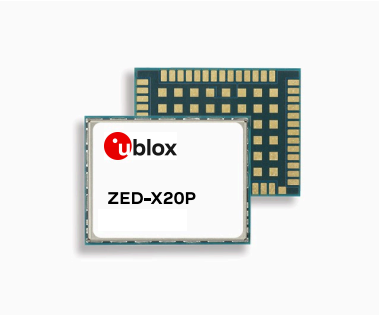
u-blox X20 HPG 2.00

u-blox X20 high precision GNSS receiver

Interface description



Abstract
This document describes the interface (version 50.00) of the ZED-X20P, an all-band GNSS module with integrated RTK offering centimeter level accuracy.

www.u-blox.com

UBXDOC-963802114-12905 - R01
C2-Restricted

Document information

Title	**u-blox X20 HPG 2.00**	
Subtitle	u-blox X20 high precision GNSS receiver	
Document type	Interface description	
Document number	UBXDOC-963802114-12905	
Revision and date	R01	09-Sep-2024
Disclosure restriction	C2-Restricted	

Contents

1 General information

1.1 Document overview

This document describes the interface of the u-blox X20 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 applicable 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.

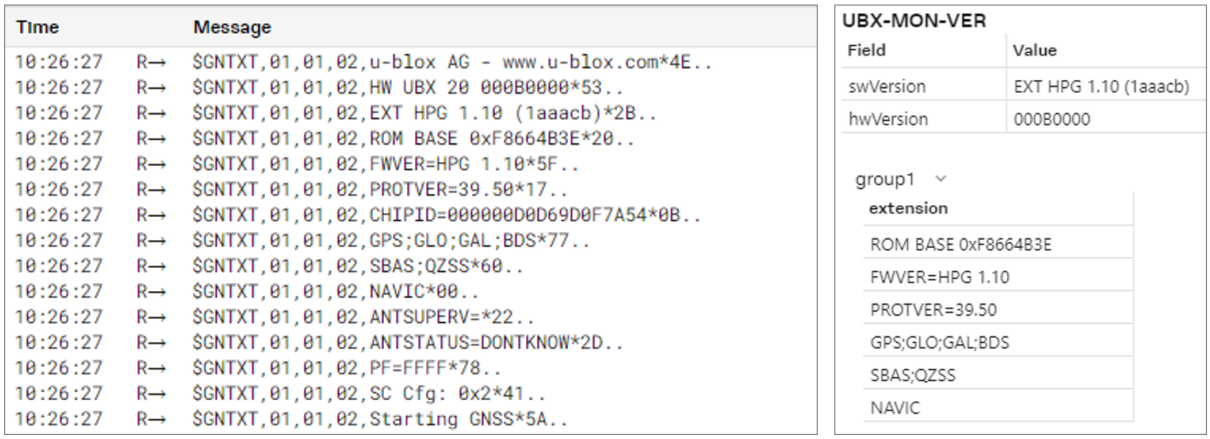
1.2 Firmware and protocol versions

u-blox receivers execute firmware from internal ROM or load an external image and execute it from internal code-RAM.

- If the product does not have internal code-RAM, the firmware runs from the ROM.
- If the product has internal code-RAM but an external image is not available, the firmware runs from the ROM. Some products have only limited ROM and enter boot mode with no GNSS function if an external image is not available.
- If the external firmware image is stored in a flash memory, it is loaded into the code-RAM before execution.
- In some products, the firmware image can be stored in the host system and loaded into the code-RAM from there.

The location and the version of 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 the flash memory or from the host processor, it is indicated by text "EXT". Running from the internal ROM is indicated by text "ROM". 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 boot information:



