLAS message on port UART1
CFG-MSGOUT-UBX_NAV_SLAS_UART2	0x20910338	U1	-	-	Output rate of the UBX-NAV-SLAS message on port UART2
CFG-MSGOUT-UBX_NAV_SLAS_USB	0x20910339	U1	-	-	Output rate of the UBX-NAV-SLAS message on port USB
CFG-MSGOUT-UBX_NAV_STATUS_I2C	0x2091001a	U1	-	-	Output rate of the UBX-NAV-STATUS message on port I2C
CFG-MSGOUT-UBX_NAV_STATUS_SPI	0x2091001e	U1	-	-	Output rate of the UBX-NAV-STATUS message on port SPI
CFG-MSGOUT-UBX_NAV_STATUS_UART1	0x2091001b	U1	-	-	Output rate of the UBX-NAV-STATUS message on port UART1
CFG-MSGOUT-UBX_NAV_STATUS_UART2	0x2091001c	U1	-	-	Output rate of the UBX-NAV-STATUS message on port UART2
CFG-MSGOUT-UBX_NAV_STATUS_USB	0x2091001d	U1	-	-	Output rate of the UBX-NAV-STATUS message on port USB
CFG-MSGOUT-UBX_NAV_SVIN_I2C	0x20910088	U1	-	-	Output rate of the UBX-NAV-SVIN message on port I2C
CFG-MSGOUT-UBX_NAV_SVIN_SPI	0x2091008c	U1	-	-	Output rate of the UBX-NAV-SVIN message on port SPI
CFG-MSGOUT-UBX_NAV_SVIN_UART1	0x20910089	U1	-	-	Output rate of the UBX-NAV-SVIN message on port UART1
CFG-MSGOUT-UBX_NAV_SVIN_UART2	0x2091008a	U1	-	-	Output rate of the UBX-NAV-SVIN message on port UART2
CFG-MSGOUT-UBX_NAV_SVIN_USB	0x2091008b	U1	-	-	Output rate of the UBX-NAV-SVIN message on port USB
CFG-MSGOUT-UBX_NAV_TIMEBDS_I2C	0x20910051	U1	-	-	Output rate of the UBX-NAV-TIMEBDS message on port I2C
CFG-MSGOUT-UBX_NAV_TIMEBDS_SPI	0x20910055	U1	-	-	Output rate of the UBX-NAV-TIMEBDS message on port SPI
CFG-MSGOUT-UBX_NAV_TIMEBDS_UART1	0x20910052	U1	-	-	Output rate of the UBX-NAV-TIMEBDS message on port UART1
CFG-MSGOUT-UBX_NAV_TIMEBDS_UART2	0x20910053	U1	-	-	Output rate of the UBX-NAV-TIMEBDS message on port UART2
CFG-MSGOUT-UBX_NAV_TIMEBDS_USB	0x20910054	U1	-	-	Output rate of the UBX-NAV-TIMEBDS message on port USB
CFG-MSGOUT-UBX_NAV_TIMEGAL_I2C	0x20910056	U1	-	-	Output rate of the UBX-NAV-TIMEGAL message on port I2C
CFG-MSGOUT-UBX_NAV_TIMEGAL_SPI	0x2091005a	U1	-	-	Output rate of the UBX-NAV-TIMEGAL message on port SPI
CFG-MSGOUT-UBX_NAV_TIMEGAL_UART1	0x20910057	U1	-	-	Output rate of the UBX-NAV-TIMEGAL message on port UART1

Configuration item	Key ID	Type	Scale	Unit	Description
CFG-MSGOUT-UBX_NAV_TIMEGAL_UART2	0x20910058	U1	-	-	Output rate of the UBX-NAV-TIMEGAL message on port UART2
CFG-MSGOUT-UBX_NAV_TIMEGAL_USB	0x20910059	U1	-	-	Output rate of the UBX-NAV-TIMEGAL message on port USB
CFG-MSGOUT-UBX_NAV_TIMEGLO_I2C	0x2091004c	U1	-	-	Output rate of the UBX-NAV-TIMEGLO message on port I2C
CFG-MSGOUT-UBX_NAV_TIMEGLO_SPI	0x20910050	U1	-	-	Output rate of the UBX-NAV-TIMEGLO message on port SPI
CFG-MSGOUT-UBX_NAV_TIMEGLO_UART1	0x2091004d	U1	-	-	Output rate of the UBX-NAV-TIMEGLO message on port UART1
CFG-MSGOUT-UBX_NAV_TIMEGLO_UART2	0x2091004e	U1	-	-	Output rate of the UBX-NAV-TIMEGLO message on port UART2
CFG-MSGOUT-UBX_NAV_TIMEGLO_USB	0x2091004f	U1	-	-	Output rate of the UBX-NAV-TIMEGLO message on port USB
CFG-MSGOUT-UBX_NAV_TIMEGPS_I2C	0x20910047	U1	-	-	Output rate of the UBX-NAV-TIMEGPS message on port I2C
CFG-MSGOUT-UBX_NAV_TIMEGPS_SPI	0x2091004b	U1	-	-	Output rate of the UBX-NAV-TIMEGPS message on port SPI
CFG-MSGOUT-UBX_NAV_TIMEGPS_UART1	0x20910048	U1	-	-	Output rate of the UBX-NAV-TIMEGPS message on port UART1
CFG-MSGOUT-UBX_NAV_TIMEGPS_UART2	0x20910049	U1	-	-	Output rate of the UBX-NAV-TIMEGPS message on port UART2
CFG-MSGOUT-UBX_NAV_TIMEGPS_USB	0x2091004a	U1	-	-	Output rate of the UBX-NAV-TIMEGPS message on port USB
CFG-MSGOUT-UBX_NAV_TIMELS_I2C	0x20910060	U1	-	-	Output rate of the UBX-NAV-TIMELS message on port I2C
CFG-MSGOUT-UBX_NAV_TIMELS_SPI	0x20910064	U1	-	-	Output rate of the UBX-NAV-TIMELS message on port SPI
CFG-MSGOUT-UBX_NAV_TIMELS_UART1	0x20910061	U1	-	-	Output rate of the UBX-NAV-TIMELS message on port UART1
CFG-MSGOUT-UBX_NAV_TIMELS_UART2	0x20910062	U1	-	-	Output rate of the UBX-NAV-TIMELS message on port UART2
CFG-MSGOUT-UBX_NAV_TIMELS_USB	0x20910063	U1	-	-	Output rate of the UBX-NAV-TIMELS message on port USB
CFG-MSGOUT-UBX_NAV_TIMEQZSS_I2C	0x20910386	U1	-	-	Output rate of the UBX-NAV-TIMEQZSS message on port I2C
CFG-MSGOUT-UBX_NAV_TIMEQZSS_SPI	0x2091038a	U1	-	-	Output rate of the UBX-NAV-TIMEQZSS message on port SPI
CFG-MSGOUT-UBX_NAV_TIMEQZSS_UART1	0x20910387	U1	-	-	Output rate of the UBX-NAV-TIMEQZSS message on port UART1
CFG-MSGOUT-UBX_NAV_TIMEQZSS_UART2	0x20910388	U1	-	-	Output rate of the UBX-NAV-TIMEQZSS message on port UART2
CFG-MSGOUT-UBX_NAV_TIMEQZSS_USB	0x20910389	U1	-	-	Output rate of the UBX-NAV-TIMEQZSS message on port USB
CFG-MSGOUT-UBX_NAV_TIMEUTC_I2C	0x2091005b	U1	-	-	Output rate of the UBX-NAV-TIMEUTC message on port I2C
CFG-MSGOUT-UBX_NAV_TIMEUTC_SPI	0x2091005f	U1	-	-	Output rate of the UBX-NAV-TIMEUTC message on port SPI
CFG-MSGOUT-UBX_NAV_TIMEUTC_UART1	0x2091005c	U1	-	-	Output rate of the UBX-NAV-TIMEUTC message on port UART1
CFG-MSGOUT-UBX_NAV_TIMEUTC_UART2	0x2091005d	U1	-	-	Output rate of the UBX-NAV-TIMEUTC message on port UART2

Configuration item	Key ID	Type	Scale	Unit	Description
CFG-MSGOUT-UBX_NAV_TIMEUTC_USB	0x2091005e	U1	-	-	Output rate of the UBX-NAV-TIMEUTC message on port USB
CFG-MSGOUT-UBX_NAV_VELECEF_I2C	0x2091003d	U1	-	-	Output rate of the UBX-NAV-VELECEF message on port I2C
CFG-MSGOUT-UBX_NAV_VELECEF_SPI	0x20910041	U1	-	-	Output rate of the UBX-NAV-VELECEF message on port SPI
CFG-MSGOUT-UBX_NAV_VELECEF_UART1	0x2091003e	U1	-	-	Output rate of the UBX-NAV-VELECEF message on port UART1
CFG-MSGOUT-UBX_NAV_VELECEF_UART2	0x2091003f	U1	-	-	Output rate of the UBX-NAV-VELECEF message on port UART2
CFG-MSGOUT-UBX_NAV_VELECEF_USB	0x20910040	U1	-	-	Output rate of the UBX-NAV-VELECEF message on port USB
CFG-MSGOUT-UBX_NAV_VELNED_I2C	0x20910042	U1	-	-	Output rate of the UBX-NAV-VELNED message on port I2C
CFG-MSGOUT-UBX_NAV_VELNED_SPI	0x20910046	U1	-	-	Output rate of the UBX-NAV-VELNED message on port SPI
CFG-MSGOUT-UBX_NAV_VELNED_UART1	0x20910043	U1	-	-	Output rate of the UBX-NAV-VELNED message on port UART1
CFG-MSGOUT-UBX_NAV_VELNED_UART2	0x20910044	U1	-	-	Output rate of the UBX-NAV-VELNED message on port UART2
CFG-MSGOUT-UBX_NAV_VELNED_USB	0x20910045	U1	-	-	Output rate of the UBX-NAV-VELNED message on port USB
CFG-MSGOUT-UBX_RXM_COR_I2C	0x209106b6	U1	-	-	Output rate of the UBX-RXM-COR message on port I2C
CFG-MSGOUT-UBX_RXM_COR_SPI	0x209106ba	U1	-	-	Output rate of the UBX-RXM-COR message on port SPI
CFG-MSGOUT-UBX_RXM_COR_UART1	0x209106b7	U1	-	-	Output rate of the UBX-RXM-COR message on port UART1
CFG-MSGOUT-UBX_RXM_COR_UART2	0x209106b8	U1	-	-	Output rate of the UBX-RXM-COR message on port UART2
CFG-MSGOUT-UBX_RXM_COR_USB	0x209106b9	U1	-	-	Output rate of the UBX-RXM-COR message on port USB
CFG-MSGOUT-UBX_RXM_MEASX_I2C	0x20910204	U1	-	-	Output rate of the UBX-RXM-MEASX message on port I2C
CFG-MSGOUT-UBX_RXM_MEASX_SPI	0x20910208	U1	-	-	Output rate of the UBX-RXM-MEASX message on port SPI
CFG-MSGOUT-UBX_RXM_MEASX_UART1	0x20910205	U1	-	-	Output rate of the UBX-RXM-MEASX message on port UART1
CFG-MSGOUT-UBX_RXM_MEASX_UART2	0x20910206	U1	-	-	Output rate of the UBX-RXM-MEASX message on port UART2
CFG-MSGOUT-UBX_RXM_MEASX_USB	0x20910207	U1	-	-	Output rate of the UBX-RXM-MEASX message on port USB
CFG-MSGOUT-UBX_RXM_RAWX_I2C	0x209102a4	U1	-	-	Output rate of the UBX-RXM-RAWX message on port I2C
CFG-MSGOUT-UBX_RXM_RAWX_SPI	0x209102a8	U1	-	-	Output rate of the UBX-RXM-RAWX message on port SPI
CFG-MSGOUT-UBX_RXM_RAWX_UART1	0x209102a5	U1	-	-	Output rate of the UBX-RXM-RAWX message on port UART1
CFG-MSGOUT-UBX_RXM_RAWX_UART2	0x209102a6	U1	-	-	Output rate of the UBX-RXM-RAWX message on port UART2
CFG-MSGOUT-UBX_RXM_RAWX_USB	0x209102a7	U1	-	-	Output rate of the UBX-RXM-RAWX message on port USB

Configuration item	Key ID	Type	Scale	Unit	Description
CFG-MSGOUT-UBX_RXM_RLM_I2C	0x2091025e	U1	-	-	Output rate of the UBX-RXM-RLM message on port I2C
CFG-MSGOUT-UBX_RXM_RLM_SPI	0x20910262	U1	-	-	Output rate of the UBX-RXM-RLM message on port SPI
CFG-MSGOUT-UBX_RXM_RLM_UART1	0x2091025f	U1	-	-	Output rate of the UBX-RXM-RLM message on port UART1
CFG-MSGOUT-UBX_RXM_RLM_UART2	0x20910260	U1	-	-	Output rate of the UBX-RXM-RLM message on port UART2
CFG-MSGOUT-UBX_RXM_RLM_USB	0x20910261	U1	-	-	Output rate of the UBX-RXM-RLM message on port USB
CFG-MSGOUT-UBX_RXM_RTCM_I2C	0x20910268	U1	-	-	Output rate of the UBX-RXM-RTCM message on port I2C
CFG-MSGOUT-UBX_RXM_RTCM_SPI	0x2091026c	U1	-	-	Output rate of the UBX-RXM-RTCM message on port SPI
CFG-MSGOUT-UBX_RXM_RTCM_UART1	0x20910269	U1	-	-	Output rate of the UBX-RXM-RTCM message on port UART1
CFG-MSGOUT-UBX_RXM_RTCM_UART2	0x2091026a	U1	-	-	Output rate of the UBX-RXM-RTCM message on port UART2
CFG-MSGOUT-UBX_RXM_RTCM_USB	0x2091026b	U1	-	-	Output rate of the UBX-RXM-RTCM message on port USB
CFG-MSGOUT-UBX_RXM_SFRBX_I2C	0x20910231	U1	-	-	Output rate of the UBX-RXM-SFRBX message on port I2C
CFG-MSGOUT-UBX_RXM_SFRBX_SPI	0x20910235	U1	-	-	Output rate of the UBX-RXM-SFRBX message on port SPI
CFG-MSGOUT-UBX_RXM_SFRBX_UART1	0x20910232	U1	-	-	Output rate of the UBX-RXM-SFRBX message on port UART1
CFG-MSGOUT-UBX_RXM_SFRBX_UART2	0x20910233	U1	-	-	Output rate of the UBX-RXM-SFRBX message on port UART2
CFG-MSGOUT-UBX_RXM_SFRBX_USB	0x20910234	U1	-	-	Output rate of the UBX-RXM-SFRBX message on port USB
CFG-MSGOUT-UBX_RXM_SPARTN_I2C	0x20910605	U1	-	-	Output rate of the UBX-RXM-SPARTN message on port I2C
CFG-MSGOUT-UBX_RXM_SPARTN_SPI	0x20910609	U1	-	-	Output rate of the UBX-RXM-SPARTN message on port SPI
CFG-MSGOUT-UBX_RXM_SPARTN_UART1	0x20910606	U1	-	-	Output rate of the UBX-RXM-SPARTN message on port UART1
CFG-MSGOUT-UBX_RXM_SPARTN_UART2	0x20910607	U1	-	-	Output rate of the UBX-RXM-SPARTN message on port UART2
CFG-MSGOUT-UBX_RXM_SPARTN_USB	0x20910608	U1	-	-	Output rate of the UBX-RXM-SPARTN message on port USB
CFG-MSGOUT-UBX_TIM_TM2_I2C	0x20910178	U1	-	-	Output rate of the UBX-TIM-TM2 message on port I2C
CFG-MSGOUT-UBX_TIM_TM2_SPI	0x2091017c	U1	-	-	Output rate of the UBX-TIM-TM2 message on port SPI
CFG-MSGOUT-UBX_TIM_TM2_UART1	0x20910179	U1	-	-	Output rate of the UBX-TIM-TM2 message on port UART1
CFG-MSGOUT-UBX_TIM_TM2_UART2	0x2091017a	U1	-	-	Output rate of the UBX-TIM-TM2 message on port UART2
CFG-MSGOUT-UBX_TIM_TM2_USB	0x2091017b	U1	-	-	Output rate of the UBX-TIM-TM2 message on port USB
CFG-MSGOUT-UBX_TIM_TP_I2C	0x2091017d	U1	-	-	Output rate of the UBX-TIM-TP message on port I2C

Configuration item	Key ID	Type	Scale	Unit	Description
CFG-MSGOUT-UBX_TIM_TP_SPI	0x20910181	U1	-	-	Output rate of the UBX-TIM-TP message on port SPI
CFG-MSGOUT-UBX_TIM_TP_UART1	0x2091017e	U1	-	-	Output rate of the UBX-TIM-TP message on port UART1
CFG-MSGOUT-UBX_TIM_TP_UART2	0x2091017f	U1	-	-	Output rate of the UBX-TIM-TP message on port UART2
CFG-MSGOUT-UBX_TIM_TP_USB	0x20910180	U1	-	-	Output rate of the UBX-TIM-TP message on port USB
CFG-MSGOUT-UBX_TIM_VRFY_I2C	0x20910092	U1	-	-	Output rate of the UBX-TIM-VRFY message on port I2C
CFG-MSGOUT-UBX_TIM_VRFY_SPI	0x20910096	U1	-	-	Output rate of the UBX-TIM-VRFY message on port SPI
CFG-MSGOUT-UBX_TIM_VRFY_UART1	0x20910093	U1	-	-	Output rate of the UBX-TIM-VRFY message on port UART1
CFG-MSGOUT-UBX_TIM_VRFY_UART2	0x20910094	U1	-	-	Output rate of the UBX-TIM-VRFY message on port UART2
CFG-MSGOUT-UBX_TIM_VRFY_USB	0x20910095	U1	-	-	Output rate of the UBX-TIM-VRFY message on port USB

Table 16: CFG-MSGOUT configuration items

6.9.12 CFG-NAV2: Secondary output configuration

This group contains configuration items related to the secondary (NAV2) output.

Configuration item	Key ID	Type	Scale	Unit	Description
CFG-NAV2-OUT_ENABLED	0x10170001	L	-	-	Enable secondary (NAV2) output
Enables the secondary output (GNSS standalone output). It can be used simultaneously with the available primary output (high precision, sensor fusion or time mode output).					
CFG-NAV2-SBAS_USE_INTEGRITY	0x10170002	L	-	-	Use SBAS integrity information in the secondary output
If enabled, the receiver will only use GPS satellites for which integrity information is available. This configuration item allows configuring the SBAS integrity feature differently for the primary output and the secondary output. For configuring the primary output, see CFG-SBAS-USE_INTEGRITY.					

Table 17: CFG-NAV2 configuration items

6.9.13 CFG-NAVHPG: High precision navigation configuration

This group configures items related to the operation of the receiver in high precision, for example Differential correction and other related features.

Configuration item	Key ID	Type	Scale	Unit	Description
CFG-NAVHPG-DGNSSMODE	0x20140011	E1	-	-	Differential corrections mode
See Table 19 below for a list of possible constants for this item.					

Table 18: CFG-NAVHPG configuration items

Constant	Value	Description
RTK_FLOAT	2	No attempts made to fix ambiguities

Constant	Value	Description
RTK_FIXED	3	Ambiguities are fixed whenever possible

Table 19: Constants for CFG-NAVHPG-DGNSSMODE

6.9.14 CFG-NAVSPG: Standard precision navigation configuration

This group contains configuration items related to the operation of the receiver at standard precision, including configuring postition fix mode, ionospheric model selection and other related items.

Configuration item	Key ID	Type	Scale	Unit	Description
CFG-NAVSPG-FIXMODE	0x20110011	E1	-	-	Position fix mode
See Table 21 below for a list of possible constants for this item.					
CFG-NAVSPG-INIFIX3D	0x10110013	L	-	-	Initial fix must be a 3D fix
CFG-NAVSPG-WKNROLLOVER	0x30110017	U2	-	-	GPS week rollover number
GPS week numbers will be set correctly from this week up to 1024 weeks after this week. Range is from 1 to 4096.					
CFG-NAVSPG-UTCSTANDARD	0x2011001c	E1	-	-	UTC standard to be used
See section GNSS time base in the integration manual. See Table 22 below for a list of possible constants for this item.					
CFG-NAVSPG-DYNMODEL	0x20110021	E1	-	-	Dynamic platform model
See Table 23 below for a list of possible constants for this item.					
CFG-NAVSPG-ACKAIDING	0x10110025	L	-	-	Acknowledge assistance input messages
CFG-NAVSPG-USE_USRDAT	0x10110061	L	-	-	Use user geodetic datum parameters
This must be set together with all CFG-NAVSPG-USERDAT_* parameters.					
CFG-NAVSPG-USRDAT_MAJA	0x50110062	R8	-	m	Geodetic datum semi-major axis
Accepted range is from 6,300,000.0 to 6,500,000.0 meters This will only be used if CFG-NAVSPG-USE_USERDAT is set. It must be set together with all other CFG-NAVSPG-USERDAT_... parameters.					
CFG-NAVSPG-USRDAT_FLAT	0x50110063	R8	-	-	Geodetic datum 1.0 / flattening
Accepted range is 0.0 to 500.0. This will only be used if CFG-NAVSPG-USE_USERDAT is set. It must be set together with all other CFG-NAVSPG-USERDAT_... parameters.					
CFG-NAVSPG-USRDAT_DX	0x40110064	R4	-	m	Geodetic datum X axis shift at the origin
Accepted range is +/- 5000.0 meters. This will only be used if CFG-NAVSPG-USE_USERDAT is set. It must be set together with all other CFG-NAVSPG-USERDAT_... parameters.					
CFG-NAVSPG-USRDAT_DY	0x40110065	R4	-	m	Geodetic datum Y axis shift at the origin
Accepted range is +/- 5000.0 meters. This will only be used if CFG-NAVSPG-USE_USERDAT is set. It must be set together with all other CFG-NAVSPG-USERDAT_... parameters.					
CFG-NAVSPG-USRDAT_DZ	0x40110066	R4	-	m	Geodetic datum Z axis shift at the origin
Accepted range is +/- 5000.0 meters. This will only be used if CFG-NAVSPG-USE_USERDAT is set. It must be set together with all other CFG-NAVSPG-USERDAT_.. parameters.					
CFG-NAVSPG-USRDAT_ROTX	0x40110067	R4	-	arcsec	Geodetic datum rotation about the X axis
Accepted range is +/- 20.0 milli arc seconds. This will only be used if CFG-NAVSPG-USE_USERDAT is set. It must be set together with all other CFG-NAVSPG-USERDAT_... parameters.					

Configuration item	Key ID	Type	Scale	Unit	Description
CFG-NAVSPG-USRDAT_ROTY	0x40110068	R4	-	arcsec	Geodetic datum rotation about the Y axis ()
Accepted range is +/- 20.0 milli-arc seconds. This will only be used if CFG-NAVSPG-USE_USERDAT is set. It must be set together with all other CFG-NAVSPG-USERDAT_* parameters.					
CFG-NAVSPG-USRDAT_ROTZ	0x40110069	R4	-	arcsec	Geodetic datum rotation about the Z axis
Accepted range is +/- 20.0 milli-arc seconds. This will only be used if CFG-NAVSPG-USE_USERDAT is set. It must be set together with all other CFG-NAVSPG-USERDAT_... parameters.					
CFG-NAVSPG-USRDAT_SCALE	0x4011006a	R4	-	ppm	Geodetic datum scale factor
Accepted range is 0.0 to 50.0 parts per million. This will only be used if CFG-NAVSPG-USE_USERDAT is set. It must be set together with all other CFG-NAVSPG-USERDAT_... parameters.					
CFG-NAVSPG-INFIL_MINSVS	0x201100a1	U1	-	-	Minimum number of satellites for navigation
CFG-NAVSPG-INFIL_MAXSVS	0x201100a2	U1	-	-	Maximum number of satellites for navigation
CFG-NAVSPG-INFIL_MINCNO	0x201100a3	U1	-	dBHz	Minimum satellite signal level for navigation
CFG-NAVSPG-INFIL_MINELEV	0x201100a4	I1	-	deg	Minimum elevation for a GNSS satellite to be used in navigation
CFG-NAVSPG-INFIL_NCNOTHRS	0x201100aa	U1	-	-	Number of satellites required to have C/N0 above CFG-NAVSPG-INFIL_CNOTHRS for a fix to be attempted
CFG-NAVSPG-INFIL_CNOTHRS	0x201100ab	U1	-	-	C/N0 threshold for deciding whether to attempt a fix
CFG-NAVSPG-OUTFIL_PDOP	0x301100b1	U2	0.1	-	Output filter position DOP mask (threshold)
CFG-NAVSPG-OUTFIL_TDOP	0x301100b2	U2	0.1	-	Output filter time DOP mask (threshold)
CFG-NAVSPG-OUTFIL_PACC	0x301100b3	U2	-	m	Output filter position accuracy mask (threshold)
CFG-NAVSPG-OUTFIL_TACC	0x301100b4	U2	-	m	Output filter time accuracy mask (threshold)
CFG-NAVSPG-OUTFIL_FACC	0x301100b5	U2	0.01	m/s	Output filter frequency accuracy mask (threshold)
CFG-NAVSPG-CONSTR_ALT	0x401100c1	I4	0.01	m	Fixed altitude (mean sea level) for 2D fix mode
CFG-NAVSPG-CONSTR_ALTVAR	0x401100c2	U4	0.0001	m^2	Fixed altitude variance for 2D mode
CFG-NAVSPG-CONSTR_DGNSSTO	0x201100c4	U1	-	s	DGNSS timeout
CFG-NAVSPG-PL_ENA	0x101100d7	L	-	-	Enable Protection level
If enabled, protection level computing will be on.					

Table 20: CFG-NAVSPG configuration items

Constant	Value	Description
2DONLY	1	2D only
3DONLY	2	3D only
AUTO	3	Auto 2D/3D

Table 21: Constants for CFG-NAVSPG-FIXMODE

Constant	Value	Description
AUTO	0	Automatic; receiver selects based on GNSS configuration
USNO	3	UTC as operated by the U.S. Naval Observatory (USNO); derived from GPS time
EU	5	UTC as combined from multiple European laboratories; derived from Galileo time

Constant	Value	Description
SU	6	UTC as operated by the former Soviet Union (SU); derived from GLONASS time
NTSC	7	UTC as operated by the National Time Service Center (NTSC), China; derived from BeiDou time
NPLI	8	UTC as operated by the National Physics Laboratory, India (NPLI); derived from NavIC time

Table 22: Constants for CFG-NAVSPG-UTCSTANDARD

Constant	Value	Description
PORT	0	Portable
STAT	2	Stationary
PED	3	Pedestrian
AUTOMOT	4	Automotive
SEA	5	Sea
AIR1	6	Airborne with <1g acceleration
AIR2	7	Airborne with <2g acceleration
AIR4	8	Airborne with <4g acceleration
WRIST	9	Wrist-worn watch (not available in all products)
BIKE	10	Motorbike (not available in all products)
MOWER	11	Robotic lawn mower (not available in all products)
ESCOOTER	12	E-scooter (not available in all products)

Table 23: Constants for CFG-NAVSPG-DYNMODEL

6.9.15 CFG-NMEA: NMEA protocol configuration

This group configures the NMEA protocol. See section NMEA protocol configuration for a detailed description of the configuration effects on NMEA output.

Configuration item	Key ID	Type	Scale	Unit	Description
CFG-NMEA-PROTVER	0x20930001	E1	-	-	NMEA protocol version
See Table 25 below for a list of possible constants for this item.					
CFG-NMEA-MAXSVS	0x20930002	E1	-	-	Maximum number of SVs to report per Talker ID
See Table 26 below for a list of possible constants for this item.					
CFG-NMEA-COMPAT	0x10930003	L	-	-	Enable compatibility mode
This might be needed for certain applications, e.g. for an NMEA parser that expects a fixed number of digits in position coordinates.					
CFG-NMEA-CONSIDER	0x10930004	L	-	-	Enable considering mode
This will affect NMEA output used satellite count. If set, also considered satellites (e.g. RAIMED) are counted as used satellites as well.					
CFG-NMEA-LIMIT82	0x10930005	L	-	-	Enable strict limit to 82 characters maximum NMEA message length
CFG-NMEA-HIGHPREC	0x10930006	L	-	-	Enable high precision mode
This flag cannot be set in conjunction with either CFG-NMEA-COMPAT or CFG-NMEA-LIMIT82 mode.					
CFG-NMEA-SVNUMBERING	0x20930007	E1	-	-	Display configuration for SVs that do not have value defined in NMEA

Configuration item	Key ID	Type	Scale	Unit	Description
Configures the display of satellites that do not have an NMEA-defined value. Note: this does not apply to satellites with an unknown ID. See also Satellite Numbering. See Table 27 below for a list of possible constants for this item.					
CFG-NMEA-FILT_GPS	0x10930011	L	-	-	Disable reporting of GPS satellites
CFG-NMEA-FILT_SBAS	0x10930012	L	-	-	Disable reporting of SBAS satellites
CFG-NMEA-FILT_GAL	0x10930013	L	-	-	Disable reporting of Galileo satellites
CFG-NMEA-FILT_QZSS	0x10930015	L	-	-	Disable reporting of QZSS satellites
CFG-NMEA-FILT_GLO	0x10930016	L	-	-	Disable reporting of GLONASS satellites
CFG-NMEA-FILT_BDS	0x10930017	L	-	-	Disable reporting of BeiDou satellites
CFG-NMEA-OUT_INVFIX	0x10930021	L	-	-	Enable position output for failed or invalid fixes
CFG-NMEA-OUT_MSKFIX	0x10930022	L	-	-	Enable position output for invalid fixes
CFG-NMEA-OUT_INVTIME	0x10930023	L	-	-	Enable time output for invalid times
CFG-NMEA-OUT_INVDATE	0x10930024	L	-	-	Enable date output for invalid dates
CFG-NMEA-OUT_ONLYGPS	0x10930025	L	-	-	Restrict output to GPS satellites only
CFG-NMEA-OUT_FROZENCOG	0x10930026	L	-	-	Enable course over ground output even if it is frozen
CFG-NMEA-MAINTALKERID	0x20930031	E1	-	-	Main Talker ID
By default the main Talker ID (i.e. the Talker ID used for all messages other than GSV) is determined by the GNSS assignment of the receiver's channels (see CFG-SIGNAL). This field enables the main Talker ID to be overridden. See Table 28 below for a list of possible constants for this item.					
CFG-NMEA-GSVTALKERID	0x20930032	E1	-	-	Talker ID for GSV NMEA messages
By default the Talker ID for GSV messages is GNSS-specific (as defined by NMEA). This field enables the GSV Talker ID to be overridden. See Table 29 below for a list of possible constants for this item.					
CFG-NMEA-BDSTALKERID	0x30930033	U2	-	-	BeiDou Talker ID
Sets the two ASCII characters that should be used for the BeiDou Talker ID. If these are set to zero, the default BeiDou Talker ID will be used.					

Table 24: CFG-NMEA configuration items

Constant	Value	Description
V21	21	NMEA protocol version 2.1
V23	23	NMEA protocol version 2.3
V40	40	NMEA protocol version 4.0 (not available in all products)
V41	41	NMEA protocol version 4.10 (not available in all products)
V411	42	NMEA protocol version 4.11 (not available in all products)

Table 25: Constants for CFG-NMEA-PROTVER

Constant	Value	Description
UNLIM	0	Unlimited
8SVS	8	8 SVs
12SVS	12	12 SVs
16SVS	16	16 SVs

Table 26: Constants for CFG-NMEA-MAXSVS

Constant	Value	Description
STRICT	0	Strict - satellites are not output
EXTENDED	1	Extended - use proprietary numbering

Table 27: Constants for CFG-NMEA-SVNUMBERING

Constant	Value	Description
AUTO	0	Main Talker ID is not overridden
GP	1	Set main Talker ID to 'GP'
GL	2	Set main Talker ID to 'GL'
GN	3	Set main Talker ID to 'GN'
GA	4	Set main Talker ID to 'GA' (not available in all products)
GB	5	Set main Talker ID to 'GB' (not available in all products)
GQ	7	Set main Talker ID to 'GQ' (not available in all products)

Table 28: Constants for CFG-NMEA-MAINTALKERID

Constant	Value	Description
GNSS	0	Use GNSS-specific Talker ID (as defined by NMEA)
MAIN	1	Use the main Talker ID

Table 29: Constants for CFG-NMEA-GSVTALKERID

6.9.16 CFG-ODO: Odometer and low-speed course over ground filter configuration

The items in this group allow the user to configure the Odometer feature and Low-Speed Course Over Ground Filter.

Configuration item	Key ID	Type	Scale	Unit	Description
CFG-ODO-USE_ODO	0x10220001	L	-	-	Use odometer
CFG-ODO-USE_COG	0x10220002	L	-	-	Use low-speed course over ground filter
CFG-ODO-OUTLPVEL	0x10220003	L	-	-	Output low-pass filtered velocity
CFG-ODO-OUTLPCOG	0x10220004	L	-	-	Output low-pass filtered course over ground (heading)
CFG-ODO-PROFILE	0x20220005	E1	-	-	Odometer profile configuration
See Table 31 below for a list of possible constants for this item.					
CFG-ODO-COGMAXSPEED	0x20220021	U1	-	m/s	Upper speed limit for low-speed course over ground filter
CFG-ODO-COGMAXPOSACC	0x20220022	U1	-	-	Maximum acceptable position accuracy for computing low-speed filtered course over ground
CFG-ODO-VELLPGAIN	0x20220031	U1	-	-	Velocity low-pass filter level
Range is from 0 to 255.					
CFG-ODO-COGLPGAIN	0x20220032	U1	-	-	Course over ground low-pass filter level (at speed < 8 m/s)
Range is from 0 to 255.					

Table 30: CFG-ODO configuration items

Constant	Value	Description
RUN	0	Running
CYCL	1	Cycling

Constant	Value	Description
SWIM	2	Swimming
CAR	3	Car
CUSTOM	4	Custom

Table 31: Constants for CFG-ODO-PROFILE

6.9.17 CFG-QZSS: QZSS system configuration

Note that enabling and disabling of individual GNSS is done via the CFG-SIGNAL configuration group.

Configuration item	Key ID	Type	Scale	Unit	Description
CFG-QZSS-USE_SLAS_DGNSS	`0x10370005`	L	-	-	Apply QZSS SLAS DGNSS corrections
CFG-QZSS-USE_SLAS_TESTMODE	`0x10370006`	L	-	-	Use QZSS SLAS data when it is in test mode (SLAS msg 0)
CFG-QZSS-USE_SLAS_RAIM_UNCORR	`0x10370007`	L	-	-	Raim out measurements that are not corrected by QZSS SLAS, if at least 5 measurements are corrected
CFG-QZSS-SLAS_MAX_BASELINE	`0x30370008`	U2	-	km	Maximum baseline distance to closest GMS
SLAS corrections are only applied if the receiver is at most this far away from the closest ground monitoring station (GMS). Note that due to the nature of the service, the usefulness of corrections degrades with distance. When far away from GMS, SBAS may be a better correction source.					

Table 32: CFG-QZSS configuration items

6.9.18 CFG-RATE: Navigation and measurement rate configuration

The configuration items in this group allow the user to alter the rate at which navigation solutions (and the measurements that they depend on) are generated by the receiver. The calculation of the navigation solution will always be aligned to the top of a second zero (first second of the week) of the configured reference time system. The navigation period is an integer multiple of the measurement period.

Configuration item	Key ID	Type	Scale	Unit	Description
CFG-RATE-MEAS	`0x30210001`	U2	0.001	s	Nominal time between GNSS measurements
E.g. 100 ms results in 10 Hz measurement rate, 1000 ms = 1 Hz measurement rate. The minimum value is 25.					
CFG-RATE-NAV	`0x30210002`	U2	-	-	Ratio of number of measurements to number of navigation solutions
E.g. 5 means five measurements for every navigation solution. The minimum value is 1. The maximum value is 127.					
CFG-RATE-TIMEREF	`0x20210003`	E1	-	-	Time system to which measurements are aligned
See Table 34 below for a list of possible constants for this item.					

Table 33: CFG-RATE configuration items

Constant	Value	Description
UTC	0	Align measurements to UTC time
GPS	1	Align measurements to GPS time
GLO	2	Align measurements to GLONASS time
BDS	3	Align measurements to BeiDou time
GAL	4	Align measurements to Galileo time

Constant	Value	Description
NAVIC	5	Align measurements to NavIC time

Table 34: Constants for CFG-RATE-TIMEREF

6.9.19 CFG-RINV: Remote inventory

The remote inventory enables storing user-defined data in the non-volatile memory of the receiver. The data can be either binary or a string of ASCII characters. In the latter case, it can optionally be output at startup after the boot screen.

Configuration item	Key ID	Type	Scale	Unit	Description
CFG-RINV-DUMP	0x10c70001	L	-	-	Dump data at startup
When true, data will be dumped to the interface on startup, unless CFG-RINV-BINARY is set.					
CFG-RINV-BINARY	0x10c70002	L	-	-	Data is binary
When true, the data is treated as binary data.					
CFG-RINV-DATA_SIZE	0x20c70003	U1	-	-	Size of data
Size of data to store/be stored in the remote inventory (maximum 30 bytes).					
CFG-RINV-CHUNK0	0x50c70004	X8	-	-	Data bytes 1-8 (LSB)
Data to store/be stored in remote inventory - max 8 bytes, left-most in LSB, e.g. string ABCD will appear as 0x44434241.					
CFG-RINV-CHUNK1	0x50c70005	X8	-	-	Data bytes 9-16
Data to store/be stored in remote inventory - max 8 bytes, left-most in LSB, e.g. string ABCD will appear as 0x44434241.					
CFG-RINV-CHUNK2	0x50c70006	X8	-	-	Data bytes 17-24
Data to store/be stored in remote inventory - max 8 bytes, left-most in LSB, e.g. string ABCD will appear as 0x44434241.					
CFG-RINV-CHUNK3	0x50c70007	X8	-	-	Data bytes 25-30 (MSB)
Data to store/be stored in remote inventory - max 6 bytes, left-most in LSB, e.g. string ABCD will appear as 0x44434241.					

Table 35: CFG-RINV configuration items

6.9.20 CFG-RTCM: RTCM protocol configuration

Configures the RTCM protocol.

Configuration item	Key ID	Type	Scale	Unit	Description
CFG-RTCM-DF003_OUT	0x30090001	U2	-	-	RTCM DF003 (Reference station ID) output value
Value to set in RTCM data field DF003 (Reference station ID) in RTCM output messages containing DF003. The value can be 0..4095.					
CFG-RTCM-DF003_IN	0x30090008	U2	-	-	RTCM DF003 (Reference station ID) input value
Value to use for filtering out RTCM input messages based on their DF003 data field (Reference station ID) value. To be used in conjunction with CFG-RTCM-DF003_IN_FILTER. The value can be 0..4095.					
CFG-RTCM-DF003_IN_FILTER	0x20090009	E1	-	-	RTCM input filter configuration based on RTCM DF003 (Reference station ID) value
Configures if and how the filtering out of RTCM input messages based on their DF003 data field (Reference station ID) operates. See Table 37 below for a list of possible constants for this item.					

Table 36: CFG-RTCM configuration items

Constant	Value	Description
DISABLED	0	Disabled RTCM input filter; all input messages allowed
RELAXED	1	Relaxed RTCM input filter; input messages allowed must contain a DF003 data field matching the CFG-RTCM-DF003_IN value or not contain by specification the DF003 data field

Constant	Value	Description
STRICT	2	Strict RTCM input filter; input messages allowed must contain a DF003 data field matching the CFG-RTCM-DF003 value

Table 37: Constants for CFG-RTCM-DF003_IN_FILTER

6.9.21 CFG-SBAS: SBAS configuration

This group configures the SBAS receiver subsystem (i.e. WAAS, EGNOS, MSAS). See SBAS configuration settings description in the integration manual for a detailed description of how these settings affect receiver operation.

Configuration item	Key ID	Type	Scale	Unit	Description
CFG-SBAS-USE_TESTMODE	0x10360002	L	-	-	Use SBAS data when it is in test mode (SBAS msg 0)
CFG-SBAS-USE_RANGING	0x10360003	L	-	-	Use SBAS GEOs as a ranging source (for navigation)
CFG-SBAS-USE_DIFFCORR	0x10360004	L	-	-	Use SBAS differential corrections
CFG-SBAS-USE_INTEGRITY	0x10360005	L	-	-	Use SBAS integrity information
If enabled, the receiver will only use GPS satellites for which integrity information is available					
CFG-SBAS-PRNSCANMASK	0x50360006	X8	-	-	SBAS PRN search configuration
This configuration item determines which SBAS PRNs should be searched. Setting it to 0 indicates auto-scanning all SBAS PRNs. For non-zero values the bits correspond to the allocated SBAS PRNs ranging from PRN120 (bit 0) to PRN158 (bit 38), where a bit set enables searching for the corresponding PRN. See Table 39 below for a list of possible constants for this item.					

Table 38: CFG-SBAS configuration items

Constant	Value	Description
ALL	0x0000000000000000	Enable search for all SBAS PRNs
PRN120	0x0000000000000001	Enable search for SBAS PRN120
PRN121	0x0000000000000002	Enable search for SBAS PRN121
PRN122	0x0000000000000004	Enable search for SBAS PRN122
PRN123	0x0000000000000008	Enable search for SBAS PRN123
PRN124	0x0000000000000010	Enable search for SBAS PRN124
PRN125	0x0000000000000020	Enable search for SBAS PRN125
PRN126	0x0000000000000040	Enable search for SBAS PRN126
PRN127	0x0000000000000080	Enable search for SBAS PRN127
PRN128	0x0000000000000100	Enable search for SBAS PRN128
PRN129	0x0000000000000200	Enable search for SBAS PRN129
PRN130	0x0000000000000400	Enable search for SBAS PRN130
PRN131	0x0000000000000800	Enable search for SBAS PRN131
PRN132	0x0000000000001000	Enable search for SBAS PRN132
PRN133	0x0000000000002000	Enable search for SBAS PRN133
PRN134	0x0000000000004000	Enable search for SBAS PRN134
PRN135	0x0000000000008000	Enable search for SBAS PRN135
PRN136	0x0000000000010000	Enable search for SBAS PRN136
PRN137	0x0000000000020000	Enable search for SBAS PRN137
PRN138	0x0000000000040000	Enable search for SBAS PRN138

Constant	Value	Description
PRN139	0x0000000000080000	Enable search for SBAS PRN139
PRN140	0x0000000000100000	Enable search for SBAS PRN140
PRN141	0x0000000000200000	Enable search for SBAS PRN141
PRN142	0x0000000000400000	Enable search for SBAS PRN142
PRN143	0x0000000000800000	Enable search for SBAS PRN143
PRN144	0x0000000001000000	Enable search for SBAS PRN144
PRN145	0x0000000002000000	Enable search for SBAS PRN145
PRN146	0x0000000004000000	Enable search for SBAS PRN146
PRN147	0x0000000008000000	Enable search for SBAS PRN147
PRN148	0x0000000010000000	Enable search for SBAS PRN148
PRN149	0x0000000020000000	Enable search for SBAS PRN149
PRN150	0x0000000040000000	Enable search for SBAS PRN150
PRN151	0x0000000080000000	Enable search for SBAS PRN151
PRN152	0x0000000100000000	Enable search for SBAS PRN152
PRN153	0x0000000200000000	Enable search for SBAS PRN153
PRN154	0x0000000400000000	Enable search for SBAS PRN154
PRN155	0x0000000800000000	Enable search for SBAS PRN155
PRN156	0x0000001000000000	Enable search for SBAS PRN156
PRN157	0x0000002000000000	Enable search for SBAS PRN157
PRN158	0x0000004000000000	Enable search for SBAS PRN158

Table 39: Constants for CFG-SBAS-PRNSCANMASK

6.9.22 CFG-SEC: Security configuration

Security configuration.

Configuration item	Key ID	Type	Scale	Unit	Description
CFG-SEC-CFG_LOCK	0x10f60009	L	-	-	Configuration lockdown
When set, receiver configuration is locked and cannot be changed any more.					
CFG-SEC-CFG_LOCK_UNLOCKGRP1	0x30f6000a	U2	-	-	Configuration lockdown exempted group 1
This item can be set before enabling the configuration lockdown. It will make writes to the specified group possible after the configuration lockdown has been enabled.					
CFG-SEC-CFG_LOCK_UNLOCKGRP2	0x30f6000b	U2	-	-	Configuration lockdown exempted group 2
This item can be set before enabling the configuration lockdown. It will make writes to the specified group possible after the configuration lockdown has been enabled.					

Table 40: CFG-SEC configuration items

6.9.23 CFG-SIGNAL: Satellite systems (GNSS) signal configuration

The enable items for individual signals are governed by their corresponding constellation enable item. It is necessary that at least one signal from a major GNSS constellation is enabled. See GNSS signal configuration in the integration manual for more details.

Configuration specific to a GNSS system is available in other groups (e.g. CFG-SBAS).

Note that changes to any items within this group will trigger a reset to the GNSS subsystem.

Configuration item	Key ID	Type	Scale	Unit	Description
CFG-SIGNAL-GPS_ENA	0x1031001f	L	-	-	GPS enable
CFG-SIGNAL-GPS_L1CA_ENA	0x10310001	L	-	-	GPS L1C/A
CFG-SIGNAL-GPS_L2C_ENA	0x10310003	L	-	-	GPS L2C
CFG-SIGNAL-SBAS_ENA	0x10310020	L	-	-	SBAS enable
CFG-SIGNAL-SBAS_L1CA_ENA	0x10310005	L	-	-	SBAS L1C/A
CFG-SIGNAL-GAL_ENA	0x10310021	L	-	-	Galileo enable
CFG-SIGNAL-GAL_E1_ENA	0x10310007	L	-	-	Galileo E1
CFG-SIGNAL-GAL_E5B_ENA	0x1031000a	L	-	-	Galileo E5b
CFG-SIGNAL-BDS_ENA	0x10310022	L	-	-	BeiDou Enable
CFG-SIGNAL-BDS_B1_ENA	0x1031000d	L	-	-	BeiDou B1I
CFG-SIGNAL-BDS_B2_ENA	0x1031000e	L	-	-	BeiDou B2I
CFG-SIGNAL-QZSS_ENA	0x10310024	L	-	-	QZSS enable
CFG-SIGNAL-QZSS_L1CA_ENA	0x10310012	L	-	-	QZSS L1C/A
CFG-SIGNAL-QZSS_L1S_ENA	0x10310014	L	-	-	QZSS L1S
CFG-SIGNAL-QZSS_L2C_ENA	0x10310015	L	-	-	QZSS L2C
CFG-SIGNAL-GLO_ENA	0x10310025	L	-	-	GLONASS enable
CFG-SIGNAL-GLO_L1_ENA	0x10310018	L	-	-	GLONASS L1
CFG-SIGNAL-GLO_L2_ENA	0x1031001a	L	-	-	GLONASS L2

Table 41: CFG-SIGNAL configuration items

6.9.24 CFG-SPARTN: SPARTN configuration

Configuration for the SPARTN input stream.