Time		Message
10:26:27	R→	$GNTXT,01,01,02,u-blox AG - www.u-blox.com*4E..
10:26:27	R→	$GNTXT,01,01,02,HW UBX 20 000B0000*53..
10:26:27	R→	$GNTXT,01,01,02,EXT HPG 1.10 (1aaacb)*2B..
10:26:27	R→	$GNTXT,01,01,02,ROM BASE 0xF8664B3E*20..
10:26:27	R→	$GNTXT,01,01,02,FWVER=HPG 1.10*5F..
10:26:27	R→	$GNTXT,01,01,02,PROTVER=39.50*17..
10:26:27	R→	$GNTXT,01,01,02,CHIPID=000000D0D69D0F7A54*0B..
10:26:27	R→	$GNTXT,01,01,02,GPS;GLO;GAL;BDS*77..
10:26:27	R→	$GNTXT,01,01,02,SBAS;QZSS*60..
10:26:27	R→	$GNTXT,01,01,02,NAVIC*00..
10:26:27	R→	$GNTXT,01,01,02,ANTSUPERV=*22..
10:26:27	R→	$GNTXT,01,01,02,ANTSTATUS=DONTKNOW*2D..
10:26:27	R→	$GNTXT,01,01,02,PF=FFFF*78..
10:26:27	R→	$GNTXT,01,01,02,SC Cfg: 0x2*41..
10:26:27	R→	$GNTXT,01,01,02,Starting GNSS*5A..

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 10 00000000	Hardware version of the u-blox receiver.
	✓	00000000	
✓	✓	ROM SPG 5.10 (000000)	Firmware version and revision identifier.
✓	✓	ROM BASE 0x118B2060	Revision of the underlying boot loader firmware in ROM.
✓	✓	FWVER=SPG 5.10	Product firmware version, 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 • ASP = Automotive standard precision • LDR = ROM bootloader, no GNSS functionality
✓	✓	PROTVER=34.00	Supported protocol version.
✓	✓	MOD=EVK-M101	Module name.
✓	✓	GPS;GLO;GAL;BDS	List of supported major GNSS (see GNSS identifiers).
✓	✓	SBAS;QZSS	List of supported augmentation systems (see GNSS identifiers).
✓	✓	NAVIC	Extended list of supported GNSS (see GNSS identifiers).
✓		ANTSUPERV=AC SD PDoS SR	Configuration of the antenna supervisor, 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
✓		PF=FFF79	Product configuration.

B	M	Example	Information
✓		`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 version and revision numbers should only be used to identify a known firmware version. They are not necessarily numeric nor are they guaranteed to increase with later firmware versions.

☞ All u-blox receivers output the start text, hardware version, and firmware version and revision. Some of the other entries in the boot screen example may be omitted.

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

Firmware version	Version and revision identifier	Protocol version
HPG 2.00	HPG 2.00B00 (5c81c8)	50.00

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. The receiver changes its current configuration immediately after receiving a configuration message. The receiver always uses 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 the integration manual for a basic receiver configuration most commonly used.

1.4 Message 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

Many UBX protocol messages contain infomation about specific satellites. Any single satellite can be identified by a `gnssId` field indicating the GNSS the satellite is part of and an `svId` (SV for space vehicle) field indicating the number of the satellite in that system. Usually, the `svId` is the native number associated with the satellite in the specific GNSS. For example, the Galileo SV4 is identified as gnssId 2, svId 4, while the GPS SV4 is gnssId 0, svId 4.

Some legacy UBX protocol messages combine both the satellite number and the GNSS identification into a one-byte (type U1) field. See the single svid mapping in Satellite identifiers to identify the corresponding GNSS and satellite.

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

Signal identifiers are used when different signals from the same GNSS satellite need to be distinguished (e.g. in the UBX-NAV-SIG message). A separate `sigId` field identifies the signal. These signal 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

Table 1 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

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

GNSS	Abbreviations		UBX gnssId	NMEA system ID		
				2.3 - 4.0	4.10	4.11
Galileo	GAL	E	2	n/a	3	3
BeiDou	BDS	B	3	n/a	(4)[1]	4
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

Table 1: GNSS identifiers

See also NMEA Talker ID.

1.5.3 Satellite identifiers

The satellite numbering scheme for the UBX protocol is provided in Table 2. The satellite numbering scheme for the NMEA protocol is provided in Table 3.