Configuration item	Key ID	Type	Scale	Unit	Description
CFG-SPARTN-USE_SOURCE	0x20a70001	E1	-	-	Selector for source SPARTN stream
See Table 43 below for a list of possible constants for this item.					

Table 42: CFG-SPARTN configuration items

Constant	Value	Description
IP	0x00	IP source (default)
Selects IP (Raw) source		
LBAND	0x01	L-Band source
Selects L-Band (UBX-RXM-PMP) source		

Table 43: Constants for CFG-SPARTN-USE_SOURCE

6.9.25 CFG-SPI: Configuration of the SPI interface

Settings needed to configure the SPI communication interface.

Configuration item	Key ID	Type	Scale	Unit	Description
CFG-SPI-MAXFF	0x20640001	U1	-	-	Number of bytes containing 0xFF to receive before switching off reception. Range: 0 (mechanism off) - 63
CFG-SPI-CPOLARITY	0x10640002	L	-	-	Clock polarity select: 0: Active Hight Clock, SCLK idles low, 1: Active Low Clock, SCLK idles high

Configuration item	Key ID	Type	Scale	Unit	Description
CFG-SPI-CPHASE	`0x10640003`	L	-	-	Clock phase select: 0: Data captured on first edge of SCLK, 1: Data captured on second edge of SCLK
CFG-SPI-EXTENDEDTIMEOUT	`0x10640005`	L	-	-	Flag to disable timeouting the interface after 1.5s
CFG-SPI-ENABLED	`0x10640006`	L	-	-	Flag to indicate if the SPI interface should be enabled

Table 44: CFG-SPI configuration items

6.9.26 CFG-SPIINPROT: Input protocol configuration of the SPI interface

Input protocol enable flags of the SPI interface.

Configuration item	Key ID	Type	Scale	Unit	Description
CFG-SPIINPROT-UBX	`0x10790001`	L	-	-	Flag to indicate if UBX should be an input protocol on SPI
CFG-SPIINPROT-NMEA	`0x10790002`	L	-	-	Flag to indicate if NMEA should be an input protocol on SPI
CFG-SPIINPROT-RTCM3X	`0x10790004`	L	-	-	Flag to indicate if RTCM3X should be an input protocol on SPI
CFG-SPIINPROT-SPARTN	`0x10790005`	L	-	-	Flag to indicate if SPARTN should be an input protocol on SPI

Table 45: CFG-SPIINPROT configuration items

6.9.27 CFG-SPIOUTPROT: Output protocol configuration of the SPI interface

Output protocol enable flags of the SPI interface.

Configuration item	Key ID	Type	Scale	Unit	Description
CFG-SPIOUTPROT-UBX	`0x107a0001`	L	-	-	Flag to indicate if UBX should be an output protocol on SPI
CFG-SPIOUTPROT-NMEA	`0x107a0002`	L	-	-	Flag to indicate if NMEA should be an output protocol on SPI
CFG-SPIOUTPROT-RTCM3X	`0x107a0004`	L	-	-	Flag to indicate if RTCM3X should be an output protocol on SPI

Table 46: CFG-SPIOUTPROT configuration items

6.9.28 CFG-TMODE: Time mode configuration

Configuration for operation of the receiver in Time mode. The position referred to in the configuration items is that of the Antenna Reference Point (ARP).

Configuration item	Key ID	Type	Scale	Unit	Description
CFG-TMODE-MODE	`0x20030001`	E1	-	-	Receiver mode
See Table 48 below for a list of possible constants for this item.					
CFG-TMODE-POS_TYPE	`0x20030002`	E1	-	-	Determines whether the ARP position is given in ECEF or LAT/LON/HEIGHT?
See Table 49 below for a list of possible constants for this item.					
CFG-TMODE-ECEF_X	`0x40030003`	I4	-	cm	ECEF X coordinate of the ARP position.
This will only be used if CFG-TMODE-MODE=FIXED and CFG-TMODE-POS_TYPE=ECEF.					
CFG-TMODE-ECEF_Y	`0x40030004`	I4	-	cm	ECEF Y coordinate of the ARP position.
This will only be used if CFG-TMODE-MODE=FIXED and CFG-TMODE-POS_TYPE=ECEF.					
CFG-TMODE-ECEF_Z	`0x40030005`	I4	-	cm	ECEF Z coordinate of the ARP position.

Configuration item	Key ID	Type	Scale	Unit	Description
					This will only be used if CFG-TMODE-MODE=FIXED and CFG-TMODE-POS_TYPE=ECEF.
CFG-TMODE-ECEF_X_HP	`0x20030006`	I1	0.1	mm	High-precision ECEF X coordinate of the ARP position.
					Accepted range is -99 to +99. This will only be used if CFG-TMODE-MODE=FIXED and CFG-TMODE-POS_TYPE=ECEF.
CFG-TMODE-ECEF_Y_HP	`0x20030007`	I1	0.1	mm	High-precision ECEF Y coordinate of the ARP position.
					Accepted range is -99 to +99. This will only be used if CFG-TMODE-MODE=FIXED and CFG-TMODE-POS_TYPE=ECEF.
CFG-TMODE-ECEF_Z_HP	`0x20030008`	I1	0.1	mm	High-precision ECEF Z coordinate of the ARP position.
					Accepted range is -99 to +99. This will only be used if CFG-TMODE-MODE=FIXED and CFG-TMODE-POS_TYPE=ECEF.
CFG-TMODE-LAT	`0x40030009`	I4	1e-7	deg	Latitude of the ARP position.
					This will only be used if CFG-TMODE-MODE=FIXED and CFG-TMODE-POS_TYPE=LLH.
CFG-TMODE-LON	`0x4003000a`	I4	1e-7	deg	Longitude of the ARP position.
					This will only be used if CFG-TMODE-MODE=FIXED and CFG-TMODE-POS_TYPE=LLH.
CFG-TMODE-HEIGHT	`0x4003000b`	I4	-	cm	Height of the ARP position.
					This will only be used if CFG-TMODE-MODE=FIXED and CFG-TMODE-POS_TYPE=LLH.
CFG-TMODE-LAT_HP	`0x2003000c`	I1	1e-9	deg	High-precision latitude of the ARP position
					Accepted range is -99 to +99. This will only be used if CFG-TMODE-MODE=FIXED and CFG-TMODE-POS_TYPE=LLH.
CFG-TMODE-LON_HP	`0x2003000d`	I1	1e-9	deg	High-precision longitude of the ARP position.
					Accepted range is -99 to +99. This will only be used if CFG-TMODE-MODE=FIXED and CFG-TMODE-POS_TYPE=LLH.
CFG-TMODE-HEIGHT_HP	`0x2003000e`	I1	0.1	mm	High-precision height of the ARP position.
					Accepted range is -99 to +99. This will only be used if CFG-TMODE-MODE=FIXED and CFG-TMODE-POS_TYPE=LLH.
CFG-TMODE-FIXED_POS_ACC	`0x4003000f`	U4	0.1	mm	Fixed position 3D accuracy
CFG-TMODE-SVIN_MIN_DUR	`0x40030010`	U4	-	s	Survey-in minimum duration
					This will only be used if CFG-TMODE-MODE=SURVEY_IN.
CFG-TMODE-SVIN_ACC_LIMIT	`0x40030011`	U4	0.1	mm	Survey-in position accuracy limit
					This will only be used if CFG-TMODE-MODE=SURVEY_IN.

Table 47: CFG-TMODE configuration items

Constant	Value	Description
DISABLED	`0`	Disabled
SURVEY_IN	`1`	Survey in
FIXED	`2`	Fixed mode (true ARP position information required)

Table 48: Constants for CFG-TMODE-MODE

Constant	Value	Description
ECEF	`0`	Position is ECEF

Constant	Value	Description
LLH	1	Position is Lat/Lon/Height

Table 49: Constants for CFG-TMODE-POS_TYPE

6.9.29 CFG-TP: Timepulse configuration

Use this group to configure the generation of timepulses.

Configuration item	Key ID	Type	Scale	Unit	Description
CFG-TP-PULSE_DEF	0x20050023	E1	-	-	Determines whether the time pulse is interpreted as frequency or period
See Table 51 below for a list of possible constants for this item.					
CFG-TP-PULSE_LENGTH_DEF	0x20050030	E1	-	-	Determines whether the time pulse length is interpreted as length[us] or pulse ratio[%]
See Table 52 below for a list of possible constants for this item.					
CFG-TP-ANT_CABLEDELAY	0x30050001	I2	1e-9	s	Antenna cable delay
CFG-TP-PERIOD_TP1	0x40050002	U4	1e-6	s	Time pulse period (TP1)
CFG-TP-PERIOD_LOCK_TP1	0x40050003	U4	1e-6	s	Time pulse period when locked to GNSS time (TP1)
Only used if CFG-TP-USE_LOCKED_TP1 is set.					
CFG-TP-FREQ_TP1	0x40050024	U4	-	Hz	Time pulse frequency (TP1)
This will only be used if CFG-TP-PULSE_DEF=FREQ.					
CFG-TP-FREQ_LOCK_TP1	0x40050025	U4	-	Hz	Time pulse frequency when locked to GNSS time (TP1)
Only used if CFG-TP-USE_LOCKED_TP1 is set.					
CFG-TP-LEN_TP1	0x40050004	U4	1e-6	s	Time pulse length (TP1)
CFG-TP-LEN_LOCK_TP1	0x40050005	U4	1e-6	s	Time pulse length when locked to GNSS time (TP1)
Only used if CFG-TP-USE_LOCKED_TP1 is set.					
CFG-TP-DUTY_TP1	0x5005002a	R8	-	%	Time pulse duty cycle (TP1)
Only used if CFG-TP-PULSE_LENGTH_DEF=RATIO is set.					
CFG-TP-DUTY_LOCK_TP1	0x5005002b	R8	-	%	Time pulse duty cycle when locked to GNSS time (TP1)
Only used if CFG-TP-PULSE_LENGTH_DEF=RATIO and CFG-TP-USE_LOCKED_TP1 are set.					
CFG-TP-USER_DELAY_TP1	0x40050006	I4	1e-9	s	User-configurable time pulse delay (TP1)
CFG-TP-TP1_ENA	0x10050007	L	-	-	Enable the first timepulse
if pin associated with time pulse is assigned for another function, the other function takes precedence. Must be set for frequency-time products.					
CFG-TP-SYNC_GNSS_TP1	0x10050008	L	-	-	Sync time pulse to GNSS time or local clock (TP1)
If set, sync to GNSS if GNSS time is valid otherwise, if not set or not available, use local clock. Ignored by time-frequency product variants, which will attempt to use the best available time/frequency reference (not necessarily GNSS). This flag can be unset only in Timing product variants.					
CFG-TP-USE_LOCKED_TP1	0x10050009	L	-	-	Use locked parameters when possible (TP1)
If set, use CFG-TP-PERIOD_LOCK_TP1 and CFG-TP-LEN_LOCK_TP1 as soon as GNSS time is valid. Otherwise if not valid or not set, use CFG-TP-PERIOD_TP1 and CFG-TP-LEN_TP1.					
CFG-TP-ALIGN_TO_TOW_TP1	0x1005000a	L	-	-	Align time pulse to top of second (TP1)

Configuration item	Key ID	Type	Scale	Unit	Description
To use this feature, CFG-TP-USE_LOCKED_TP1 must be set. Time pulse period must be an integer fraction of 1 second. Ignored in time-frequency product variants, where it is assumed always enabled.					
CFG-TP-POL_TP1	`0x1005000b`	L	-	-	Set time pulse polarity (TP1)
false (0) : falling edge at top of second. true (1) : rising edge at top of second.					
CFG-TP-TIMEGRID_TP1	`0x2005000c`	E1	-	-	Time grid to use (TP1)
Only relevant if CFG-TP-USE_LOCKED_TP1 and ALIGN_TO_TOW_TP1 are set. Note that configured GNSS time is estimated by the receiver if locked to any GNSS system. If the receiver has a valid GNSS fix it will attempt to steer the TP to the specified time grid even if the specified time is not based on information from the constellation's satellites. To ensure timing based purely on a given GNSS, restrict the supported constellations in CFG-SIGNAL-*. See Table 53 below for a list of possible constants for this item.					
CFG-TP-DRSTR_TP1	`0x20050035`	E1	-	-	Set drive strength of TP1
Time Pulse pin 1 (TP1) can support 4 possible drive strength cases: 2, 4, 8 and 12 mA See Table 54 below for a list of possible constants for this item.					

Table 50: CFG-TP configuration items

Constant	Value	Description
PERIOD	0	Time pulse period [us]
FREQ	1	Time pulse frequency [Hz]

Table 51: Constants for CFG-TP-PULSE_DEF

Constant	Value	Description
RATIO	0	Time pulse ratio
LENGTH	1	Time pulse length

Table 52: Constants for CFG-TP-PULSE_LENGTH_DEF

Constant	Value	Description
UTC	0	UTC time reference
GPS	1	GPS time reference
GLO	2	GLONASS time reference
BDS	3	BeiDou time reference
GAL	4	Galileo time reference
NAVIC	5	NavIC time reference

Table 53: Constants for CFG-TP-TIMEGRID_TP1

Constant	Value	Description
DRIVE_STRENGTH_2MA	0	2 mA drive strength
DRIVE_STRENGTH_4MA	1	4 mA drive strength
DRIVE_STRENGTH_8MA	2	8 mA drive strength
DRIVE_STRENGTH_12MA	3	12 mA drive strength

Table 54: Constants for CFG-TP-DRSTR_TP1

6.9.30 CFG-TXREADY: TX ready configuration

Configuration of the TX ready pin.

Configuration item	Key ID	Type	Scale	Unit	Description
CFG-TXREADY-ENABLED	0x10a20001	L	-	-	Flag to indicate if TX ready pin mechanism should be enabled
CFG-TXREADY-POLARITY	0x10a20002	L	-	-	The polarity of the TX ready pin: false:high-active, true:low-active
CFG-TXREADY-PIN	0x20a20003	U1	-	-	Pin number to use for the TX ready functionality
CFG-TXREADY-THRESHOLD	0x30a20004	U2	-	-	Amount of data that should be ready on the interface before triggering the TX ready pin
CFG-TXREADY-INTERFACE	0x20a20005	E1	-	-	Interface where the TX ready feature should be linked to
See Table 56 below for a list of possible constants for this item.					

Table 55: CFG-TXREADY configuration items

Constant	Value	Description
I2C	0	I2C interface
SPI	1	SPI interface

Table 56: Constants for CFG-TXREADY-INTERFACE

6.9.31 CFG-UART1: Configuration of the UART1 interface

Settings needed to configure the UART1 communication interface.

Configuration item	Key ID	Type	Scale	Unit	Description
CFG-UART1-BAUDRATE	0x40520001	U4	-	-	The baud rate that should be configured on the UART1
CFG-UART1-STOPBITS	0x20520002	E1	-	-	Number of stopbits that should be used on UART1
See Table 58 below for a list of possible constants for this item.					
CFG-UART1-DATABITS	0x20520003	E1	-	-	Number of databits that should be used on UART1
See Table 59 below for a list of possible constants for this item.					
CFG-UART1-PARITY	0x20520004	E1	-	-	Parity mode that should be used on UART1
See Table 60 below for a list of possible constants for this item.					
CFG-UART1-ENABLED	0x10520005	L	-	-	Flag to indicate if the UART1 should be enabled

Table 57: CFG-UART1 configuration items

Constant	Value	Description
HALF	0	0.5 stopbits
ONE	1	1.0 stopbits
ONEHALF	2	1.5 stopbits
TWO	3	2.0 stopbits

Table 58: Constants for CFG-UART1-STOPBITS

Constant	Value	Description
EIGHT	0	8 databits
SEVEN	1	7 databits

Table 59: Constants for CFG-UART1-DATABITS

Constant	Value	Description
NONE	0	No parity bit

Constant	Value	Description
ODD	1	Add an odd parity bit
EVEN	2	Add an even parity bit

Table 60: Constants for CFG-UART1-PARITY

6.9.32 CFG-UART1INPROT: Input protocol configuration of the UART1 interface

Input protocol enable flags of the UART1 interface.

Configuration item	Key ID	Type	Scale	Unit	Description
CFG-UART1INPROT-UBX	0x10730001	L	-	-	Flag to indicate if UBX should be an input protocol on UART1
CFG-UART1INPROT-NMEA	0x10730002	L	-	-	Flag to indicate if NMEA should be an input protocol on UART1
CFG-UART1INPROT-RTCM3X	0x10730004	L	-	-	Flag to indicate if RTCM3X should be an input protocol on UART1
CFG-UART1INPROT-SPARTN	0x10730005	L	-	-	Flag to indicate if SPARTN should be an input protocol on UART1

Table 61: CFG-UART1INPROT configuration items

6.9.33 CFG-UART1OUTPROT: Output protocol configuration of the UART1 interface

Output protocol enable flags of the UART1 interface.

Configuration item	Key ID	Type	Scale	Unit	Description
CFG-UART1OUTPROT-UBX	0x10740001	L	-	-	Flag to indicate if UBX should be an output protocol on UART1
CFG-UART1OUTPROT-NMEA	0x10740002	L	-	-	Flag to indicate if NMEA should be an output protocol on UART1
CFG-UART1OUTPROT-RTCM3X	0x10740004	L	-	-	Flag to indicate if RTCM3X should be an output protocol on UART1

Table 62: CFG-UART1OUTPROT configuration items

6.9.34 CFG-UART2: Configuration of the UART2 interface

Settings needed to configure the UART2 communication interface.

Configuration item	Key ID	Type	Scale	Unit	Description
CFG-UART2-BAUDRATE	0x40530001	U4	-	-	The baud rate that should be configured on the UART2
CFG-UART2-STOPBITS	0x20530002	E1	-	-	Number of stopbits that should be used on UART2
See Table 64 below for a list of possible constants for this item.					
CFG-UART2-DATABITS	0x20530003	E1	-	-	Number of databits that should be used on UART2
See Table 65 below for a list of possible constants for this item.					
CFG-UART2-PARITY	0x20530004	E1	-	-	Parity mode that should be used on UART2
See Table 66 below for a list of possible constants for this item.					
CFG-UART2-ENABLED	0x10530005	L	-	-	Flag to indicate if the UART2 should be enabled

Table 63: CFG-UART2 configuration items

Constant	Value	Description
HALF	0	0.5 stopbits

Constant	Value	Description
ONE	1	1.0 stopbits
ONEHALF	2	1.5 stopbits
TWO	3	2.0 stopbits

Table 64: Constants for CFG-UART2-STOPBITS

Constant	Value	Description
EIGHT	0	8 databits
SEVEN	1	7 databits

Table 65: Constants for CFG-UART2-DATABITS

Constant	Value	Description
NONE	0	No parity bit
ODD	1	Add an odd parity bit
EVEN	2	Add an even parity bit

Table 66: Constants for CFG-UART2-PARITY

6.9.35 CFG-UART2INPROT: Input protocol configuration of the UART2 interface

Input protocol enable flags of the UART2 interface.

Configuration item	Key ID	Type	Scale	Unit	Description
CFG-UART2INPROT-UBX	`0x10750001`	L	-	-	Flag to indicate if UBX should be an input protocol on UART2
CFG-UART2INPROT-NMEA	`0x10750002`	L	-	-	Flag to indicate if NMEA should be an input protocol on UART2
CFG-UART2INPROT-RTCM3X	`0x10750004`	L	-	-	Flag to indicate if RTCM3X should be an input protocol on UART2
CFG-UART2INPROT-SPARTN	`0x10750005`	L	-	-	Flag to indicate if SPARTN should be an input protocol on UART2

Table 67: CFG-UART2INPROT configuration items

6.9.36 CFG-UART2OUTPROT: Output protocol configuration of the UART2 interface

Output protocol enable flags of the UART2 interface.

Configuration item	Key ID	Type	Scale	Unit	Description
CFG-UART2OUTPROT-UBX	`0x10760001`	L	-	-	Flag to indicate if UBX should be an output protocol on UART2
CFG-UART2OUTPROT-NMEA	`0x10760002`	L	-	-	Flag to indicate if NMEA should be an output protocol on UART2
CFG-UART2OUTPROT-RTCM3X	`0x10760004`	L	-	-	Flag to indicate if RTCM3X should be an output protocol on UART2

Table 68: CFG-UART2OUTPROT configuration items

6.9.37 CFG-USB: Configuration of the USB interface

Settings needed to configure the USB communication interface.

Configuration item	Key ID	Type	Scale	Unit	Description
CFG-USB-ENABLED	`0x10650001`	L	-	-	Flag to indicate if the USB interface should be enabled
CFG-USB-SELFPOW	`0x10650002`	L	-	-	Self-powered device

Configuration item	Key ID	Type	Scale	Unit	Description
CFG-USB-VENDOR_ID	0x3065000a	U2	-	-	Vendor ID
CFG-USB-PRODUCT_ID	0x3065000b	U2	-	-	Vendor ID
CFG-USB-POWER	0x3065000c	U2	-	mA	Power consumption
CFG-USB-VENDOR_STR0	0x5065000d	X8	-	-	Vendor string characters 0-7
CFG-USB-VENDOR_STR1	0x5065000e	X8	-	-	Vendor string characters 8-15
CFG-USB-VENDOR_STR2	0x5065000f	X8	-	-	Vendor string characters 16-23
CFG-USB-VENDOR_STR3	0x50650010	X8	-	-	Vendor string characters 24-31
CFG-USB-PRODUCT_STR0	0x50650011	X8	-	-	Product string characters 0-7
CFG-USB-PRODUCT_STR1	0x50650012	X8	-	-	Product string characters 8-15
CFG-USB-PRODUCT_STR2	0x50650013	X8	-	-	Product string characters 16-23
CFG-USB-PRODUCT_STR3	0x50650014	X8	-	-	Product string characters 24-31
CFG-USB-SERIAL_NO_STR0	0x50650015	X8	-	-	Serial number string characters 0-7
CFG-USB-SERIAL_NO_STR1	0x50650016	X8	-	-	Serial number string characters 8-15
CFG-USB-SERIAL_NO_STR2	0x50650017	X8	-	-	Serial number string characters 16-23
CFG-USB-SERIAL_NO_STR3	0x50650018	X8	-	-	Serial number string characters 24-31

Table 69: CFG-USB configuration items

6.9.38 CFG-USBINPROT: Input protocol configuration of the USB interface

Input protocol enable flags of the USB interface.