GNSS	SV Range	gnssId:svId	single svid
GPS	G1-G32	0:1-32	1-32
SBAS	S120-S158	1:120-158	120-158
Galileo	E1-E36	2:1-36	211-246
BeiDou	B1-B5	3:1-5	159-163
	B6-B37	3:6-37	33-64
	B38-B63	3:38-63	n/a
QZSS	Q1-Q10	5:1-10	193-202
GLONASS	R1-R32	6:1-32	65-96
	R?	6:255	255
NavIC	N1-N7	7:1-7	247-253
	N8-N14	7:8-14	n/a

Table 2: UBX protocol satellite numbering scheme

		NMEA 2.3 - 4.0		NMEA 4.10		NMEA 4.11	
GNSS	SV Range	strict	extended	strict	extended	strict	extended
GPS	G1-G32	1-32	1-32	1-32	1-32	1-32	1-32
SBAS	S120-S158	33-64	33-64, 152-158	33-64	33-64, 152-158	33-64	33-64, 152-158
Galileo	E1-E36	n/a	301-336	1-36	1-36	1-36	1-36
BeiDou	B1-B5	n/a	401-405	1-5	1-5	1-5	1-5
	B6-B37	n/a	406-437	6-37	6-37	6-37	6-37
	B38-B63	n/a	438-463	38-63	38-63	38-63	38-63
QZSS	Q1-Q10	n/a	193-202	n/a	193-202	1-10	1-10
GLONASS	R1-R32	65-96	65-96	65-96	65-96	65-96	65-96
	R?	null	null	null	null	null	null
NavIC	N1-N7	n/a	n/a	n/a	n/a	1-7	1-7
	N8-N14	n/a	n/a	n/a	n/a	8-14	8-14

Table 3: NMEA protocol satellite numbering scheme

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

the NMEA protocol, system and signal identifiers are in hexadecimal format. An unknown signal identifier is presented as 0 in the NMEA protocol.

	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
Galileo E6 B	2	8	3	5	3	5
Galileo E6 C	2	9	3	5	3	5
Galileo E6 A	2	10	3	4	3	4
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 B3I D1	3	4				
BeiDou B3I D2	3	10				
BeiDou B1 Cp (pilot)	3	5	(4)[3]	N/A	4	3
BeiDou B1 Cd (data)	3	6	(4)[3]	N/A	4	3
BeiDou B2 ap (pilot)	3	7	(4)[3]	N/A	4	5
BeiDou B2 ad (data)	3	8	(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

[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, in this mode, u-blox receivers 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.

Signal	UBX Protocol		NMEA Protocol 4.10		NMEA Protocol 4.11	
	gnssId	sigId	System ID	Signal ID	System ID	Signal ID
NavIC L5 A[2]	7	0	N/A	N/A	6	1

Table 4: Signal identifiers

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.

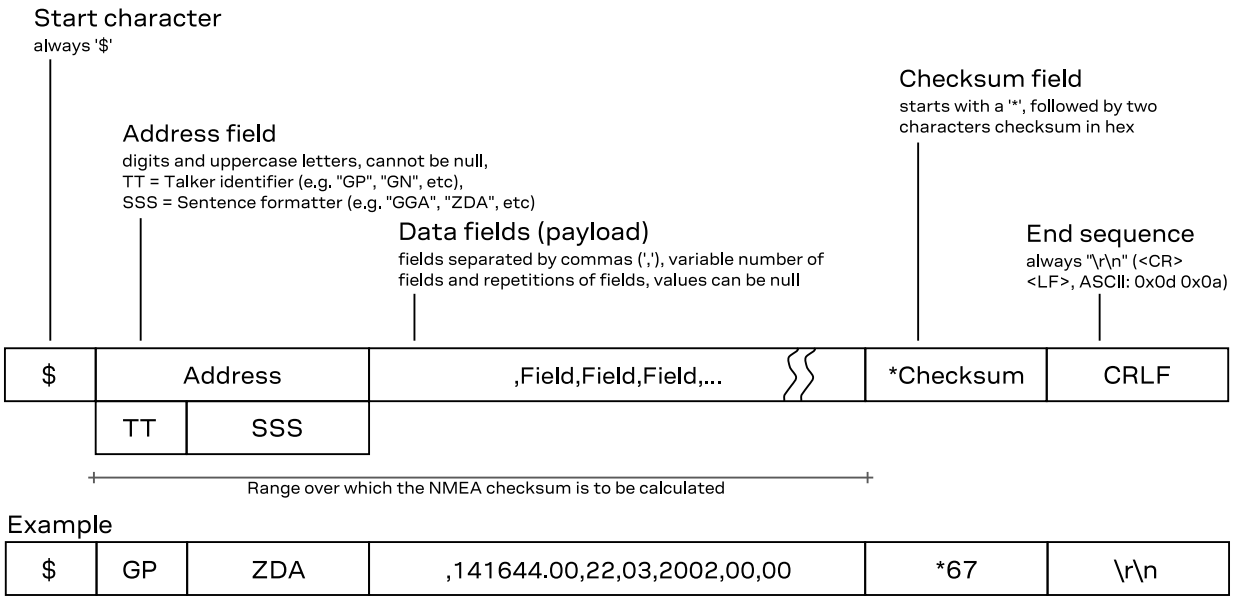
2 NMEA protocol

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 is "GN" (e.g. instead of "GP" for a GPS-only receiver).