Configuration item	Key ID	Type	Scale	Unit	Description
CFG-USBINPROT-UBX	0x10770001	L	-	-	Flag to indicate if UBX should be an input protocol on USB
CFG-USBINPROT-NMEA	0x10770002	L	-	-	Flag to indicate if NMEA should be an input protocol on USB
CFG-USBINPROT-RTCM3X	0x10770004	L	-	-	Flag to indicate if RTCM3X should be an input protocol on USB
CFG-USBINPROT-SPARTN	0x10770005	L	-	-	Flag to indicate if SPARTN should be an input protocol on USB

Table 70: CFG-USBINPROT configuration items

6.9.39 CFG-USBOUTPROT: Output protocol configuration of the USB interface

Output protocol enable flags of the USB interface.

Configuration item	Key ID	Type	Scale	Unit	Description
CFG-USBOUTPROT-UBX	0x10780001	L	-	-	Flag to indicate if UBX should be an output protocol on USB
CFG-USBOUTPROT-NMEA	0x10780002	L	-	-	Flag to indicate if NMEA should be an output protocol on USB
CFG-USBOUTPROT-RTCM3X	0x10780004	L	-	-	Flag to indicate if RTCM3X should be an output protocol on USB

Table 71: CFG-USBOUTPROT configuration items

6.10 Legacy UBX message fields reference

The following table lists the legacy UBX message fields and the corresponding configuration item. Note that the mapping from UBX-CFG message fields to configuration items is not necessarily 1:1 and that that some legacy UBX-CFG messages may not be available for certain products.

UBX message and field	Configuration item(s)
UBX-CFG-ANT	
UBX-CFG-ANT.ocd	CFG-HW-ANT_CFG_OPENDET
UBX-CFG-ANT.pdwnOnSCD	CFG-HW-ANT_CFG_PWRDOWN
UBX-CFG-ANT.pinOCD	CFG-HW-ANT_SUP_OPEN_PIN
UBX-CFG-ANT.pinSCD	CFG-HW-ANT_SUP_SHORT_PIN
UBX-CFG-ANT.pinSwitch	CFG-HW-ANT_SUP_SWITCH_PIN
UBX-CFG-ANT.recovery	CFG-HW-ANT_CFG_RECOVER
UBX-CFG-ANT.scd	CFG-HW-ANT_CFG_SHORTDET
UBX-CFG-ANT.svcs	CFG-HW-ANT_CFG_VOLTCTRL
UBX-CFG-DAT	
UBX-CFG-DAT.dX	CFG-NAVSPG-USRDAT_DX
UBX-CFG-DAT.dY	CFG-NAVSPG-USRDAT_DY
UBX-CFG-DAT.dZ	CFG-NAVSPG-USRDAT_DZ
UBX-CFG-DAT.flat	CFG-NAVSPG-USRDAT_FLAT
UBX-CFG-DAT.majA	CFG-NAVSPG-USE_USRDAT, CFG-NAVSPG-USRDAT_MAJA
UBX-CFG-DAT.rotX	CFG-NAVSPG-USRDAT_ROTX
UBX-CFG-DAT.rotY	CFG-NAVSPG-USRDAT_ROTY
UBX-CFG-DAT.rotZ	CFG-NAVSPG-USRDAT_ROTZ
UBX-CFG-DAT.scale	CFG-NAVSPG-USRDAT_SCALE
UBX-CFG-DGNSS	
UBX-CFG-DGNSS.dgnssMode	CFG-NAVHPG-DGNSSMODE
UBX-CFG-GEOFENCE	
UBX-CFG-GEOFENCE.confLvl	CFG-GEOFENCE-CONFLVL
UBX-CFG-GEOFENCE.lat	CFG-GEOFENCE-FENCE1_LAT, CFG-GEOFENCE-FENCE2_LAT, CFG-GEOFENCE-FENCE3_LAT, CFG-GEOFENCE-FENCE4_LAT
UBX-CFG-GEOFENCE.lon	CFG-GEOFENCE-FENCE1_LON, CFG-GEOFENCE-FENCE2_LON, CFG-GEOFENCE-FENCE3_LON, CFG-GEOFENCE-FENCE4_LON
UBX-CFG-GEOFENCE.numFences	CFG-GEOFENCE-USE_FENCE1, CFG-GEOFENCE-USE_FENCE2, CFG-GEOFENCE-USE_FENCE3, CFG-GEOFENCE-USE_FENCE4
UBX-CFG-GEOFENCE.pin	CFG-GEOFENCE-PIN
UBX-CFG-GEOFENCE.pinPolarity	CFG-GEOFENCE-PINPOL
UBX-CFG-GEOFENCE.pioEnabled	CFG-GEOFENCE-USE_PIO
UBX-CFG-GEOFENCE.radius	CFG-GEOFENCE-FENCE1_RAD, CFG-GEOFENCE-FENCE2_RAD, CFG-GEOFENCE-FENCE3_RAD, CFG-GEOFENCE-FENCE4_RAD
UBX-CFG-GNSS	
UBX-CFG-GNSS.gnssId	CFG-SIGNAL-GPS_ENA, CFG-SIGNAL-SBAS_ENA, CFG-SIGNAL-BDS_ENA, CFG-SIGNAL-QZSS_ENA, CFG-SIGNAL-GLO_ENA
UBX-CFG-INF	
UBX-CFG-INF.infMsgMask	CFG-INFMSG-UBX_I2C, CFG-INFMSG-UBX_UART1, CFG-INFMSG-UBX_UART2, CFG-INFMSG-UBX_USB, CFG-INFMSG-UBX_SPI, CFG-INFMSG-NMEA_I2C, CFG-INFMSG-NMEA_UART1, CFG-INFMSG-NMEA_UART2, CFG-INFMSG-NMEA_USB, CFG-INFMSG-NMEA_SPI

UBX message and field	Configuration item(s)
UBX-CFG-INF.protocolID	CFG-INFMSG-UBX_UART1, CFG-INFMSG-UBX_UART2, CFG-INFMSG-UBX_USB, CFG-INFMSG-UBX_SPI, CFG-INFMSG-NMEA_I2C, CFG-INFMSG-NMEA_UART1, CFG-INFMSG-NMEA_UART2, CFG-INFMSG-NMEA_USB, CFG-INFMSG-NMEA_SPI
UBX-CFG-ITFM	
UBX-CFG-ITFM.antSetting	CFG-ITFM-ANTSETTING
UBX-CFG-ITFM.bbThreshold	CFG-ITFM-BBTHRESHOLD
UBX-CFG-ITFM.cwThreshold	CFG-ITFM-CWTHRESHOLD
UBX-CFG-ITFM.enable	CFG-ITFM-ENABLE
UBX-CFG-ITFM.enable2	CFG-ITFM-ENABLE_AUX
UBX-CFG-LOGFILTER	
UBX-CFG-LOGFILTER.applyAllFilterSettings	CFG-LOGFILTER-APPLY_ALL_FILTERS
UBX-CFG-LOGFILTER.minInterval	CFG-LOGFILTER-MIN_INTERVAL
UBX-CFG-LOGFILTER.positionThreshold	CFG-LOGFILTER-POSITION_THRS
UBX-CFG-LOGFILTER.psmOncePerWakupEnabled	CFG-LOGFILTER-ONCE_PER_WAKE_UP_ENA
UBX-CFG-LOGFILTER.recordEnabled	CFG-LOGFILTER-RECORD_ENA
UBX-CFG-LOGFILTER.speedThreshold	CFG-LOGFILTER-SPEED_THRS
UBX-CFG-LOGFILTER.timeThreshold	CFG-LOGFILTER-TIME_THRS
UBX-CFG-MOT	
UBX-CFG-MOT.gnssDistThdl	CFG-MOT-GNSSDIST_THRS
UBX-CFG-MOT.gnssSpeedThdl	CFG-MOT-GNSSSPEED_THRS
UBX-CFG-NAV5	
UBX-CFG-NAV5.cnoThresh	CFG-NAVSPG-INFIL_CNOTHRS
UBX-CFG-NAV5.cnoThreshNumSVs	CFG-NAVSPG-INFIL_NCNOTHRS
UBX-CFG-NAV5.dgnssTimeout	CFG-NAVSPG-CONSTR_DGNSSTO
UBX-CFG-NAV5.dynModel	CFG-NAVSPG-DYNMODEL
UBX-CFG-NAV5.fixMode	CFG-NAVSPG-FIXMODE
UBX-CFG-NAV5.fixedAlt	CFG-NAVSPG-CONSTR_ALT
UBX-CFG-NAV5.fixedAltVar	CFG-NAVSPG-CONSTR_ALTVAR
UBX-CFG-NAV5.minElev	CFG-NAVSPG-INFIL_MINELEV
UBX-CFG-NAV5.pAcc	CFG-NAVSPG-OUTFIL_PACC
UBX-CFG-NAV5.pDop	CFG-NAVSPG-OUTFIL_PDOP
UBX-CFG-NAV5.staticHoldMaxDist	CFG-MOT-GNSSDIST_THRS
UBX-CFG-NAV5.staticHoldThresh	CFG-MOT-GNSSSPEED_THRS
UBX-CFG-NAV5.tAcc	CFG-NAVSPG-OUTFIL_TACC, CFG-NAVSPG-OUTFIL_FACC
UBX-CFG-NAV5.tDop	CFG-NAVSPG-OUTFIL_TDOP
UBX-CFG-NAV5.utcStandard	CFG-NAVSPG-UTCSTANDARD
UBX-CFG-NAVX5	
UBX-CFG-NAVX5.ackAiding	CFG-NAVSPG-ACKAIDING
UBX-CFG-NAVX5.iniFix3D	CFG-NAVSPG-INIFIX3D
UBX-CFG-NAVX5.maxSVs	CFG-NAVSPG-INFIL_MAXSVS
UBX-CFG-NAVX5.minCNO	CFG-NAVSPG-INFIL_MINCNO
UBX-CFG-NAVX5.minSVs	CFG-NAVSPG-INFIL_MINSVS
UBX-CFG-NAVX5.wknRollover	CFG-NAVSPG-WKNROLLOVER

UBX message and field	Configuration item(s)
UBX-CFG-NMEA	
UBX-CFG-NMEA.bdsTalkerId	CFG-NMEA-BDSTALKERID
UBX-CFG-NMEA.beidou	CFG-NMEA-FILT_BDS
UBX-CFG-NMEA.compat	CFG-NMEA-COMPAT
UBX-CFG-NMEA.consider	CFG-NMEA-CONSIDER
UBX-CFG-NMEA.dateFilt	CFG-NMEA-OUT_INVDATE
UBX-CFG-NMEA.galileo	CFG-NMEA-FILT_GAL
UBX-CFG-NMEA.glonass	CFG-NMEA-FILT_GLO
UBX-CFG-NMEA.gps	CFG-NMEA-FILT_GPS
UBX-CFG-NMEA.gpsOnlyFilter	CFG-NMEA-OUT_ONLYGPS
UBX-CFG-NMEA.gsvTalkerId	CFG-NMEA-GSVTALKERID
UBX-CFG-NMEA.highPrec	CFG-NMEA-HIGHPREC
UBX-CFG-NMEA.limit82	CFG-NMEA-LIMIT82
UBX-CFG-NMEA.mainTalkerId	CFG-NMEA-MAINTALKERID
UBX-CFG-NMEA.mskPosFilt	CFG-NMEA-OUT_MSKFIX
UBX-CFG-NMEA.nmeaVersion	CFG-NMEA-PROTVER
UBX-CFG-NMEA.numSV	CFG-NMEA-MAXSVS
UBX-CFG-NMEA.posFilt	CFG-NMEA-OUT_INVFIX
UBX-CFG-NMEA.qzss	CFG-NMEA-FILT_QZSS
UBX-CFG-NMEA.sbas	CFG-NMEA-FILT_SBAS
UBX-CFG-NMEA.svNumbering	CFG-NMEA-SVNUMBERING
UBX-CFG-NMEA.timeFilt	CFG-NMEA-OUT_INVTIME
UBX-CFG-NMEA.trackFilt	CFG-NMEA-OUT_FROZENCOG
UBX-CFG-ODO	
UBX-CFG-ODO.cogLpGain	CFG-ODO-COGLPGAIN
UBX-CFG-ODO.cogMaxPosAcc	CFG-ODO-COGMAXPOSACC
UBX-CFG-ODO.cogMaxSpeed	CFG-ODO-COGMAXSPEED
UBX-CFG-ODO.outLPCog	CFG-ODO-OUTLPCOG
UBX-CFG-ODO.outLPVel	CFG-ODO-OUTLPVEL
UBX-CFG-ODO.profile	CFG-ODO-PROFILE
UBX-CFG-ODO.useCOG	CFG-ODO-USE_COG
UBX-CFG-ODO.useODO	CFG-ODO-USE_ODO
UBX-CFG-ODO.velLpGain	CFG-ODO-VELLPGAIN
UBX-CFG-PRT	
UBX-CFG-PRT.en	CFG-TXREADY-ENABLED
UBX-CFG-PRT.extendedTxTimeout	CFG-I2C-EXTENDEDTIMEOUT
UBX-CFG-PRT.inNmea	CFG-I2CINPROT-NMEA
UBX-CFG-PRT.inProtoMask	CFG-I2C-ENABLED
UBX-CFG-PRT.inRtcm3	CFG-I2CINPROT-RTCM3X
UBX-CFG-PRT.inUbx	CFG-I2CINPROT-UBX
UBX-CFG-PRT.outNmea	CFG-I2COUTPROT-NMEA
UBX-CFG-PRT.outProtoMask	CFG-I2C-ENABLED
UBX-CFG-PRT.outRtcm3	CFG-I2COUTPROT-RTCM3X
UBX-CFG-PRT.outUbx	CFG-I2COUTPROT-UBX

UBX message and field	Configuration item(s)
UBX-CFG-PRT.pin	CFG-TXREADY-PIN
UBX-CFG-PRT.pol	CFG-TXREADY-POLARITY
UBX-CFG-PRT.slaveAddr	CFG-I2C-ADDRESS
UBX-CFG-PRT.thres	CFG-TXREADY-THRESHOLD
UBX-CFG-PRT.en	CFG-TXREADY-ENABLED
UBX-CFG-PRT.extendedTxTimeout	CFG-SPI-EXTENDEDTIMEOUT
UBX-CFG-PRT.ffCnt	CFG-SPI-MAXFF
UBX-CFG-PRT.inNmea	CFG-SPIINPROT-NMEA
UBX-CFG-PRT.inProtoMask	CFG-SPI-ENABLED
UBX-CFG-PRT.inRtcm3	CFG-SPIINPROT-RTCM3X
UBX-CFG-PRT.inUbx	CFG-SPIINPROT-UBX
UBX-CFG-PRT.outNmea	CFG-SPIOUTPROT-NMEA
UBX-CFG-PRT.outProtoMask	CFG-SPI-ENABLED
UBX-CFG-PRT.outRtcm3	CFG-SPIOUTPROT-RTCM3X
UBX-CFG-PRT.outUbx	CFG-SPIOUTPROT-UBX
UBX-CFG-PRT.pin	CFG-TXREADY-PIN
UBX-CFG-PRT.pol	CFG-TXREADY-POLARITY
UBX-CFG-PRT.spiMode	CFG-SPI-CPOLARITY, CFG-SPI-CPHASE
UBX-CFG-PRT.thres	CFG-TXREADY-THRESHOLD
UBX-CFG-PRT.baudRate	CFG-UART1-BAUDRATE, CFG-UART2-BAUDRATE
UBX-CFG-PRT.charLen	CFG-UART1-DATABITS, CFG-UART2-DATABITS
UBX-CFG-PRT.inNmea	CFG-UART1INPROT-NMEA, CFG-UART2INPROT-NMEA
UBX-CFG-PRT.inProtoMask	CFG-UART1-ENABLED, CFG-UART2-ENABLED
UBX-CFG-PRT.inRtcm3	CFG-UART1INPROT-RTCM3X, CFG-UART2INPROT-RTCM3X
UBX-CFG-PRT.inUbx	CFG-UART1INPROT-UBX, CFG-UART2INPROT-UBX
UBX-CFG-PRT.nStopBits	CFG-UART1-STOPBITS, CFG-UART2-STOPBITS
UBX-CFG-PRT.outNmea	CFG-UART1OUTPROT-NMEA, CFG-UART2OUTPROT-NMEA
UBX-CFG-PRT.outProtoMask	CFG-UART1-ENABLED, CFG-UART2-ENABLED
UBX-CFG-PRT.outRtcm3	CFG-UART1OUTPROT-RTCM3X, CFG-UART2OUTPROT-RTCM3X
UBX-CFG-PRT.outUbx	CFG-UART1OUTPROT-UBX, CFG-UART2OUTPROT-UBX
UBX-CFG-PRT.parity	CFG-UART1-PARITY, CFG-UART2-PARITY
UBX-CFG-PRT.inNmea	CFG-USBINPROT-NMEA
UBX-CFG-PRT.inProtoMask	CFG-USB-ENABLED
UBX-CFG-PRT.inRtcm3	CFG-USBINPROT-RTCM3X
UBX-CFG-PRT.inUbx	CFG-USBINPROT-UBX
UBX-CFG-PRT.outNmea	CFG-USBOUTPROT-NMEA
UBX-CFG-PRT.outProtoMask	CFG-USB-ENABLED
UBX-CFG-PRT.outRtcm3	CFG-USBOUTPROT-RTCM3X
UBX-CFG-PRT.outUbx	CFG-USBOUTPROT-UBX
UBX-CFG-RATE	
UBX-CFG-RATE.measRate	CFG-RATE-MEAS
UBX-CFG-RATE.navRate	CFG-RATE-NAV
UBX-CFG-RATE.timeRef	CFG-RATE-TIMEREF

UBX message and field	Configuration item(s)
UBX-CFG-RINV	
UBX-CFG-RINV.data	CFG-RINV-DATA_SIZE, CFG-RINV-CHUNK0, CFG-RINV-CHUNK1, CFG-RINV-CHUNK2, CFG-RINV-CHUNK3
UBX-CFG-RINV.flags	CFG-RINV-DUMP, CFG-RINV-BINARY
UBX-CFG-SBAS	
UBX-CFG-SBAS.diffCorr	CFG-SBAS-USE_DIFFCORR
UBX-CFG-SBAS.integrity	CFG-SBAS-USE_INTEGRITY
UBX-CFG-SBAS.range	CFG-SBAS-USE_RANGING
UBX-CFG-SBAS.scanmode1	CFG-SBAS-PRNSCANMASK
UBX-CFG-SBAS.test	CFG-SBAS-USE_TESTMODE
UBX-CFG-SLAS	
UBX-CFG-SLAS.enabled	CFG-QZSS-USE_SLAS_DGNSS
UBX-CFG-SLAS.raim	CFG-QZSS-USE_SLAS_RAIM_UNCORR
UBX-CFG-SLAS.test	CFG-QZSS-USE_SLAS_TESTMODE
UBX-CFG-TMODE3	
UBX-CFG-TMODE3.ecefXOrLat	CFG-TMODE-ECEF_X, CFG-TMODE-LAT
UBX-CFG-TMODE3.ecefXOrLatHP	CFG-TMODE-ECEF_X_HP, CFG-TMODE-LAT_HP
UBX-CFG-TMODE3.ecefYOrLon	CFG-TMODE-ECEF_Y, CFG-TMODE-LON
UBX-CFG-TMODE3.ecefYOrLonHP	CFG-TMODE-ECEF_Y_HP, CFG-TMODE-LON_HP
UBX-CFG-TMODE3.ecefZOrAlt	CFG-TMODE-ECEF_Z, CFG-TMODE-HEIGHT
UBX-CFG-TMODE3.ecefZOrAltHP	CFG-TMODE-ECEF_Z_HP, CFG-TMODE-HEIGHT_HP
UBX-CFG-TMODE3.fixedPosAcc	CFG-TMODE-FIXED_POS_ACC
UBX-CFG-TMODE3.flags	CFG-TMODE-MODE, CFG-TMODE-POS_TYPE
UBX-CFG-TMODE3.svinAccLimit	CFG-TMODE-SVIN_ACC_LIMIT
UBX-CFG-TMODE3.svinMinDur	CFG-TMODE-SVIN_MIN_DUR
UBX-CFG-TP5	
UBX-CFG-TP5.active	CFG-TP-TP1_ENA
UBX-CFG-TP5.alignToTow	CFG-TP-ALIGN_TO_TOW_TP1
UBX-CFG-TP5.antCableDelay	CFG-TP-ANT_CABLEDELAY
UBX-CFG-TP5.freqPeriod	CFG-TP-PERIOD_TP1, CFG-TP-FREQ_TP1
UBX-CFG-TP5.freqPeriodLock	CFG-TP-PERIOD_LOCK_TP1, CFG-TP-FREQ_LOCK_TP1
UBX-CFG-TP5.gridUtcGnss	CFG-TP-TIMEGRID_TP1
UBX-CFG-TP5.isFreq	CFG-TP-PULSE_DEF
UBX-CFG-TP5.isLength	CFG-TP-PULSE_LENGTH_DEF
UBX-CFG-TP5.lockGnssFreq	CFG-TP-SYNC_GNSS_TP1
UBX-CFG-TP5.lockedOtherSet	CFG-TP-USE_LOCKED_TP1
UBX-CFG-TP5.polarity	CFG-TP-POL_TP1
UBX-CFG-TP5.pulseLenRatio	CFG-TP-LEN_TP1, CFG-TP-DUTY_TP1
UBX-CFG-TP5.pulseLenRatioLock	CFG-TP-LEN_LOCK_TP1, CFG-TP-DUTY_LOCK_TP1
UBX-CFG-TP5.userConfigDelay	CFG-TP-USER_DELAY_TP1
UBX-CFG-USB	
UBX-CFG-USB.powerConsumption	CFG-USB-POWER
UBX-CFG-USB.powerMode	CFG-USB-SELFPOW
UBX-CFG-USB.productID	CFG-USB-PRODUCT_ID

UBX message and field	Configuration item(s)
UBX-CFG-USB.productString	CFG-USB-PRODUCT_STR0, CFG-USB-PRODUCT_STR1, CFG-USB-PRODUCT_STR2, CFG-USB-PRODUCT_STR3
UBX-CFG-USB.serialNumber	CFG-USB-SERIAL_NO_STR0, CFG-USB-SERIAL_NO_STR1, CFG-USB-SERIAL_NO_STR2, CFG-USB-SERIAL_NO_STR3
UBX-CFG-USB.vendorID	CFG-USB-VENDOR_ID
UBX-CFG-USB.vendorString	CFG-USB-VENDOR_STR0, CFG-USB-VENDOR_STR1, CFG-USB-VENDOR_STR2, CFG-USB-VENDOR_STR3

Table 72: Legacy UBX message fields and the corresponding configuration items

Configuration defaults

The following tables contain the configuration defaults for the firmware. Some of these values may be changed in production. Refer to the integration manual for product-specific details.