GSV Talker and Signal IDs The GSV message reports the signal strength of the visible satellites. In multi-GNSS operation, other messages use the main Talker ID "GN" but the Talker ID in the GSV message is specific to the GNSS it is reporting information for.

The GSV messages are grouped by the Talker and Signal IDs. Separate sets of GSV messages are sent for each GNSS and signal. The Signal ID of a satellite may be unknown. Such satellites are presented in their own set with Signal ID 0. Grouping the GSV messages by the Signal ID is supported in protocol versions 27.12 and later.

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)

GNSS	Talker ID	Comments
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

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. In NMEA GNS, u-blox uses a non-standard implementation where same single status is reported for all enabled and not filtered out constellations.

NMEA Message	GLL, RMC	GGA	GLL, VTG	RMC, GNS
Field	**status**[5]	**quality**[6]	**posMode**[7]	**posMode**[7]
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
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
```

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. In NMEA GNS, u-blox uses a non-standard implementation where same single status is reported for all enabled and not filtered out constellations.

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.

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-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	**NMEA-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. The first four characters indicate the status for GPS, GLONASS, Galileo and BeiDou. Note that the NMEA GNS message only reports a single status. It indicates the status for all enabled constellations that have not been filtered out. To obtain a more detailed status report, refer to the status provided in the UBX messages.
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
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	**NMEA-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. The messages are grouped by the signal ID and separate messages are output for each signal ID. (supported for protocol versions 27.12 and later)
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`
	`$GPGSV,1,1,01,03,05,218,,0*59\r\n`
	`$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	-	00000078A9FBAD5	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 PUBX messages

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

2.8.1 CONFIG (PUBX,41)

2.8.1.1 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

2.8.2 POSITION (PUBX,00)

2.8.2.1 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

2.8.2.2 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

2.8.3 RATE (PUBX,40)

2.8.3.1 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*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

2.8.4 SVSTATUS (PUBX,03)

2.8.4.1 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

2.8.4.2 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

2.8.5 TIME (PUBX,04)

2.8.5.1 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

2.8.5.2 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

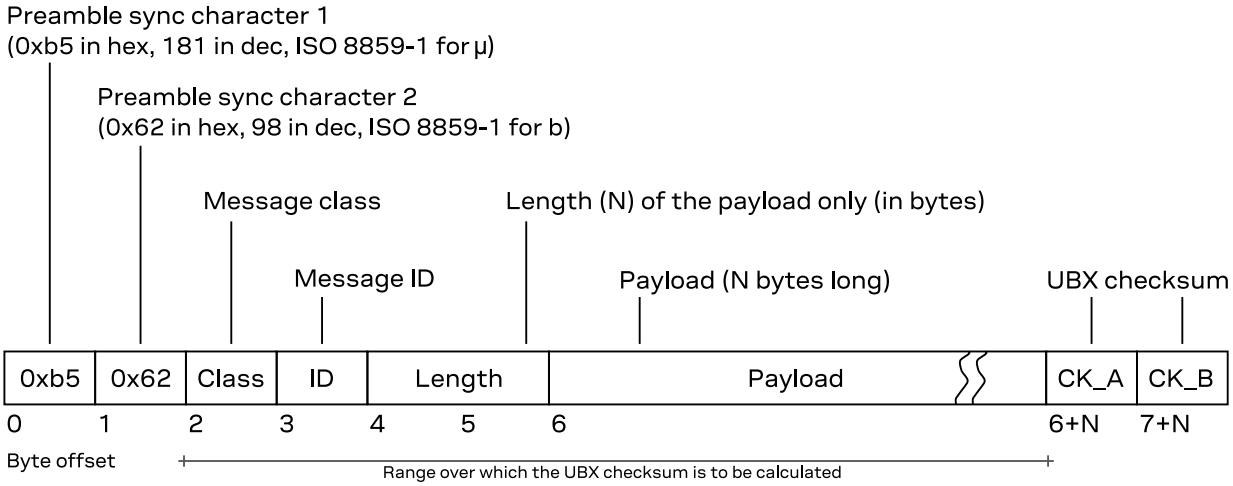
3.1 UBX protocol key features

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
RU1_3	unsigned 8-bit integer interpreted as binary floating point with 3 bit exponent, `eeeb bbbb` with `b` the base and `e` the exponent, (value & 0x1f) << (value >> 5)	1	$0...31 \cdot 2^7$ (non-continuous)	$\sim 2^{(value >> 5)}$
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