Configuration item	Key ID	Type	Scale	Unit	Default value
CFG-BDS-USE_GEO_PRN	0x10340014	L	-	-	0 (false)

Table 73: CFG-BDS configuration defaults

Configuration item	Key ID	Type	Scale	Unit	Default value
CFG-GEOFENCE-CONFLVL	0x20240011	E1	-	-	0 (L000)
CFG-GEOFENCE-USE_PIO	0x10240012	L	-	-	0 (false)
CFG-GEOFENCE-PINPOL	0x20240013	E1	-	-	0 (LOW_IN)
CFG-GEOFENCE-PIN	0x20240014	U1	-	-	3
CFG-GEOFENCE-USE_FENCE1	0x10240020	L	-	-	0 (false)
CFG-GEOFENCE-FENCE1_LAT	0x40240021	I4	1e-7	deg	0
CFG-GEOFENCE-FENCE1_LON	0x40240022	I4	1e-7	deg	0
CFG-GEOFENCE-FENCE1_RAD	0x40240023	U4	0.01	m	0
CFG-GEOFENCE-USE_FENCE2	0x10240030	L	-	-	0 (false)
CFG-GEOFENCE-FENCE2_LAT	0x40240031	I4	1e-7	deg	0
CFG-GEOFENCE-FENCE2_LON	0x40240032	I4	1e-7	deg	0
CFG-GEOFENCE-FENCE2_RAD	0x40240033	U4	0.01	m	0
CFG-GEOFENCE-USE_FENCE3	0x10240040	L	-	-	0 (false)
CFG-GEOFENCE-FENCE3_LAT	0x40240041	I4	1e-7	deg	0
CFG-GEOFENCE-FENCE3_LON	0x40240042	I4	1e-7	deg	0
CFG-GEOFENCE-FENCE3_RAD	0x40240043	U4	0.01	m	0
CFG-GEOFENCE-USE_FENCE4	0x10240050	L	-	-	0 (false)
CFG-GEOFENCE-FENCE4_LAT	0x40240051	I4	1e-7	deg	0
CFG-GEOFENCE-FENCE4_LON	0x40240052	I4	1e-7	deg	0
CFG-GEOFENCE-FENCE4_RAD	0x40240053	U4	0.01	m	0

Table 74: CFG-GEOFENCE configuration defaults

Configuration item	Key ID	Type	Scale	Unit	Default value
CFG-HW-ANT_CFG_VOLTCTRL	0x10a3002e	L	-	-	0 (false)
CFG-HW-ANT_CFG_SHORTDET	0x10a3002f	L	-	-	0 (false)
CFG-HW-ANT_CFG_SHORTDET_POL	0x10a30030	L	-	-	1 (true)
CFG-HW-ANT_CFG_OPENDET	0x10a30031	L	-	-	0 (false)
CFG-HW-ANT_CFG_OPENDET_POL	0x10a30032	L	-	-	1 (true)
CFG-HW-ANT_CFG_PWRDOWN	0x10a30033	L	-	-	0 (false)
CFG-HW-ANT_CFG_PWRDOWN_POL	0x10a30034	L	-	-	1 (true)
CFG-HW-ANT_CFG_RECOVER	0x10a30035	L	-	-	0 (false)
CFG-HW-ANT_SUP_SWITCH_PIN	0x20a30036	U1	-	-	16
CFG-HW-ANT_SUP_SHORT_PIN	0x20a30037	U1	-	-	15
CFG-HW-ANT_SUP_OPEN_PIN	0x20a30038	U1	-	-	8

Configuration item	Key ID	Type	Scale	Unit	Default value
CFG-HW-ANT_SUP_ENGINE	0x20a30054	E1	-	-	0 (EXT)
CFG-HW-ANT_SUP_SHORT_THR	0x20a30055	U1	-	mV	0
CFG-HW-ANT_SUP_OPEN_THR	0x20a30056	U1	-	mV	0

Table 75: CFG-HW configuration defaults

Configuration item	Key ID	Type	Scale	Unit	Default value
CFG-I2C-ADDRESS	0x20510001	U1	-	-	132
CFG-I2C-EXTENDEDTIMEOUT	0x10510002	L	-	-	0 (false)
CFG-I2C-ENABLED	0x10510003	L	-	-	1 (true)

Table 76: CFG-I2C configuration defaults

Configuration item	Key ID	Type	Scale	Unit	Default value
CFG-I2CINPROT-UBX	0x10710001	L	-	-	1 (true)
CFG-I2CINPROT-NMEA	0x10710002	L	-	-	1 (true)
CFG-I2CINPROT-RTCM3X	0x10710004	L	-	-	1 (true)
CFG-I2CINPROT-SPARTN	0x10710005	L	-	-	1 (true)

Table 77: CFG-I2CINPROT configuration defaults

Configuration item	Key ID	Type	Scale	Unit	Default value
CFG-I2COUTPROT-UBX	0x10720001	L	-	-	1 (true)
CFG-I2COUTPROT-NMEA	0x10720002	L	-	-	1 (true)
CFG-I2COUTPROT-RTCM3X	0x10720004	L	-	-	1 (true)

Table 78: CFG-I2COUTPROT configuration defaults

Configuration item	Key ID	Type	Scale	Unit	Default value
CFG-INFMSG-UBX_I2C	0x20920001	X1	-	-	0x00
CFG-INFMSG-UBX_UART1	0x20920002	X1	-	-	0x00
CFG-INFMSG-UBX_UART2	0x20920003	X1	-	-	0x00
CFG-INFMSG-UBX_USB	0x20920004	X1	-	-	0x00
CFG-INFMSG-UBX_SPI	0x20920005	X1	-	-	0x00
CFG-INFMSG-NMEA_I2C	0x20920006	X1	-	-	0x07 (ERROR | WARNING | NOTICE)
CFG-INFMSG-NMEA_UART1	0x20920007	X1	-	-	0x07 (ERROR | WARNING | NOTICE)
CFG-INFMSG-NMEA_UART2	0x20920008	X1	-	-	0x07 (ERROR | WARNING | NOTICE)
CFG-INFMSG-NMEA_USB	0x20920009	X1	-	-	0x07 (ERROR | WARNING | NOTICE)
CFG-INFMSG-NMEA_SPI	0x2092000a	X1	-	-	0x07 (ERROR | WARNING | NOTICE)

Table 79: CFG-INFMSG configuration defaults

Configuration item	Key ID	Type	Scale	Unit	Default value
CFG-ITFM-BBTHRESHOLD	0x20410001	U1	-	-	3
CFG-ITFM-CWTHRESHOLD	0x20410002	U1	-	-	15
CFG-ITFM-ENABLE	0x1041000d	L	-	-	0 (false)
CFG-ITFM-ANTSETTING	0x20410010	E1	-	-	0 (UNKNOWN)

Configuration item	Key ID	Type	Scale	Unit	Default value
CFG-ITFM-ENABLE_AUX	0x10410013	L	-	-	0 (false)

Table 80: CFG-ITFM configuration defaults

Configuration item	Key ID	Type	Scale	Unit	Default value
CFG-LOGFILTER-RECORD_ENA	0x10de0002	L	-	-	0 (false)
CFG-LOGFILTER-ONCE_PER_WAKE_UP_ENA	0x10de0003	L	-	-	0 (false)
CFG-LOGFILTER-APPLY_ALL_FILTERS	0x10de0004	L	-	-	0 (false)
CFG-LOGFILTER-MIN_INTERVAL	0x30de0005	U2	-	s	0
CFG-LOGFILTER-TIME_THRS	0x30de0006	U2	-	s	0
CFG-LOGFILTER-SPEED_THRS	0x30de0007	U2	-	m/s	0
CFG-LOGFILTER-POSITION_THRS	0x40de0008	U4	-	m	0

Table 81: CFG-LOGFILTER configuration defaults

Configuration item	Key ID	Type	Scale	Unit	Default value
CFG-MOT-GNSSSPEED_THRS	0x20250038	U1	0.01	m/s	0
CFG-MOT-GNSSDIST_THRS	0x3025003b	U2	-	-	0

Table 82: CFG-MOT configuration defaults

Configuration item	Key ID	Type	Scale	Unit	Default value
CFG-MSGOUT-NMEA_ID_DTM_I2C	0x209100a6	U1	-	-	0
CFG-MSGOUT-NMEA_ID_DTM_SPI	0x209100aa	U1	-	-	0
CFG-MSGOUT-NMEA_ID_DTM_UART1	0x209100a7	U1	-	-	0
CFG-MSGOUT-NMEA_ID_DTM_UART2	0x209100a8	U1	-	-	0
CFG-MSGOUT-NMEA_ID_DTM_USB	0x209100a9	U1	-	-	0
CFG-MSGOUT-NMEA_ID_GBS_I2C	0x209100dd	U1	-	-	0
CFG-MSGOUT-NMEA_ID_GBS_SPI	0x209100e1	U1	-	-	0
CFG-MSGOUT-NMEA_ID_GBS_UART1	0x209100de	U1	-	-	0
CFG-MSGOUT-NMEA_ID_GBS_UART2	0x209100df	U1	-	-	0
CFG-MSGOUT-NMEA_ID_GBS_USB	0x209100e0	U1	-	-	0
CFG-MSGOUT-NMEA_ID_GGA_I2C	0x209100ba	U1	-	-	1
CFG-MSGOUT-NMEA_ID_GGA_SPI	0x209100be	U1	-	-	1
CFG-MSGOUT-NMEA_ID_GGA_UART1	0x209100bb	U1	-	-	1
CFG-MSGOUT-NMEA_ID_GGA_UART2	0x209100bc	U1	-	-	1
CFG-MSGOUT-NMEA_ID_GGA_USB	0x209100bd	U1	-	-	1
CFG-MSGOUT-NMEA_ID_GLL_I2C	0x209100c9	U1	-	-	1
CFG-MSGOUT-NMEA_ID_GLL_SPI	0x209100cd	U1	-	-	1
CFG-MSGOUT-NMEA_ID_GLL_UART1	0x209100ca	U1	-	-	1
CFG-MSGOUT-NMEA_ID_GLL_UART2	0x209100cb	U1	-	-	1
CFG-MSGOUT-NMEA_ID_GLL_USB	0x209100cc	U1	-	-	1
CFG-MSGOUT-NMEA_ID_GNS_I2C	0x209100b5	U1	-	-	0
CFG-MSGOUT-NMEA_ID_GNS_SPI	0x209100b9	U1	-	-	0
CFG-MSGOUT-NMEA_ID_GNS_UART1	0x209100b6	U1	-	-	0
CFG-MSGOUT-NMEA_ID_GNS_UART2	0x209100b7	U1	-	-	0

Configuration item	Key ID	Type	Scale	Unit	Default value
CFG-MSGOUT-NMEA_ID_GNS_USB	0x209100b8	U1	-	-	0
CFG-MSGOUT-NMEA_ID_GRS_I2C	0x209100ce	U1	-	-	0
CFG-MSGOUT-NMEA_ID_GRS_SPI	0x209100d2	U1	-	-	0
CFG-MSGOUT-NMEA_ID_GRS_UART1	0x209100cf	U1	-	-	0
CFG-MSGOUT-NMEA_ID_GRS_UART2	0x209100d0	U1	-	-	0
CFG-MSGOUT-NMEA_ID_GRS_USB	0x209100d1	U1	-	-	0
CFG-MSGOUT-NMEA_ID_GSA_I2C	0x209100bf	U1	-	-	1
CFG-MSGOUT-NMEA_ID_GSA_SPI	0x209100c3	U1	-	-	1
CFG-MSGOUT-NMEA_ID_GSA_UART1	0x209100c0	U1	-	-	1
CFG-MSGOUT-NMEA_ID_GSA_UART2	0x209100c1	U1	-	-	1
CFG-MSGOUT-NMEA_ID_GSA_USB	0x209100c2	U1	-	-	1
CFG-MSGOUT-NMEA_ID_GST_I2C	0x209100d3	U1	-	-	0
CFG-MSGOUT-NMEA_ID_GST_SPI	0x209100d7	U1	-	-	0
CFG-MSGOUT-NMEA_ID_GST_UART1	0x209100d4	U1	-	-	0
CFG-MSGOUT-NMEA_ID_GST_UART2	0x209100d5	U1	-	-	0
CFG-MSGOUT-NMEA_ID_GST_USB	0x209100d6	U1	-	-	0
CFG-MSGOUT-NMEA_ID_GSV_I2C	0x209100c4	U1	-	-	1
CFG-MSGOUT-NMEA_ID_GSV_SPI	0x209100c8	U1	-	-	1
CFG-MSGOUT-NMEA_ID_GSV_UART1	0x209100c5	U1	-	-	1
CFG-MSGOUT-NMEA_ID_GSV_UART2	0x209100c6	U1	-	-	1
CFG-MSGOUT-NMEA_ID_GSV_USB	0x209100c7	U1	-	-	1
CFG-MSGOUT-NMEA_ID_RLM_I2C	0x20910400	U1	-	-	0
CFG-MSGOUT-NMEA_ID_RLM_SPI	0x20910404	U1	-	-	0
CFG-MSGOUT-NMEA_ID_RLM_UART1	0x20910401	U1	-	-	0
CFG-MSGOUT-NMEA_ID_RLM_UART2	0x20910402	U1	-	-	0
CFG-MSGOUT-NMEA_ID_RLM_USB	0x20910403	U1	-	-	0
CFG-MSGOUT-NMEA_ID_RMC_I2C	0x209100ab	U1	-	-	1
CFG-MSGOUT-NMEA_ID_RMC_SPI	0x209100af	U1	-	-	1
CFG-MSGOUT-NMEA_ID_RMC_UART1	0x209100ac	U1	-	-	1
CFG-MSGOUT-NMEA_ID_RMC_UART2	0x209100ad	U1	-	-	1
CFG-MSGOUT-NMEA_ID_RMC_USB	0x209100ae	U1	-	-	1
CFG-MSGOUT-NMEA_ID_VLW_I2C	0x209100e7	U1	-	-	0
CFG-MSGOUT-NMEA_ID_VLW_SPI	0x209100eb	U1	-	-	0
CFG-MSGOUT-NMEA_ID_VLW_UART1	0x209100e8	U1	-	-	0
CFG-MSGOUT-NMEA_ID_VLW_UART2	0x209100e9	U1	-	-	0
CFG-MSGOUT-NMEA_ID_VLW_USB	0x209100ea	U1	-	-	0
CFG-MSGOUT-NMEA_ID_VTG_I2C	0x209100b0	U1	-	-	1
CFG-MSGOUT-NMEA_ID_VTG_SPI	0x209100b4	U1	-	-	1
CFG-MSGOUT-NMEA_ID_VTG_UART1	0x209100b1	U1	-	-	1
CFG-MSGOUT-NMEA_ID_VTG_UART2	0x209100b2	U1	-	-	1
CFG-MSGOUT-NMEA_ID_VTG_USB	0x209100b3	U1	-	-	1

Configuration item	Key ID	Type	Scale	Unit	Default value
CFG-MSGOUT-NMEA_ID_ZDA_I2C	0x209100d8	U1	-	-	0
CFG-MSGOUT-NMEA_ID_ZDA_SPI	0x209100dc	U1	-	-	0
CFG-MSGOUT-NMEA_ID_ZDA_UART1	0x209100d9	U1	-	-	0
CFG-MSGOUT-NMEA_ID_ZDA_UART2	0x209100da	U1	-	-	0
CFG-MSGOUT-NMEA_ID_ZDA_USB	0x209100db	U1	-	-	0
CFG-MSGOUT-NMEA_NAV2_ID_GGA_I2C	0x20910661	U1	-	-	0
CFG-MSGOUT-NMEA_NAV2_ID_GGA_SPI	0x20910665	U1	-	-	0
CFG-MSGOUT-NMEA_NAV2_ID_GGA_UART1	0x20910662	U1	-	-	0
CFG-MSGOUT-NMEA_NAV2_ID_GGA_UART2	0x20910663	U1	-	-	0
CFG-MSGOUT-NMEA_NAV2_ID_GGA_USB	0x20910664	U1	-	-	0
CFG-MSGOUT-NMEA_NAV2_ID_GLL_I2C	0x20910670	U1	-	-	0
CFG-MSGOUT-NMEA_NAV2_ID_GLL_SPI	0x20910674	U1	-	-	0
CFG-MSGOUT-NMEA_NAV2_ID_GLL_UART1	0x20910671	U1	-	-	0
CFG-MSGOUT-NMEA_NAV2_ID_GLL_UART2	0x20910672	U1	-	-	0
CFG-MSGOUT-NMEA_NAV2_ID_GLL_USB	0x20910673	U1	-	-	0
CFG-MSGOUT-NMEA_NAV2_ID_GNS_I2C	0x2091065c	U1	-	-	0
CFG-MSGOUT-NMEA_NAV2_ID_GNS_SPI	0x20910660	U1	-	-	0
CFG-MSGOUT-NMEA_NAV2_ID_GNS_UART1	0x2091065d	U1	-	-	0
CFG-MSGOUT-NMEA_NAV2_ID_GNS_UART2	0x2091065e	U1	-	-	0
CFG-MSGOUT-NMEA_NAV2_ID_GNS_USB	0x2091065f	U1	-	-	0
CFG-MSGOUT-NMEA_NAV2_ID_GSA_I2C	0x20910666	U1	-	-	0
CFG-MSGOUT-NMEA_NAV2_ID_GSA_SPI	0x2091066a	U1	-	-	0
CFG-MSGOUT-NMEA_NAV2_ID_GSA_UART1	0x20910667	U1	-	-	0
CFG-MSGOUT-NMEA_NAV2_ID_GSA_UART2	0x20910668	U1	-	-	0
CFG-MSGOUT-NMEA_NAV2_ID_GSA_USB	0x20910669	U1	-	-	0
CFG-MSGOUT-NMEA_NAV2_ID_RMC_I2C	0x20910652	U1	-	-	0
CFG-MSGOUT-NMEA_NAV2_ID_RMC_SPI	0x20910656	U1	-	-	0
CFG-MSGOUT-NMEA_NAV2_ID_RMC_UART1	0x20910653	U1	-	-	0
CFG-MSGOUT-NMEA_NAV2_ID_RMC_UART2	0x20910654	U1	-	-	0
CFG-MSGOUT-NMEA_NAV2_ID_RMC_USB	0x20910655	U1	-	-	0
CFG-MSGOUT-NMEA_NAV2_ID_VTG_I2C	0x20910657	U1	-	-	0
CFG-MSGOUT-NMEA_NAV2_ID_VTG_SPI	0x2091065b	U1	-	-	0
CFG-MSGOUT-NMEA_NAV2_ID_VTG_UART1	0x20910658	U1	-	-	0
CFG-MSGOUT-NMEA_NAV2_ID_VTG_UART2	0x20910659	U1	-	-	0
CFG-MSGOUT-NMEA_NAV2_ID_VTG_USB	0x2091065a	U1	-	-	0
CFG-MSGOUT-NMEA_NAV2_ID_ZDA_I2C	0x2091067f	U1	-	-	0
CFG-MSGOUT-NMEA_NAV2_ID_ZDA_SPI	0x20910683	U1	-	-	0
CFG-MSGOUT-NMEA_NAV2_ID_ZDA_UART1	0x20910680	U1	-	-	0
CFG-MSGOUT-NMEA_NAV2_ID_ZDA_UART2	0x20910681	U1	-	-	0
CFG-MSGOUT-NMEA_NAV2_ID_ZDA_USB	0x20910682	U1	-	-	0
CFG-MSGOUT-PUBX_ID_POLYP_I2C	0x209100ec	U1	-	-	0

Configuration item	Key ID	Type	Scale	Unit	Default value
CFG-MSGOUT-PUBX_ID_POLYP_SPI	0x209100f0	U1	-	-	0
CFG-MSGOUT-PUBX_ID_POLYP_UART1	0x209100ed	U1	-	-	0
CFG-MSGOUT-PUBX_ID_POLYP_UART2	0x209100ee	U1	-	-	0
CFG-MSGOUT-PUBX_ID_POLYP_USB	0x209100ef	U1	-	-	0
CFG-MSGOUT-PUBX_ID_POLYS_I2C	0x209100f1	U1	-	-	0
CFG-MSGOUT-PUBX_ID_POLYS_SPI	0x209100f5	U1	-	-	0
CFG-MSGOUT-PUBX_ID_POLYS_UART1	0x209100f2	U1	-	-	0
CFG-MSGOUT-PUBX_ID_POLYS_UART2	0x209100f3	U1	-	-	0
CFG-MSGOUT-PUBX_ID_POLYS_USB	0x209100f4	U1	-	-	0
CFG-MSGOUT-PUBX_ID_POLYT_I2C	0x209100f6	U1	-	-	0
CFG-MSGOUT-PUBX_ID_POLYT_SPI	0x209100fa	U1	-	-	0
CFG-MSGOUT-PUBX_ID_POLYT_UART1	0x209100f7	U1	-	-	0
CFG-MSGOUT-PUBX_ID_POLYT_UART2	0x209100f8	U1	-	-	0
CFG-MSGOUT-PUBX_ID_POLYT_USB	0x209100f9	U1	-	-	0
CFG-MSGOUT-RTCM_3X_TYPE1005_I2C	0x209102bd	U1	-	-	0
CFG-MSGOUT-RTCM_3X_TYPE1005_SPI	0x209102c1	U1	-	-	0
CFG-MSGOUT-RTCM_3X_TYPE1005_UART1	0x209102be	U1	-	-	0
CFG-MSGOUT-RTCM_3X_TYPE1005_UART2	0x209102bf	U1	-	-	0
CFG-MSGOUT-RTCM_3X_TYPE1005_USB	0x209102c0	U1	-	-	0
CFG-MSGOUT-RTCM_3X_TYPE1074_I2C	0x2091035e	U1	-	-	0
CFG-MSGOUT-RTCM_3X_TYPE1074_SPI	0x20910362	U1	-	-	0
CFG-MSGOUT-RTCM_3X_TYPE1074_UART1	0x2091035f	U1	-	-	0
CFG-MSGOUT-RTCM_3X_TYPE1074_UART2	0x20910360	U1	-	-	0
CFG-MSGOUT-RTCM_3X_TYPE1074_USB	0x20910361	U1	-	-	0
CFG-MSGOUT-RTCM_3X_TYPE1077_I2C	0x209102cc	U1	-	-	0
CFG-MSGOUT-RTCM_3X_TYPE1077_SPI	0x209102d0	U1	-	-	0
CFG-MSGOUT-RTCM_3X_TYPE1077_UART1	0x209102cd	U1	-	-	0
CFG-MSGOUT-RTCM_3X_TYPE1077_UART2	0x209102ce	U1	-	-	0
CFG-MSGOUT-RTCM_3X_TYPE1077_USB	0x209102cf	U1	-	-	0
CFG-MSGOUT-RTCM_3X_TYPE1084_I2C	0x20910363	U1	-	-	0
CFG-MSGOUT-RTCM_3X_TYPE1084_SPI	0x20910367	U1	-	-	0
CFG-MSGOUT-RTCM_3X_TYPE1084_UART1	0x20910364	U1	-	-	0
CFG-MSGOUT-RTCM_3X_TYPE1084_UART2	0x20910365	U1	-	-	0
CFG-MSGOUT-RTCM_3X_TYPE1084_USB	0x20910366	U1	-	-	0
CFG-MSGOUT-RTCM_3X_TYPE1087_I2C	0x209102d1	U1	-	-	0
CFG-MSGOUT-RTCM_3X_TYPE1087_SPI	0x209102d5	U1	-	-	0
CFG-MSGOUT-RTCM_3X_TYPE1087_UART1	0x209102d2	U1	-	-	0
CFG-MSGOUT-RTCM_3X_TYPE1087_UART2	0x209102d3	U1	-	-	0
CFG-MSGOUT-RTCM_3X_TYPE1087_USB	0x209102d4	U1	-	-	0
CFG-MSGOUT-RTCM_3X_TYPE1094_I2C	0x20910368	U1	-	-	0
CFG-MSGOUT-RTCM_3X_TYPE1094_SPI	0x2091036c	U1	-	-	0

Configuration item	Key ID	Type	Scale	Unit	Default value
CFG-MSGOUT-RTCM_3X_TYPE1094_UART1	0x20910369	U1	-	-	0
CFG-MSGOUT-RTCM_3X_TYPE1094_UART2	0x2091036a	U1	-	-	0
CFG-MSGOUT-RTCM_3X_TYPE1094_USB	0x2091036b	U1	-	-	0
CFG-MSGOUT-RTCM_3X_TYPE1097_I2C	0x20910318	U1	-	-	0
CFG-MSGOUT-RTCM_3X_TYPE1097_SPI	0x2091031c	U1	-	-	0
CFG-MSGOUT-RTCM_3X_TYPE1097_UART1	0x20910319	U1	-	-	0
CFG-MSGOUT-RTCM_3X_TYPE1097_UART2	0x2091031a	U1	-	-	0
CFG-MSGOUT-RTCM_3X_TYPE1097_USB	0x2091031b	U1	-	-	0
CFG-MSGOUT-RTCM_3X_TYPE1124_I2C	0x2091036d	U1	-	-	0
CFG-MSGOUT-RTCM_3X_TYPE1124_SPI	0x20910371	U1	-	-	0
CFG-MSGOUT-RTCM_3X_TYPE1124_UART1	0x2091036e	U1	-	-	0
CFG-MSGOUT-RTCM_3X_TYPE1124_UART2	0x2091036f	U1	-	-	0
CFG-MSGOUT-RTCM_3X_TYPE1124_USB	0x20910370	U1	-	-	0
CFG-MSGOUT-RTCM_3X_TYPE1127_I2C	0x209102d6	U1	-	-	0
CFG-MSGOUT-RTCM_3X_TYPE1127_SPI	0x209102da	U1	-	-	0
CFG-MSGOUT-RTCM_3X_TYPE1127_UART1	0x209102d7	U1	-	-	0
CFG-MSGOUT-RTCM_3X_TYPE1127_UART2	0x209102d8	U1	-	-	0
CFG-MSGOUT-RTCM_3X_TYPE1127_USB	0x209102d9	U1	-	-	0
CFG-MSGOUT-RTCM_3X_TYPE1230_I2C	0x20910303	U1	-	-	0
CFG-MSGOUT-RTCM_3X_TYPE1230_SPI	0x20910307	U1	-	-	0
CFG-MSGOUT-RTCM_3X_TYPE1230_UART1	0x20910304	U1	-	-	0
CFG-MSGOUT-RTCM_3X_TYPE1230_UART2	0x20910305	U1	-	-	0
CFG-MSGOUT-RTCM_3X_TYPE1230_USB	0x20910306	U1	-	-	0
CFG-MSGOUT-RTCM_3X_TYPE4072_0_I2C	0x209102fe	U1	-	-	0
CFG-MSGOUT-RTCM_3X_TYPE4072_0_SPI	0x20910302	U1	-	-	0
CFG-MSGOUT-RTCM_3X_TYPE4072_0_UART1	0x209102ff	U1	-	-	0
CFG-MSGOUT-RTCM_3X_TYPE4072_0_UART2	0x20910300	U1	-	-	0
CFG-MSGOUT-RTCM_3X_TYPE4072_0_USB	0x20910301	U1	-	-	0
CFG-MSGOUT-RTCM_3X_TYPE4072_1_I2C	0x20910381	U1	-	-	0
CFG-MSGOUT-RTCM_3X_TYPE4072_1_SPI	0x20910385	U1	-	-	0
CFG-MSGOUT-RTCM_3X_TYPE4072_1_UART1	0x20910382	U1	-	-	0
CFG-MSGOUT-RTCM_3X_TYPE4072_1_UART2	0x20910383	U1	-	-	0
CFG-MSGOUT-RTCM_3X_TYPE4072_1_USB	0x20910384	U1	-	-	0
CFG-MSGOUT-UBX_LOG_INFO_I2C	0x20910259	U1	-	-	0
CFG-MSGOUT-UBX_LOG_INFO_SPI	0x2091025d	U1	-	-	0
CFG-MSGOUT-UBX_LOG_INFO_UART1	0x2091025a	U1	-	-	0
CFG-MSGOUT-UBX_LOG_INFO_UART2	0x2091025b	U1	-	-	0
CFG-MSGOUT-UBX_LOG_INFO_USB	0x2091025c	U1	-	-	0
CFG-MSGOUT-UBX_MON_COMMS_I2C	0x2091034f	U1	-	-	0
CFG-MSGOUT-UBX_MON_COMMS_SPI	0x20910353	U1	-	-	0
CFG-MSGOUT-UBX_MON_COMMS_UART1	0x20910350	U1	-	-	0

Configuration item	Key ID	Type	Scale	Unit	Default value
CFG-MSGOUT-UBX_MON_COMMS_UART2	0x20910351	U1	-	-	0
CFG-MSGOUT-UBX_MON_COMMS_USB	0x20910352	U1	-	-	0
CFG-MSGOUT-UBX_MON_HW2_I2C	0x209101b9	U1	-	-	0
CFG-MSGOUT-UBX_MON_HW2_SPI	0x209101bd	U1	-	-	0
CFG-MSGOUT-UBX_MON_HW2_UART1	0x209101ba	U1	-	-	0
CFG-MSGOUT-UBX_MON_HW2_UART2	0x209101bb	U1	-	-	0
CFG-MSGOUT-UBX_MON_HW2_USB	0x209101bc	U1	-	-	0
CFG-MSGOUT-UBX_MON_HW3_I2C	0x20910354	U1	-	-	0
CFG-MSGOUT-UBX_MON_HW3_SPI	0x20910358	U1	-	-	0
CFG-MSGOUT-UBX_MON_HW3_UART1	0x20910355	U1	-	-	0
CFG-MSGOUT-UBX_MON_HW3_UART2	0x20910356	U1	-	-	0
CFG-MSGOUT-UBX_MON_HW3_USB	0x20910357	U1	-	-	0
CFG-MSGOUT-UBX_MON_HW_I2C	0x209101b4	U1	-	-	0
CFG-MSGOUT-UBX_MON_HW_SPI	0x209101b8	U1	-	-	0
CFG-MSGOUT-UBX_MON_HW_UART1	0x209101b5	U1	-	-	0
CFG-MSGOUT-UBX_MON_HW_UART2	0x209101b6	U1	-	-	0
CFG-MSGOUT-UBX_MON_HW_USB	0x209101b7	U1	-	-	0
CFG-MSGOUT-UBX_MON_IO_I2C	0x209101a5	U1	-	-	0
CFG-MSGOUT-UBX_MON_IO_SPI	0x209101a9	U1	-	-	0
CFG-MSGOUT-UBX_MON_IO_UART1	0x209101a6	U1	-	-	0
CFG-MSGOUT-UBX_MON_IO_UART2	0x209101a7	U1	-	-	0
CFG-MSGOUT-UBX_MON_IO_USB	0x209101a8	U1	-	-	0
CFG-MSGOUT-UBX_MON_MSGPP_I2C	0x20910196	U1	-	-	0
CFG-MSGOUT-UBX_MON_MSGPP_SPI	0x2091019a	U1	-	-	0
CFG-MSGOUT-UBX_MON_MSGPP_UART1	0x20910197	U1	-	-	0
CFG-MSGOUT-UBX_MON_MSGPP_UART2	0x20910198	U1	-	-	0
CFG-MSGOUT-UBX_MON_MSGPP_USB	0x20910199	U1	-	-	0
CFG-MSGOUT-UBX_MON_RF_I2C	0x20910359	U1	-	-	0
CFG-MSGOUT-UBX_MON_RF_SPI	0x2091035d	U1	-	-	0
CFG-MSGOUT-UBX_MON_RF_UART1	0x2091035a	U1	-	-	0
CFG-MSGOUT-UBX_MON_RF_UART2	0x2091035b	U1	-	-	0
CFG-MSGOUT-UBX_MON_RF_USB	0x2091035c	U1	-	-	0
CFG-MSGOUT-UBX_MON_RXBUF_I2C	0x209101a0	U1	-	-	0
CFG-MSGOUT-UBX_MON_RXBUF_SPI	0x209101a4	U1	-	-	0
CFG-MSGOUT-UBX_MON_RXBUF_UART1	0x209101a1	U1	-	-	0
CFG-MSGOUT-UBX_MON_RXBUF_UART2	0x209101a2	U1	-	-	0
CFG-MSGOUT-UBX_MON_RXBUF_USB	0x209101a3	U1	-	-	0
CFG-MSGOUT-UBX_MON_RXR_I2C	0x20910187	U1	-	-	0
CFG-MSGOUT-UBX_MON_RXR_SPI	0x2091018b	U1	-	-	0
CFG-MSGOUT-UBX_MON_RXR_UART1	0x20910188	U1	-	-	0
CFG-MSGOUT-UBX_MON_RXR_UART2	0x20910189	U1	-	-	0

Configuration item	Key ID	Type	Scale	Unit	Default value
CFG-MSGOUT-UBX_MON_RXR_USB	0x2091018a	U1	-	-	0
CFG-MSGOUT-UBX_MON_SPAN_I2C	0x2091038b	U1	-	-	0
CFG-MSGOUT-UBX_MON_SPAN_SPI	0x2091038f	U1	-	-	0
CFG-MSGOUT-UBX_MON_SPAN_UART1	0x2091038c	U1	-	-	0
CFG-MSGOUT-UBX_MON_SPAN_UART2	0x2091038d	U1	-	-	0
CFG-MSGOUT-UBX_MON_SPAN_USB	0x2091038e	U1	-	-	0
CFG-MSGOUT-UBX_MON_SYS_I2C	0x2091069d	U1	-	-	0
CFG-MSGOUT-UBX_MON_SYS_SPI	0x209106a1	U1	-	-	0
CFG-MSGOUT-UBX_MON_SYS_UART1	0x2091069e	U1	-	-	0
CFG-MSGOUT-UBX_MON_SYS_UART2	0x2091069f	U1	-	-	0
CFG-MSGOUT-UBX_MON_SYS_USB	0x209106a0	U1	-	-	0
CFG-MSGOUT-UBX_MON_TXBUF_I2C	0x2091019b	U1	-	-	0
CFG-MSGOUT-UBX_MON_TXBUF_SPI	0x2091019f	U1	-	-	0
CFG-MSGOUT-UBX_MON_TXBUF_UART1	0x2091019c	U1	-	-	0
CFG-MSGOUT-UBX_MON_TXBUF_UART2	0x2091019d	U1	-	-	0
CFG-MSGOUT-UBX_MON_TXBUF_USB	0x2091019e	U1	-	-	0
CFG-MSGOUT-UBX_NAV2_CLOCK_I2C	0x20910430	U1	-	-	0
CFG-MSGOUT-UBX_NAV2_CLOCK_SPI	0x20910434	U1	-	-	0
CFG-MSGOUT-UBX_NAV2_CLOCK_UART1	0x20910431	U1	-	-	0
CFG-MSGOUT-UBX_NAV2_CLOCK_UART2	0x20910432	U1	-	-	0
CFG-MSGOUT-UBX_NAV2_CLOCK_USB	0x20910433	U1	-	-	0
CFG-MSGOUT-UBX_NAV2_COV_I2C	0x20910435	U1	-	-	0
CFG-MSGOUT-UBX_NAV2_COV_SPI	0x20910439	U1	-	-	0
CFG-MSGOUT-UBX_NAV2_COV_UART1	0x20910436	U1	-	-	0
CFG-MSGOUT-UBX_NAV2_COV_UART2	0x20910437	U1	-	-	0
CFG-MSGOUT-UBX_NAV2_COV_USB	0x20910438	U1	-	-	0
CFG-MSGOUT-UBX_NAV2_DOP_I2C	0x20910465	U1	-	-	0
CFG-MSGOUT-UBX_NAV2_DOP_SPI	0x20910469	U1	-	-	0
CFG-MSGOUT-UBX_NAV2_DOP_UART1	0x20910466	U1	-	-	0
CFG-MSGOUT-UBX_NAV2_DOP_UART2	0x20910467	U1	-	-	0
CFG-MSGOUT-UBX_NAV2_DOP_USB	0x20910468	U1	-	-	0
CFG-MSGOUT-UBX_NAV2_EOE_I2C	0x20910565	U1	-	-	0
CFG-MSGOUT-UBX_NAV2_EOE_SPI	0x20910569	U1	-	-	0
CFG-MSGOUT-UBX_NAV2_EOE_UART1	0x20910566	U1	-	-	0
CFG-MSGOUT-UBX_NAV2_EOE_UART2	0x20910567	U1	-	-	0
CFG-MSGOUT-UBX_NAV2_EOE_USB	0x20910568	U1	-	-	0
CFG-MSGOUT-UBX_NAV2_ODO_I2C	0x20910475	U1	-	-	0
CFG-MSGOUT-UBX_NAV2_ODO_SPI	0x20910479	U1	-	-	0
CFG-MSGOUT-UBX_NAV2_ODO_UART1	0x20910476	U1	-	-	0
CFG-MSGOUT-UBX_NAV2_ODO_UART2	0x20910477	U1	-	-	0
CFG-MSGOUT-UBX_NAV2_ODO_USB	0x20910478	U1	-	-	0

Configuration item	Key ID	Type	Scale	Unit	Default value
CFG-MSGOUT-UBX_NAV2_POSECEF_I2C	0x20910480	U1	-	-	0
CFG-MSGOUT-UBX_NAV2_POSECEF_SPI	0x20910484	U1	-	-	0
CFG-MSGOUT-UBX_NAV2_POSECEF_UART1	0x20910481	U1	-	-	0
CFG-MSGOUT-UBX_NAV2_POSECEF_UART2	0x20910482	U1	-	-	0
CFG-MSGOUT-UBX_NAV2_POSECEF_USB	0x20910483	U1	-	-	0
CFG-MSGOUT-UBX_NAV2_POSLLH_I2C	0x20910485	U1	-	-	0
CFG-MSGOUT-UBX_NAV2_POSLLH_SPI	0x20910489	U1	-	-	0
CFG-MSGOUT-UBX_NAV2_POSLLH_UART1	0x20910486	U1	-	-	0
CFG-MSGOUT-UBX_NAV2_POSLLH_UART2	0x20910487	U1	-	-	0
CFG-MSGOUT-UBX_NAV2_POSLLH_USB	0x20910488	U1	-	-	0
CFG-MSGOUT-UBX_NAV2_PVT_I2C	0x20910490	U1	-	-	0
CFG-MSGOUT-UBX_NAV2_PVT_SPI	0x20910494	U1	-	-	0
CFG-MSGOUT-UBX_NAV2_PVT_UART1	0x20910491	U1	-	-	0
CFG-MSGOUT-UBX_NAV2_PVT_UART2	0x20910492	U1	-	-	0
CFG-MSGOUT-UBX_NAV2_PVT_USB	0x20910493	U1	-	-	0
CFG-MSGOUT-UBX_NAV2_SAT_I2C	0x20910495	U1	-	-	0
CFG-MSGOUT-UBX_NAV2_SAT_SPI	0x20910499	U1	-	-	0
CFG-MSGOUT-UBX_NAV2_SAT_UART1	0x20910496	U1	-	-	0
CFG-MSGOUT-UBX_NAV2_SAT_UART2	0x20910497	U1	-	-	0
CFG-MSGOUT-UBX_NAV2_SAT_USB	0x20910498	U1	-	-	0
CFG-MSGOUT-UBX_NAV2_SBAS_I2C	0x20910500	U1	-	-	0
CFG-MSGOUT-UBX_NAV2_SBAS_SPI	0x20910504	U1	-	-	0
CFG-MSGOUT-UBX_NAV2_SBAS_UART1	0x20910501	U1	-	-	0
CFG-MSGOUT-UBX_NAV2_SBAS_UART2	0x20910502	U1	-	-	0
CFG-MSGOUT-UBX_NAV2_SBAS_USB	0x20910503	U1	-	-	0
CFG-MSGOUT-UBX_NAV2_SIG_I2C	0x20910505	U1	-	-	0
CFG-MSGOUT-UBX_NAV2_SIG_SPI	0x20910509	U1	-	-	0
CFG-MSGOUT-UBX_NAV2_SIG_UART1	0x20910506	U1	-	-	0
CFG-MSGOUT-UBX_NAV2_SIG_UART2	0x20910507	U1	-	-	0
CFG-MSGOUT-UBX_NAV2_SIG_USB	0x20910508	U1	-	-	0
CFG-MSGOUT-UBX_NAV2_SLAS_I2C	0x20910510	U1	-	-	0
CFG-MSGOUT-UBX_NAV2_SLAS_SPI	0x20910514	U1	-	-	0
CFG-MSGOUT-UBX_NAV2_SLAS_UART1	0x20910511	U1	-	-	0
CFG-MSGOUT-UBX_NAV2_SLAS_UART2	0x20910512	U1	-	-	0
CFG-MSGOUT-UBX_NAV2_SLAS_USB	0x20910513	U1	-	-	0
CFG-MSGOUT-UBX_NAV2_STATUS_I2C	0x20910515	U1	-	-	0
CFG-MSGOUT-UBX_NAV2_STATUS_SPI	0x20910519	U1	-	-	0
CFG-MSGOUT-UBX_NAV2_STATUS_UART1	0x20910516	U1	-	-	0
CFG-MSGOUT-UBX_NAV2_STATUS_UART2	0x20910517	U1	-	-	0
CFG-MSGOUT-UBX_NAV2_STATUS_USB	0x20910518	U1	-	-	0
CFG-MSGOUT-UBX_NAV2_SVIN_I2C	0x20910520	U1	-	-	0

Configuration item	Key ID	Type	Scale	Unit	Default value
CFG-MSGOUT-UBX_NAV2_SVIN_SPI	0x20910524	U1	-	-	0
CFG-MSGOUT-UBX_NAV2_SVIN_UART1	0x20910521	U1	-	-	0
CFG-MSGOUT-UBX_NAV2_SVIN_UART2	0x20910522	U1	-	-	0
CFG-MSGOUT-UBX_NAV2_SVIN_USB	0x20910523	U1	-	-	0
CFG-MSGOUT-UBX_NAV2_TIMEBDS_I2C	0x20910525	U1	-	-	0
CFG-MSGOUT-UBX_NAV2_TIMEBDS_SPI	0x20910529	U1	-	-	0
CFG-MSGOUT-UBX_NAV2_TIMEBDS_UART1	0x20910526	U1	-	-	0
CFG-MSGOUT-UBX_NAV2_TIMEBDS_UART2	0x20910527	U1	-	-	0
CFG-MSGOUT-UBX_NAV2_TIMEBDS_USB	0x20910528	U1	-	-	0
CFG-MSGOUT-UBX_NAV2_TIMEGAL_I2C	0x20910530	U1	-	-	0
CFG-MSGOUT-UBX_NAV2_TIMEGAL_SPI	0x20910534	U1	-	-	0
CFG-MSGOUT-UBX_NAV2_TIMEGAL_UART1	0x20910531	U1	-	-	0
CFG-MSGOUT-UBX_NAV2_TIMEGAL_UART2	0x20910532	U1	-	-	0
CFG-MSGOUT-UBX_NAV2_TIMEGAL_USB	0x20910533	U1	-	-	0
CFG-MSGOUT-UBX_NAV2_TIMEGLO_I2C	0x20910535	U1	-	-	0
CFG-MSGOUT-UBX_NAV2_TIMEGLO_SPI	0x20910539	U1	-	-	0
CFG-MSGOUT-UBX_NAV2_TIMEGLO_UART1	0x20910536	U1	-	-	0
CFG-MSGOUT-UBX_NAV2_TIMEGLO_UART2	0x20910537	U1	-	-	0
CFG-MSGOUT-UBX_NAV2_TIMEGLO_USB	0x20910538	U1	-	-	0
CFG-MSGOUT-UBX_NAV2_TIMEGPS_I2C	0x20910540	U1	-	-	0
CFG-MSGOUT-UBX_NAV2_TIMEGPS_SPI	0x20910544	U1	-	-	0
CFG-MSGOUT-UBX_NAV2_TIMEGPS_UART1	0x20910541	U1	-	-	0
CFG-MSGOUT-UBX_NAV2_TIMEGPS_UART2	0x20910542	U1	-	-	0
CFG-MSGOUT-UBX_NAV2_TIMEGPS_USB	0x20910543	U1	-	-	0
CFG-MSGOUT-UBX_NAV2_TIMELS_I2C	0x20910545	U1	-	-	0
CFG-MSGOUT-UBX_NAV2_TIMELS_SPI	0x20910549	U1	-	-	0
CFG-MSGOUT-UBX_NAV2_TIMELS_UART1	0x20910546	U1	-	-	0
CFG-MSGOUT-UBX_NAV2_TIMELS_UART2	0x20910547	U1	-	-	0
CFG-MSGOUT-UBX_NAV2_TIMELS_USB	0x20910548	U1	-	-	0
CFG-MSGOUT-UBX_NAV2_TIMEQZSS_I2C	0x20910575	U1	-	-	0
CFG-MSGOUT-UBX_NAV2_TIMEQZSS_SPI	0x20910579	U1	-	-	0
CFG-MSGOUT-UBX_NAV2_TIMEQZSS_UART1	0x20910576	U1	-	-	0
CFG-MSGOUT-UBX_NAV2_TIMEQZSS_UART2	0x20910577	U1	-	-	0
CFG-MSGOUT-UBX_NAV2_TIMEQZSS_USB	0x20910578	U1	-	-	0
CFG-MSGOUT-UBX_NAV2_TIMEUTC_I2C	0x20910550	U1	-	-	0
CFG-MSGOUT-UBX_NAV2_TIMEUTC_SPI	0x20910554	U1	-	-	0
CFG-MSGOUT-UBX_NAV2_TIMEUTC_UART1	0x20910551	U1	-	-	0
CFG-MSGOUT-UBX_NAV2_TIMEUTC_UART2	0x20910552	U1	-	-	0
CFG-MSGOUT-UBX_NAV2_TIMEUTC_USB	0x20910553	U1	-	-	0
CFG-MSGOUT-UBX_NAV2_VELECEF_I2C	0x20910555	U1	-	-	0
CFG-MSGOUT-UBX_NAV2_VELECEF_SPI	0x20910559	U1	-	-	0

Configuration item	Key ID	Type	Scale	Unit	Default value
CFG-MSGOUT-UBX_NAV2_VELECEF_UART1	0x20910556	U1	-	-	0
CFG-MSGOUT-UBX_NAV2_VELECEF_UART2	0x20910557	U1	-	-	0
CFG-MSGOUT-UBX_NAV2_VELECEF_USB	0x20910558	U1	-	-	0
CFG-MSGOUT-UBX_NAV2_VELNED_I2C	0x20910560	U1	-	-	0
CFG-MSGOUT-UBX_NAV2_VELNED_SPI	0x20910564	U1	-	-	0
CFG-MSGOUT-UBX_NAV2_VELNED_UART1	0x20910561	U1	-	-	0
CFG-MSGOUT-UBX_NAV2_VELNED_UART2	0x20910562	U1	-	-	0
CFG-MSGOUT-UBX_NAV2_VELNED_USB	0x20910563	U1	-	-	0
CFG-MSGOUT-UBX_NAV_CLOCK_I2C	0x20910065	U1	-	-	0
CFG-MSGOUT-UBX_NAV_CLOCK_SPI	0x20910069	U1	-	-	0
CFG-MSGOUT-UBX_NAV_CLOCK_UART1	0x20910066	U1	-	-	0
CFG-MSGOUT-UBX_NAV_CLOCK_UART2	0x20910067	U1	-	-	0
CFG-MSGOUT-UBX_NAV_CLOCK_USB	0x20910068	U1	-	-	0
CFG-MSGOUT-UBX_NAV_COV_I2C	0x20910083	U1	-	-	0
CFG-MSGOUT-UBX_NAV_COV_SPI	0x20910087	U1	-	-	0
CFG-MSGOUT-UBX_NAV_COV_UART1	0x20910084	U1	-	-	0
CFG-MSGOUT-UBX_NAV_COV_UART2	0x20910085	U1	-	-	0
CFG-MSGOUT-UBX_NAV_COV_USB	0x20910086	U1	-	-	0
CFG-MSGOUT-UBX_NAV_DOP_I2C	0x20910038	U1	-	-	0
CFG-MSGOUT-UBX_NAV_DOP_SPI	0x2091003c	U1	-	-	0
CFG-MSGOUT-UBX_NAV_DOP_UART1	0x20910039	U1	-	-	0
CFG-MSGOUT-UBX_NAV_DOP_UART2	0x2091003a	U1	-	-	0
CFG-MSGOUT-UBX_NAV_DOP_USB	0x2091003b	U1	-	-	0
CFG-MSGOUT-UBX_NAV_EOE_I2C	0x2091015f	U1	-	-	0
CFG-MSGOUT-UBX_NAV_EOE_SPI	0x20910163	U1	-	-	0
CFG-MSGOUT-UBX_NAV_EOE_UART1	0x20910160	U1	-	-	0
CFG-MSGOUT-UBX_NAV_EOE_UART2	0x20910161	U1	-	-	0
CFG-MSGOUT-UBX_NAV_EOE_USB	0x20910162	U1	-	-	0
CFG-MSGOUT-UBX_NAV_GEOFENCE_I2C	0x209100a1	U1	-	-	0
CFG-MSGOUT-UBX_NAV_GEOFENCE_SPI	0x209100a5	U1	-	-	0
CFG-MSGOUT-UBX_NAV_GEOFENCE_UART1	0x209100a2	U1	-	-	0
CFG-MSGOUT-UBX_NAV_GEOFENCE_UART2	0x209100a3	U1	-	-	0
CFG-MSGOUT-UBX_NAV_GEOFENCE_USB	0x209100a4	U1	-	-	0
CFG-MSGOUT-UBX_NAV_HPPOSECEF_I2C	0x2091002e	U1	-	-	0
CFG-MSGOUT-UBX_NAV_HPPOSECEF_SPI	0x20910032	U1	-	-	0
CFG-MSGOUT-UBX_NAV_HPPOSECEF_UART1	0x2091002f	U1	-	-	0
CFG-MSGOUT-UBX_NAV_HPPOSECEF_UART2	0x20910030	U1	-	-	0
CFG-MSGOUT-UBX_NAV_HPPOSECEF_USB	0x20910031	U1	-	-	0
CFG-MSGOUT-UBX_NAV_HPPOSLLH_I2C	0x20910033	U1	-	-	0
CFG-MSGOUT-UBX_NAV_HPPOSLLH_SPI	0x20910037	U1	-	-	0
CFG-MSGOUT-UBX_NAV_HPPOSLLH_UART1	0x20910034	U1	-	-	0

Configuration item	Key ID	Type	Scale	Unit	Default value
CFG-MSGOUT-UBX_NAV_HPPOSLLH_UART2	0x20910035	U1	-	-	0
CFG-MSGOUT-UBX_NAV_HPPOSLLH_USB	0x20910036	U1	-	-	0
CFG-MSGOUT-UBX_NAV_ODO_I2C	0x2091007e	U1	-	-	0
CFG-MSGOUT-UBX_NAV_ODO_SPI	0x20910082	U1	-	-	0
CFG-MSGOUT-UBX_NAV_ODO_UART1	0x2091007f	U1	-	-	0
CFG-MSGOUT-UBX_NAV_ODO_UART2	0x20910080	U1	-	-	0
CFG-MSGOUT-UBX_NAV_ODO_USB	0x20910081	U1	-	-	0
CFG-MSGOUT-UBX_NAV_ORB_I2C	0x20910010	U1	-	-	0
CFG-MSGOUT-UBX_NAV_ORB_SPI	0x20910014	U1	-	-	0
CFG-MSGOUT-UBX_NAV_ORB_UART1	0x20910011	U1	-	-	0
CFG-MSGOUT-UBX_NAV_ORB_UART2	0x20910012	U1	-	-	0
CFG-MSGOUT-UBX_NAV_ORB_USB	0x20910013	U1	-	-	0
CFG-MSGOUT-UBX_NAV_PL_I2C	0x20910415	U1	-	-	0
CFG-MSGOUT-UBX_NAV_PL_SPI	0x20910419	U1	-	-	0
CFG-MSGOUT-UBX_NAV_PL_UART1	0x20910416	U1	-	-	0
CFG-MSGOUT-UBX_NAV_PL_UART2	0x20910417	U1	-	-	0
CFG-MSGOUT-UBX_NAV_PL_USB	0x20910418	U1	-	-	0
CFG-MSGOUT-UBX_NAV_POSECEF_I2C	0x20910024	U1	-	-	0
CFG-MSGOUT-UBX_NAV_POSECEF_SPI	0x20910028	U1	-	-	0
CFG-MSGOUT-UBX_NAV_POSECEF_UART1	0x20910025	U1	-	-	0
CFG-MSGOUT-UBX_NAV_POSECEF_UART2	0x20910026	U1	-	-	0
CFG-MSGOUT-UBX_NAV_POSECEF_USB	0x20910027	U1	-	-	0
CFG-MSGOUT-UBX_NAV_POSLLH_I2C	0x20910029	U1	-	-	0
CFG-MSGOUT-UBX_NAV_POSLLH_SPI	0x2091002d	U1	-	-	0
CFG-MSGOUT-UBX_NAV_POSLLH_UART1	0x2091002a	U1	-	-	0
CFG-MSGOUT-UBX_NAV_POSLLH_UART2	0x2091002b	U1	-	-	0
CFG-MSGOUT-UBX_NAV_POSLLH_USB	0x2091002c	U1	-	-	0
CFG-MSGOUT-UBX_NAV_PVT_I2C	0x20910006	U1	-	-	0
CFG-MSGOUT-UBX_NAV_PVT_SPI	0x2091000a	U1	-	-	0
CFG-MSGOUT-UBX_NAV_PVT_UART1	0x20910007	U1	-	-	0
CFG-MSGOUT-UBX_NAV_PVT_UART2	0x20910008	U1	-	-	0
CFG-MSGOUT-UBX_NAV_PVT_USB	0x20910009	U1	-	-	0
CFG-MSGOUT-UBX_NAV_RELPOSNED_I2C	0x2091008d	U1	-	-	0
CFG-MSGOUT-UBX_NAV_RELPOSNED_SPI	0x20910091	U1	-	-	0
CFG-MSGOUT-UBX_NAV_RELPOSNED_UART1	0x2091008e	U1	-	-	0
CFG-MSGOUT-UBX_NAV_RELPOSNED_UART2	0x2091008f	U1	-	-	0
CFG-MSGOUT-UBX_NAV_RELPOSNED_USB	0x20910090	U1	-	-	0
CFG-MSGOUT-UBX_NAV_SAT_I2C	0x20910015	U1	-	-	0
CFG-MSGOUT-UBX_NAV_SAT_SPI	0x20910019	U1	-	-	0
CFG-MSGOUT-UBX_NAV_SAT_UART1	0x20910016	U1	-	-	0
CFG-MSGOUT-UBX_NAV_SAT_UART2	0x20910017	U1	-	-	0

Configuration item	Key ID	Type	Scale	Unit	Default value
CFG-MSGOUT-UBX_NAV_SAT_USB	0x20910018	U1	-	-	0
CFG-MSGOUT-UBX_NAV_SBAS_I2C	0x2091006a	U1	-	-	0
CFG-MSGOUT-UBX_NAV_SBAS_SPI	0x2091006e	U1	-	-	0
CFG-MSGOUT-UBX_NAV_SBAS_UART1	0x2091006b	U1	-	-	0
CFG-MSGOUT-UBX_NAV_SBAS_UART2	0x2091006c	U1	-	-	0
CFG-MSGOUT-UBX_NAV_SBAS_USB	0x2091006d	U1	-	-	0
CFG-MSGOUT-UBX_NAV_SIG_I2C	0x20910345	U1	-	-	0
CFG-MSGOUT-UBX_NAV_SIG_SPI	0x20910349	U1	-	-	0
CFG-MSGOUT-UBX_NAV_SIG_UART1	0x20910346	U1	-	-	0
CFG-MSGOUT-UBX_NAV_SIG_UART2	0x20910347	U1	-	-	0
CFG-MSGOUT-UBX_NAV_SIG_USB	0x20910348	U1	-	-	0
CFG-MSGOUT-UBX_NAV_SLAS_I2C	0x20910336	U1	-	-	0
CFG-MSGOUT-UBX_NAV_SLAS_SPI	0x2091033a	U1	-	-	0
CFG-MSGOUT-UBX_NAV_SLAS_UART1	0x20910337	U1	-	-	0
CFG-MSGOUT-UBX_NAV_SLAS_UART2	0x20910338	U1	-	-	0
CFG-MSGOUT-UBX_NAV_SLAS_USB	0x20910339	U1	-	-	0
CFG-MSGOUT-UBX_NAV_STATUS_I2C	0x2091001a	U1	-	-	0
CFG-MSGOUT-UBX_NAV_STATUS_SPI	0x2091001e	U1	-	-	0
CFG-MSGOUT-UBX_NAV_STATUS_UART1	0x2091001b	U1	-	-	0
CFG-MSGOUT-UBX_NAV_STATUS_UART2	0x2091001c	U1	-	-	0
CFG-MSGOUT-UBX_NAV_STATUS_USB	0x2091001d	U1	-	-	0
CFG-MSGOUT-UBX_NAV_SVIN_I2C	0x20910088	U1	-	-	0
CFG-MSGOUT-UBX_NAV_SVIN_SPI	0x2091008c	U1	-	-	0
CFG-MSGOUT-UBX_NAV_SVIN_UART1	0x20910089	U1	-	-	0
CFG-MSGOUT-UBX_NAV_SVIN_UART2	0x2091008a	U1	-	-	0
CFG-MSGOUT-UBX_NAV_SVIN_USB	0x2091008b	U1	-	-	0
CFG-MSGOUT-UBX_NAV_TIMEBDS_I2C	0x20910051	U1	-	-	0
CFG-MSGOUT-UBX_NAV_TIMEBDS_SPI	0x20910055	U1	-	-	0
CFG-MSGOUT-UBX_NAV_TIMEBDS_UART1	0x20910052	U1	-	-	0
CFG-MSGOUT-UBX_NAV_TIMEBDS_UART2	0x20910053	U1	-	-	0
CFG-MSGOUT-UBX_NAV_TIMEBDS_USB	0x20910054	U1	-	-	0
CFG-MSGOUT-UBX_NAV_TIMEGAL_I2C	0x20910056	U1	-	-	0
CFG-MSGOUT-UBX_NAV_TIMEGAL_SPI	0x2091005a	U1	-	-	0
CFG-MSGOUT-UBX_NAV_TIMEGAL_UART1	0x20910057	U1	-	-	0
CFG-MSGOUT-UBX_NAV_TIMEGAL_UART2	0x20910058	U1	-	-	0
CFG-MSGOUT-UBX_NAV_TIMEGAL_USB	0x20910059	U1	-	-	0
CFG-MSGOUT-UBX_NAV_TIMEGLO_I2C	0x2091004c	U1	-	-	0
CFG-MSGOUT-UBX_NAV_TIMEGLO_SPI	0x20910050	U1	-	-	0
CFG-MSGOUT-UBX_NAV_TIMEGLO_UART1	0x2091004d	U1	-	-	0
CFG-MSGOUT-UBX_NAV_TIMEGLO_UART2	0x2091004e	U1	-	-	0
CFG-MSGOUT-UBX_NAV_TIMEGLO_USB	0x2091004f	U1	-	-	0

Configuration item	Key ID	Type	Scale	Unit	Default value
CFG-MSGOUT-UBX_NAV_TIMEGPS_I2C	0x20910047	U1	-	-	0
CFG-MSGOUT-UBX_NAV_TIMEGPS_SPI	0x2091004b	U1	-	-	0
CFG-MSGOUT-UBX_NAV_TIMEGPS_UART1	0x20910048	U1	-	-	0
CFG-MSGOUT-UBX_NAV_TIMEGPS_UART2	0x20910049	U1	-	-	0
CFG-MSGOUT-UBX_NAV_TIMEGPS_USB	0x2091004a	U1	-	-	0
CFG-MSGOUT-UBX_NAV_TIMELS_I2C	0x20910060	U1	-	-	0
CFG-MSGOUT-UBX_NAV_TIMELS_SPI	0x20910064	U1	-	-	0
CFG-MSGOUT-UBX_NAV_TIMELS_UART1	0x20910061	U1	-	-	0
CFG-MSGOUT-UBX_NAV_TIMELS_UART2	0x20910062	U1	-	-	0
CFG-MSGOUT-UBX_NAV_TIMELS_USB	0x20910063	U1	-	-	0
CFG-MSGOUT-UBX_NAV_TIMEQZSS_I2C	0x20910386	U1	-	-	0
CFG-MSGOUT-UBX_NAV_TIMEQZSS_SPI	0x2091038a	U1	-	-	0
CFG-MSGOUT-UBX_NAV_TIMEQZSS_UART1	0x20910387	U1	-	-	0
CFG-MSGOUT-UBX_NAV_TIMEQZSS_UART2	0x20910388	U1	-	-	0
CFG-MSGOUT-UBX_NAV_TIMEQZSS_USB	0x20910389	U1	-	-	0
CFG-MSGOUT-UBX_NAV_TIMEUTC_I2C	0x2091005b	U1	-	-	0
CFG-MSGOUT-UBX_NAV_TIMEUTC_SPI	0x2091005f	U1	-	-	0
CFG-MSGOUT-UBX_NAV_TIMEUTC_UART1	0x2091005c	U1	-	-	0
CFG-MSGOUT-UBX_NAV_TIMEUTC_UART2	0x2091005d	U1	-	-	0
CFG-MSGOUT-UBX_NAV_TIMEUTC_USB	0x2091005e	U1	-	-	0
CFG-MSGOUT-UBX_NAV_VELECEF_I2C	0x2091003d	U1	-	-	0
CFG-MSGOUT-UBX_NAV_VELECEF_SPI	0x20910041	U1	-	-	0
CFG-MSGOUT-UBX_NAV_VELECEF_UART1	0x2091003e	U1	-	-	0
CFG-MSGOUT-UBX_NAV_VELECEF_UART2	0x2091003f	U1	-	-	0
CFG-MSGOUT-UBX_NAV_VELECEF_USB	0x20910040	U1	-	-	0
CFG-MSGOUT-UBX_NAV_VELNED_I2C	0x20910042	U1	-	-	0
CFG-MSGOUT-UBX_NAV_VELNED_SPI	0x20910046	U1	-	-	0
CFG-MSGOUT-UBX_NAV_VELNED_UART1	0x20910043	U1	-	-	0
CFG-MSGOUT-UBX_NAV_VELNED_UART2	0x20910044	U1	-	-	0
CFG-MSGOUT-UBX_NAV_VELNED_USB	0x20910045	U1	-	-	0
CFG-MSGOUT-UBX_RXM_COR_I2C	0x209106b6	U1	-	-	0
CFG-MSGOUT-UBX_RXM_COR_SPI	0x209106ba	U1	-	-	0
CFG-MSGOUT-UBX_RXM_COR_UART1	0x209106b7	U1	-	-	0
CFG-MSGOUT-UBX_RXM_COR_UART2	0x209106b8	U1	-	-	0
CFG-MSGOUT-UBX_RXM_COR_USB	0x209106b9	U1	-	-	0
CFG-MSGOUT-UBX_RXM_MEASX_I2C	0x20910204	U1	-	-	0
CFG-MSGOUT-UBX_RXM_MEASX_SPI	0x20910208	U1	-	-	0
CFG-MSGOUT-UBX_RXM_MEASX_UART1	0x20910205	U1	-	-	0
CFG-MSGOUT-UBX_RXM_MEASX_UART2	0x20910206	U1	-	-	0
CFG-MSGOUT-UBX_RXM_MEASX_USB	0x20910207	U1	-	-	0
CFG-MSGOUT-UBX_RXM_RAWX_I2C	0x209102a4	U1	-	-	0

Configuration item	Key ID	Type	Scale	Unit	Default value
CFG-MSGOUT-UBX_RXM_RAWX_SPI	0x209102a8	U1	-	-	0
CFG-MSGOUT-UBX_RXM_RAWX_UART1	0x209102a5	U1	-	-	0
CFG-MSGOUT-UBX_RXM_RAWX_UART2	0x209102a6	U1	-	-	0
CFG-MSGOUT-UBX_RXM_RAWX_USB	0x209102a7	U1	-	-	0
CFG-MSGOUT-UBX_RXM_RLM_I2C	0x2091025e	U1	-	-	0
CFG-MSGOUT-UBX_RXM_RLM_SPI	0x20910262	U1	-	-	0
CFG-MSGOUT-UBX_RXM_RLM_UART1	0x2091025f	U1	-	-	0
CFG-MSGOUT-UBX_RXM_RLM_UART2	0x20910260	U1	-	-	0
CFG-MSGOUT-UBX_RXM_RLM_USB	0x20910261	U1	-	-	0
CFG-MSGOUT-UBX_RXM_RTCM_I2C	0x20910268	U1	-	-	0
CFG-MSGOUT-UBX_RXM_RTCM_SPI	0x2091026c	U1	-	-	0
CFG-MSGOUT-UBX_RXM_RTCM_UART1	0x20910269	U1	-	-	0
CFG-MSGOUT-UBX_RXM_RTCM_UART2	0x2091026a	U1	-	-	0
CFG-MSGOUT-UBX_RXM_RTCM_USB	0x2091026b	U1	-	-	0
CFG-MSGOUT-UBX_RXM_SFRBX_I2C	0x20910231	U1	-	-	0
CFG-MSGOUT-UBX_RXM_SFRBX_SPI	0x20910235	U1	-	-	0
CFG-MSGOUT-UBX_RXM_SFRBX_UART1	0x20910232	U1	-	-	0
CFG-MSGOUT-UBX_RXM_SFRBX_UART2	0x20910233	U1	-	-	0
CFG-MSGOUT-UBX_RXM_SFRBX_USB	0x20910234	U1	-	-	0
CFG-MSGOUT-UBX_RXM_SPARTN_I2C	0x20910605	U1	-	-	0
CFG-MSGOUT-UBX_RXM_SPARTN_SPI	0x20910609	U1	-	-	0
CFG-MSGOUT-UBX_RXM_SPARTN_UART1	0x20910606	U1	-	-	0
CFG-MSGOUT-UBX_RXM_SPARTN_UART2	0x20910607	U1	-	-	0
CFG-MSGOUT-UBX_RXM_SPARTN_USB	0x20910608	U1	-	-	0
CFG-MSGOUT-UBX_TIM_TM2_I2C	0x20910178	U1	-	-	0
CFG-MSGOUT-UBX_TIM_TM2_SPI	0x2091017c	U1	-	-	0
CFG-MSGOUT-UBX_TIM_TM2_UART1	0x20910179	U1	-	-	0
CFG-MSGOUT-UBX_TIM_TM2_UART2	0x2091017a	U1	-	-	0
CFG-MSGOUT-UBX_TIM_TM2_USB	0x2091017b	U1	-	-	0
CFG-MSGOUT-UBX_TIM_TP_I2C	0x2091017d	U1	-	-	0
CFG-MSGOUT-UBX_TIM_TP_SPI	0x20910181	U1	-	-	0
CFG-MSGOUT-UBX_TIM_TP_UART1	0x2091017e	U1	-	-	0
CFG-MSGOUT-UBX_TIM_TP_UART2	0x2091017f	U1	-	-	0
CFG-MSGOUT-UBX_TIM_TP_USB	0x20910180	U1	-	-	0
CFG-MSGOUT-UBX_TIM_VRFY_I2C	0x20910092	U1	-	-	0
CFG-MSGOUT-UBX_TIM_VRFY_SPI	0x20910096	U1	-	-	0
CFG-MSGOUT-UBX_TIM_VRFY_UART1	0x20910093	U1	-	-	0
CFG-MSGOUT-UBX_TIM_VRFY_UART2	0x20910094	U1	-	-	0
CFG-MSGOUT-UBX_TIM_VRFY_USB	0x20910095	U1	-	-	0

Table 83: CFG-MSGOUT configuration defaults

Configuration item	Key ID	Type	Scale	Unit	Default value
CFG-NAV2-OUT_ENABLED	0x10170001	L	-	-	0 (false)
CFG-NAV2-SBAS_USE_INTEGRITY	0x10170002	L	-	-	0 (false)

Table 84: CFG-NAV2 configuration defaults

Configuration item	Key ID	Type	Scale	Unit	Default value
CFG-NAVHPG-DGNSSMODE	0x20140011	E1	-	-	3 (RTK_FIXED)

Table 85: CFG-NAVHPG configuration defaults

Configuration item	Key ID	Type	Scale	Unit	Default value
CFG-NAVSPG-FIXMODE	0x20110011	E1	-	-	3 (AUTO)
CFG-NAVSPG-INIFIX3D	0x10110013	L	-	-	0 (false)
CFG-NAVSPG-WKNROLLOVER	0x30110017	U2	-	-	2188
CFG-NAVSPG-UTCSTANDARD	0x2011001c	E1	-	-	0 (AUTO)
CFG-NAVSPG-DYNMODEL	0x20110021	E1	-	-	0 (PORT)
CFG-NAVSPG-ACKAIDING	0x10110025	L	-	-	0 (false)
CFG-NAVSPG-USE_USRDAT	0x10110061	L	-	-	0 (false)
CFG-NAVSPG-USRDAT_MAJA	0x50110062	R8	-	m	6378137
CFG-NAVSPG-USRDAT_FLAT	0x50110063	R8	-	-	298.257223563000002502
CFG-NAVSPG-USRDAT_DX	0x40110064	R4	-	m	0
CFG-NAVSPG-USRDAT_DY	0x40110065	R4	-	m	0
CFG-NAVSPG-USRDAT_DZ	0x40110066	R4	-	m	0
CFG-NAVSPG-USRDAT_ROTX	0x40110067	R4	-	arcsec	0
CFG-NAVSPG-USRDAT_ROTY	0x40110068	R4	-	arcsec	0
CFG-NAVSPG-USRDAT_ROTZ	0x40110069	R4	-	arcsec	0
CFG-NAVSPG-USRDAT_SCALE	0x4011006a	R4	-	ppm	0
CFG-NAVSPG-INFIL_MINSVS	0x201100a1	U1	-	-	3
CFG-NAVSPG-INFIL_MAXSVS	0x201100a2	U1	-	-	32
CFG-NAVSPG-INFIL_MINCNO	0x201100a3	U1	-	dBHz	6
CFG-NAVSPG-INFIL_MINELEV	0x201100a4	I1	-	deg	10
CFG-NAVSPG-INFIL_NCNOTHRS	0x201100aa	U1	-	-	0
CFG-NAVSPG-INFIL_CNOTHRS	0x201100ab	U1	-	-	0
CFG-NAVSPG-OUTFIL_PDOP	0x301100b1	U2	0.1	-	250
CFG-NAVSPG-OUTFIL_TDOP	0x301100b2	U2	0.1	-	250
CFG-NAVSPG-OUTFIL_PACC	0x301100b3	U2	-	m	100
CFG-NAVSPG-OUTFIL_TACC	0x301100b4	U2	-	m	350
CFG-NAVSPG-OUTFIL_FACC	0x301100b5	U2	0.01	m/s	150
CFG-NAVSPG-CONSTR_ALT	0x401100c1	I4	0.01	m	0
CFG-NAVSPG-CONSTR_ALTVAR	0x401100c2	U4	0.0001	m^2	10000
CFG-NAVSPG-CONSTR_DGNSSTO	0x201100c4	U1	-	s	60
CFG-NAVSPG-PL_ENA	0x101100d7	L	-	-	1 (true)

Table 86: CFG-NAVSPG configuration defaults

Configuration item	Key ID	Type	Scale	Unit	Default value
CFG-NMEA-PROTVER	0x20930001	E1	-	-	42 (V411)
CFG-NMEA-MAXSVS	0x20930002	E1	-	-	0 (UNLIM)
CFG-NMEA-COMPAT	0x10930003	L	-	-	0 (false)
CFG-NMEA-CONSIDER	0x10930004	L	-	-	1 (true)
CFG-NMEA-LIMIT82	0x10930005	L	-	-	0 (false)
CFG-NMEA-HIGHPREC	0x10930006	L	-	-	0 (false)
CFG-NMEA-SVNUMBERING	0x20930007	E1	-	-	0 (STRICT)
CFG-NMEA-FILT_GPS	0x10930011	L	-	-	0 (false)
CFG-NMEA-FILT_SBAS	0x10930012	L	-	-	0 (false)
CFG-NMEA-FILT_GAL	0x10930013	L	-	-	0 (false)
CFG-NMEA-FILT_QZSS	0x10930015	L	-	-	0 (false)
CFG-NMEA-FILT_GLO	0x10930016	L	-	-	0 (false)
CFG-NMEA-FILT_BDS	0x10930017	L	-	-	0 (false)
CFG-NMEA-OUT_INVFIX	0x10930021	L	-	-	0 (false)
CFG-NMEA-OUT_MSKFIX	0x10930022	L	-	-	0 (false)
CFG-NMEA-OUT_INVTIME	0x10930023	L	-	-	0 (false)
CFG-NMEA-OUT_INVDATE	0x10930024	L	-	-	0 (false)
CFG-NMEA-OUT_ONLYGPS	0x10930025	L	-	-	0 (false)
CFG-NMEA-OUT_FROZENCOG	0x10930026	L	-	-	0 (false)
CFG-NMEA-MAINTALKERID	0x20930031	E1	-	-	0 (AUTO)
CFG-NMEA-GSVTALKERID	0x20930032	E1	-	-	0 (GNSS)
CFG-NMEA-BDSTALKERID	0x30930033	U2	-	-	0

Table 87: CFG-NMEA configuration defaults

Configuration item	Key ID	Type	Scale	Unit	Default value
CFG-ODO-USE_ODO	0x10220001	L	-	-	0 (false)
CFG-ODO-USE_COG	0x10220002	L	-	-	0 (false)
CFG-ODO-OUTLPVEL	0x10220003	L	-	-	0 (false)
CFG-ODO-OUTLPCOG	0x10220004	L	-	-	0 (false)
CFG-ODO-PROFILE	0x20220005	E1	-	-	0 (RUN)
CFG-ODO-COGMAXSPEED	0x20220021	U1	-	m/s	10
CFG-ODO-COGMAXPOSACC	0x20220022	U1	-	-	50
CFG-ODO-VELLPGAIN	0x20220031	U1	-	-	153
CFG-ODO-COGLPGAIN	0x20220032	U1	-	-	76

Table 88: CFG-ODO configuration defaults

Configuration item	Key ID	Type	Scale	Unit	Default value
CFG-QZSS-USE_SLAS_DGNSS	0x10370005	L	-	-	1 (true)
CFG-QZSS-USE_SLAS_TESTMODE	0x10370006	L	-	-	0 (false)
CFG-QZSS-USE_SLAS_RAIM_UNCORR	0x10370007	L	-	-	0 (false)
CFG-QZSS-SLAS_MAX_BASELINE	0x30370008	U2	-	km	200

Table 89: CFG-QZSS configuration defaults

Configuration item	Key ID	Type	Scale	Unit	Default value
CFG-RATE-MEAS	0x30210001	U2	0.001	s	1000
CFG-RATE-NAV	0x30210002	U2	-	-	1
CFG-RATE-TIMEREF	0x20210003	E1	-	-	1 (GPS)

Table 90: CFG-RATE configuration defaults

Configuration item	Key ID	Type	Scale	Unit	Default value
CFG-RINV-DUMP	0x10c70001	L	-	-	0 (false)
CFG-RINV-BINARY	0x10c70002	L	-	-	0 (false)
CFG-RINV-DATA_SIZE	0x20c70003	U1	-	-	22
CFG-RINV-CHUNK0	0x50c70004	X8	-	-	0x203a656369746f4e ("Notice: ")
CFG-RINV-CHUNK1	0x50c70005	X8	-	-	0x2061746164206f6e ("no data ")
CFG-RINV-CHUNK2	0x50c70006	X8	-	-	0x000021646576617 3 ("saved!\0\0")
CFG-RINV-CHUNK3	0x50c70007	X8	-	-	0x0000000000000000

Table 91: CFG-RINV configuration defaults

Configuration item	Key ID	Type	Scale	Unit	Default value
CFG-RTCM-DF003_OUT	0x30090001	U2	-	-	0
CFG-RTCM-DF003_IN	0x30090008	U2	-	-	0
CFG-RTCM-DF003_IN_FILTER	0x20090009	E1	-	-	0 (DISABLED)

Table 92: CFG-RTCM configuration defaults

Configuration item	Key ID	Type	Scale	Unit	Default value
CFG-SBAS-USE_TESTMODE	0x10360002	L	-	-	0 (false)
CFG-SBAS-USE_RANGING	0x10360003	L	-	-	1 (true)
CFG-SBAS-USE_DIFFCORR	0x10360004	L	-	-	1 (true)
CFG-SBAS-USE_INTEGRITY	0x10360005	L	-	-	0 (false)
CFG-SBAS-PRNSCANMASK	0x50360006	X8	-	-	0x0000000000072b88 (ALL | PRN123 | PRN127 | PRN128 | PRN129 | PRN131 | PRN133 | PRN136 | PRN137 | PRN138)

Table 93: CFG-SBAS configuration defaults

Configuration item	Key ID	Type	Scale	Unit	Default value
CFG-SEC-CFG_LOCK	0x10f60009	L	-	-	0 (false)
CFG-SEC-CFG_LOCK_UNLOCKGRP1	0x30f6000a	U2	-	-	0
CFG-SEC-CFG_LOCK_UNLOCKGRP2	0x30f6000b	U2	-	-	0

Table 94: CFG-SEC configuration defaults

Configuration item	Key ID	Type	Scale	Unit	Default value
CFG-SIGNAL-GPS_ENA	0x1031001f	L	-	-	1 (true)
CFG-SIGNAL-GPS_L1CA_ENA	0x10310001	L	-	-	1 (true)
CFG-SIGNAL-GPS_L2C_ENA	0x10310003	L	-	-	1 (true)
CFG-SIGNAL-SBAS_ENA	0x10310020	L	-	-	1 (true)
CFG-SIGNAL-SBAS_L1CA_ENA	0x10310005	L	-	-	1 (true)

Configuration item	Key ID	Type	Scale	Unit	Default value
CFG-SIGNAL-GAL_ENA	0x10310021	L	-	-	1 (true)
CFG-SIGNAL-GAL_E1_ENA	0x10310007	L	-	-	1 (true)
CFG-SIGNAL-GAL_E5B_ENA	0x1031000a	L	-	-	1 (true)
CFG-SIGNAL-BDS_ENA	0x10310022	L	-	-	1 (true)
CFG-SIGNAL-BDS_B1_ENA	0x1031000d	L	-	-	1 (true)
CFG-SIGNAL-BDS_B2_ENA	0x1031000e	L	-	-	1 (true)
CFG-SIGNAL-QZSS_ENA	0x10310024	L	-	-	1 (true)
CFG-SIGNAL-QZSS_L1CA_ENA	0x10310012	L	-	-	1 (true)
CFG-SIGNAL-QZSS_L1S_ENA	0x10310014	L	-	-	0 (false)
CFG-SIGNAL-QZSS_L2C_ENA	0x10310015	L	-	-	1 (true)
CFG-SIGNAL-GLO_ENA	0x10310025	L	-	-	1 (true)
CFG-SIGNAL-GLO_L1_ENA	0x10310018	L	-	-	1 (true)
CFG-SIGNAL-GLO_L2_ENA	0x1031001a	L	-	-	1 (true)

Table 95: CFG-SIGNAL configuration defaults

Configuration item	Key ID	Type	Scale	Unit	Default value
CFG-SPARTN-USE_SOURCE	0x20a70001	E1	-	-	0 (IP)

Table 96: CFG-SPARTN configuration defaults

Configuration item	Key ID	Type	Scale	Unit	Default value
CFG-SPI-MAXFF	0x20640001	U1	-	-	50
CFG-SPI-CPOLARITY	0x10640002	L	-	-	0 (false)
CFG-SPI-CPHASE	0x10640003	L	-	-	0 (false)
CFG-SPI-EXTENDEDTIMEOUT	0x10640005	L	-	-	0 (false)
CFG-SPI-ENABLED	0x10640006	L	-	-	0 (false)

Table 97: CFG-SPI configuration defaults

Configuration item	Key ID	Type	Scale	Unit	Default value
CFG-SPIINPROT-UBX	0x10790001	L	-	-	1 (true)
CFG-SPIINPROT-NMEA	0x10790002	L	-	-	1 (true)
CFG-SPIINPROT-RTCM3X	0x10790004	L	-	-	1 (true)
CFG-SPIINPROT-SPARTN	0x10790005	L	-	-	1 (true)

Table 98: CFG-SPIINPROT configuration defaults

Configuration item	Key ID	Type	Scale	Unit	Default value
CFG-SPIOUTPROT-UBX	0x107a0001	L	-	-	1 (true)
CFG-SPIOUTPROT-NMEA	0x107a0002	L	-	-	1 (true)
CFG-SPIOUTPROT-RTCM3X	0x107a0004	L	-	-	1 (true)

Table 99: CFG-SPIOUTPROT configuration defaults

Configuration item	Key ID	Type	Scale	Unit	Default value
CFG-TMODE-MODE	0x20030001	E1	-	-	0 (DISABLED)
CFG-TMODE-POS_TYPE	0x20030002	E1	-	-	0 (ECEF)
CFG-TMODE-ECEF_X	0x40030003	I4	-	cm	0

Configuration item	Key ID	Type	Scale	Unit	Default value
CFG-TMODE-ECEF_Y	0x40030004	I4	-	cm	0
CFG-TMODE-ECEF_Z	0x40030005	I4	-	cm	0
CFG-TMODE-ECEF_X_HP	0x20030006	I1	0.1	mm	0
CFG-TMODE-ECEF_Y_HP	0x20030007	I1	0.1	mm	0
CFG-TMODE-ECEF_Z_HP	0x20030008	I1	0.1	mm	0
CFG-TMODE-LAT	0x40030009	I4	1e-7	deg	0
CFG-TMODE-LON	0x4003000a	I4	1e-7	deg	0
CFG-TMODE-HEIGHT	0x4003000b	I4	-	cm	0
CFG-TMODE-LAT_HP	0x2003000c	I1	1e-9	deg	0
CFG-TMODE-LON_HP	0x2003000d	I1	1e-9	deg	0
CFG-TMODE-HEIGHT_HP	0x2003000e	I1	0.1	mm	0
CFG-TMODE-FIXED_POS_ACC	0x4003000f	U4	0.1	mm	0
CFG-TMODE-SVIN_MIN_DUR	0x40030010	U4	-	s	0
CFG-TMODE-SVIN_ACC_LIMIT	0x40030011	U4	0.1	mm	0

Table 100: CFG-TMODE configuration defaults

Configuration item	Key ID	Type	Scale	Unit	Default value
CFG-TP-PULSE_DEF	0x20050023	E1	-	-	0 (PERIOD)
CFG-TP-PULSE_LENGTH_DEF	0x20050030	E1	-	-	1 (LENGTH)
CFG-TP-ANT_CABLEDELAY	0x30050001	I2	1e-9	s	50
CFG-TP-PERIOD_TP1	0x40050002	U4	1e-6	s	1000000
CFG-TP-PERIOD_LOCK_TP1	0x40050003	U4	1e-6	s	1000000
CFG-TP-FREQ_TP1	0x40050024	U4	-	Hz	1
CFG-TP-FREQ_LOCK_TP1	0x40050025	U4	-	Hz	1
CFG-TP-LEN_TP1	0x40050004	U4	1e-6	s	0
CFG-TP-LEN_LOCK_TP1	0x40050005	U4	1e-6	s	100000
CFG-TP-DUTY_TP1	0x5005002a	R8	-	%	0
CFG-TP-DUTY_LOCK_TP1	0x5005002b	R8	-	%	10
CFG-TP-USER_DELAY_TP1	0x40050006	I4	1e-9	s	0
CFG-TP-TP1_ENA	0x10050007	L	-	-	1 (true)
CFG-TP-SYNC_GNSS_TP1	0x10050008	L	-	-	1 (true)
CFG-TP-USE_LOCKED_TP1	0x10050009	L	-	-	1 (true)
CFG-TP-ALIGN_TO_TOW_TP1	0x1005000a	L	-	-	1 (true)
CFG-TP-POL_TP1	0x1005000b	L	-	-	1 (true)
CFG-TP-TIMEGRID_TP1	0x2005000c	E1	-	-	0 (UTC)
CFG-TP-DRSTR_TP1	0x20050035	E1	-	-	1 (DRIVE_STRENGTH_4MA)

Table 101: CFG-TP configuration defaults

Configuration item	Key ID	Type	Scale	Unit	Default value
CFG-TXREADY-ENABLED	0x10a20001	L	-	-	0 (false)
CFG-TXREADY-POLARITY	0x10a20002	L	-	-	0 (false)
CFG-TXREADY-PIN	0x20a20003	U1	-	-	0

Configuration item	Key ID	Type	Scale	Unit	Default value
CFG-TXREADY-THRESHOLD	0x30a20004	U2	-	-	0
CFG-TXREADY-INTERFACE	0x20a20005	E1	-	-	0 (I2C)

Table 102: CFG-TXREADY configuration defaults

Configuration item	Key ID	Type	Scale	Unit	Default value
CFG-UART1-BAUDRATE	0x40520001	U4	-	-	38400
CFG-UART1-STOPBITS	0x20520002	E1	-	-	1 (ONE)
CFG-UART1-DATABITS	0x20520003	E1	-	-	0 (EIGHT)
CFG-UART1-PARITY	0x20520004	E1	-	-	0 (NONE)
CFG-UART1-ENABLED	0x10520005	L	-	-	1 (true)

Table 103: CFG-UART1 configuration defaults

Configuration item	Key ID	Type	Scale	Unit	Default value
CFG-UART1INPROT-UBX	0x10730001	L	-	-	1 (true)
CFG-UART1INPROT-NMEA	0x10730002	L	-	-	1 (true)
CFG-UART1INPROT-RTCM3X	0x10730004	L	-	-	1 (true)
CFG-UART1INPROT-SPARTN	0x10730005	L	-	-	1 (true)

Table 104: CFG-UART1INPROT configuration defaults

Configuration item	Key ID	Type	Scale	Unit	Default value
CFG-UART1OUTPROT-UBX	0x10740001	L	-	-	1 (true)
CFG-UART1OUTPROT-NMEA	0x10740002	L	-	-	1 (true)
CFG-UART1OUTPROT-RTCM3X	0x10740004	L	-	-	1 (true)

Table 105: CFG-UART1OUTPROT configuration defaults

Configuration item	Key ID	Type	Scale	Unit	Default value
CFG-UART2-BAUDRATE	0x40530001	U4	-	-	38400
CFG-UART2-STOPBITS	0x20530002	E1	-	-	1 (ONE)
CFG-UART2-DATABITS	0x20530003	E1	-	-	0 (EIGHT)
CFG-UART2-PARITY	0x20530004	E1	-	-	0 (NONE)
CFG-UART2-ENABLED	0x10530005	L	-	-	1 (true)

Table 106: CFG-UART2 configuration defaults

Configuration item	Key ID	Type	Scale	Unit	Default value
CFG-UART2INPROT-UBX	0x10750001	L	-	-	0 (false)
CFG-UART2INPROT-NMEA	0x10750002	L	-	-	0 (false)
CFG-UART2INPROT-RTCM3X	0x10750004	L	-	-	1 (true)
CFG-UART2INPROT-SPARTN	0x10750005	L	-	-	1 (true)

Table 107: CFG-UART2INPROT configuration defaults

Configuration item	Key ID	Type	Scale	Unit	Default value
CFG-UART2OUTPROT-UBX	0x10760001	L	-	-	0 (false)
CFG-UART2OUTPROT-NMEA	0x10760002	L	-	-	0 (false)
CFG-UART2OUTPROT-RTCM3X	0x10760004	L	-	-	1 (true)

Table 108: CFG-UART2OUTPROT configuration defaults

Configuration item	Key ID	Type	Scale	Unit	Default value
CFG-USB-ENABLED	0x10650001	L	-	-	1 (true)
CFG-USB-SELFPOW	0x10650002	L	-	-	1 (true)
CFG-USB-VENDOR_ID	0x3065000a	U2	-	-	5446
CFG-USB-PRODUCT_ID	0x3065000b	U2	-	-	425
CFG-USB-POWER	0x3065000c	U2	-	mA	0
CFG-USB-VENDOR_STR0	0x5065000d	X8	-	-	0x4120786f6c622d75 ("u-blox A")
CFG-USB-VENDOR_STR1	0x5065000e	X8	-	-	0x2e777777202d2047 ("G - www.")
CFG-USB-VENDOR_STR2	0x5065000f	X8	-	-	0x632e786f6c622d75 ("u-blox.c")
CFG-USB-VENDOR_STR3	0x50650010	X8	-	-	0x0000000000006d6f ("om\0\0\0\0\0\0")
CFG-USB-PRODUCT_STR0	0x50650011	X8	-	-	0x4720786f6c622d75 ("u-blox G")
CFG-USB-PRODUCT_STR1	0x50650012	X8	-	-	0x656365722053534e ("NSS rece")
CFG-USB-PRODUCT_STR2	0x50650013	X8	-	-	0x0000000072657669 ("iver\0\0\0\0")
CFG-USB-PRODUCT_STR3	0x50650014	X8	-	-	0x0000000000000000
CFG-USB-SERIAL_NO_STR0	0x50650015	X8	-	-	0x0000000000000000
CFG-USB-SERIAL_NO_STR1	0x50650016	X8	-	-	0x0000000000000000
CFG-USB-SERIAL_NO_STR2	0x50650017	X8	-	-	0x0000000000000000
CFG-USB-SERIAL_NO_STR3	0x50650018	X8	-	-	0x0000000000000000

Table 109: CFG-USB configuration defaults

Configuration item	Key ID	Type	Scale	Unit	Default value
CFG-USBINPROT-UBX	0x10770001	L	-	-	1 (true)
CFG-USBINPROT-NMEA	0x10770002	L	-	-	1 (true)
CFG-USBINPROT-RTCM3X	0x10770004	L	-	-	1 (true)
CFG-USBINPROT-SPARTN	0x10770005	L	-	-	1 (true)

Table 110: CFG-USBINPROT configuration defaults

Configuration item	Key ID	Type	Scale	Unit	Default value
CFG-USBOUTPROT-UBX	0x10780001	L	-	-	1 (true)
CFG-USBOUTPROT-NMEA	0x10780002	L	-	-	1 (true)
CFG-USBOUTPROT-RTCM3X	0x10780004	L	-	-	1 (true)

Table 111: CFG-USBOUTPROT configuration defaults

Related documents

[1] ZED-F9P-04B Data sheet, UBX-21044850

[2] ZED-F9P integration manual, UBX-18010802

[3] RTCM Standard 10403.3 Recommended Standards for Differential GNSS (Global Navigation Satellite Systems) Service, Version 3

[4] Radio Resource LCS Protocol (RRLP), (3GPP TS 44.031 version 11.0.0 Release 11)

[5] NMEA 0183 Standard for Interfacing Marine Electronic Devices, Version 4.11, November 2018

[6] Secure Position Augmentation for Real-Time Navigation (SPARTN) Interface Control Document, Version 2.0.1, September 2021

☞ For regular updates to u-blox documentation and to receive product change notifications please register on our homepage (https://www.u-blox.com).

Revision history

Revision	Date	Name	Status / Comments
R01	16-Dec-2021	gste	HPG 1.30 release

Contact

For complete contact information visit us at www.u-blox.com.

u-blox Offices

North, Central and South America

u-blox America, Inc.
Phone: +1 703 483 3180
E-mail: info_us@u-blox.com

Regional Office West Coast
Phone: +1 408 573 3640
E-mail: info_us@u-blox.com

Technical Support
Phone: +1 703 483 3185
E-mail: support_us@u-blox.com

Headquarters

Europe, Middle East, Africa

u-blox AG
Phone: +41 44 722 74 44
E-mail: info@u-blox.com
Support: support@u-blox.com

Asia, Australia, Pacific

u-blox Singapore Pte. Ltd.
Phone: +65 6734 3811
E-mail: info_ap@u-blox.com
Support: support_ap@u-blox.com

Regional Office Australia
Phone: +61 3 9566 7255
E-mail: info_anz@u-blox.com
Support: support_ap@u-blox.com

Regional Office China (Beijing)
Phone: +86 10 68 133 545
E-mail: info_cn@u-blox.com
Support: support_cn@u-blox.com

Regional Office China (Chongqing)
Phone: +86 23 6815 1588
E-mail: info_cn@u-blox.com
Support: support_cn@u-blox.com

Regional Office China (Shanghai)
Phone: +86 21 6090 4832
E-mail: info_cn@u-blox.com
Support: support_cn@u-blox.com

Regional Office China (Shenzhen)
Phone: +86 755 8627 1083
E-mail: info_cn@u-blox.com
Support: support_cn@u-blox.com

Regional Office India
Phone: +91 80 4050 9200
E-mail: info_in@u-blox.com
Support: support_in@u-blox.com

Regional Office Japan (Osaka)
Phone: +81 6 6941 3660
E-mail: info_jp@u-blox.com
Support: support_jp@u-blox.com

Regional Office Japan (Tokyo)
Phone: +81 3 5775 3850
E-mail: info_jp@u-blox.com
Support: support_jp@u-blox.com

Regional Office Korea
Phone: +82 2 542 0861
E-mail: info_kr@u-blox.com
Support: support_kr@u-blox.com

Regional Office Taiwan
Phone: +886 2 2657 1090
E-mail: info_tw@u-blox.com
Support: support_tw@u-blox.com