RU2_5	unsigned 16-bit integer interpreted as binary floating point with 5 bit exponent, `eeee ebbb bbbb bbbb` with `b` the base and `e` the exponent, (value & 0x7FF) << (value >> 11)	2	$0...2047 \cdot 2^{31}$ (non-continuous)	$\sim 2^{(value >> 11)}$

Name	Type	Size (Bytes)	Range	Resolution
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
R4	IEEE 754 single (32-bit) precision	4	$-2^{127}...2^{127}$	~ value·2^{-24}
R8	IEEE 754 double (64-bit) precision	8	$-2^{1023}...2^{1023}$	~ value·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	*Header*	*Class ID*	*Length (bytes)*	*Payload*	*Checksum*
Structure	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 Message 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-CFG	0x06 0x09	• Clear, save and load configurations (Command)
UBX-CFG-MSG	0x06 0x01	• Poll a message configuration (Poll request) • Set message rate(s) (Get/set) • Set message rate (Get/set)
UBX-CFG-OTP	0x06 0x41	• Poll OTP memory content (Poll request) • Write OTP memory content (Set) • Write file 0x20: USB vendor ID (Set) • Write file 0x21: USB vendor string (Set) • Write file 0x22: USB product ID (Set) • Write file 0x23: USB product string (Set) • Write file 0x36: oscillator offset calibration (Set) • Write file 0xa4: receiver configuration items (Set)
UBX-CFG-PT2	0x06 0x59	• Production test configuration (Set)
UBX-CFG-RST	0x06 0x04	• Reset receiver / Clear backup data structures (Command)
UBX-CFG-USBTEST	0x06 0x58	• USB testing (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)

Message	Class/ID	Description (Type)
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-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 for satellites svId 1..37 (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)
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 ZYX (Input) • Initial position assistance LLH (Input) • Initial time assistance UTC (Input) • Initial time assistance GNSS (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-HW3	0x0a 0x37	• I/O pin status (Periodic/polled)
UBX-MON-PATCH	0x0a 0x27	• Poll request for installed patches (Poll request) • Installed patches (Polled)

Message	Class/ID	Description (Type)
UBX-MON-PT2	0x0a 0x2b	• Multi-GNSS production test monitor (Periodic/polled)
UBX-MON-RF	0x0a 0x38	• RF information (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-VER	0x0a 0x04	• Poll receiver and software version (Poll request) • 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-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)
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-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-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-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)

Message	Class/ID	Description (Type)
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-SIG	0x27 0x09	• Signal security information (Periodic/polled)
UBX-SEC-SIGLOG	0x27 0x10	• Signal security log (Periodic/polled)
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-VRFY	0x0d 0x06	• Sourced time verification (Periodic/polled)
UBX-UPD – Firmware update messages		
UBX-UPD-CERASE	0x09 0x16	• Chip erase the connected SQI flash (Command) • Chip erase the connected SQI flash acknowledgement (Output)
UBX-UPD-CRC	0x09 0x0d	• Check CRC of binary (Command)
UBX-UPD-ERASE	0x09 0x0b	• Erase flash sector (Command) • Erase flash sector acknowledgement (Output)
UBX-UPD-FLDET	0x09 0x08	• Get the flash manufacturer and device IDs (Poll request) • Flash manufacturer and device IDs (Get)
UBX-UPD-FLWRI	0x09 0x0c	• Write flash data (area must be erased before) (Command) • Write flash data success indication (Output)
UBX-UPD-IDEN	0x09 0x06	• Identify flash loader version (Poll request) • Flash loader version (Get)
UBX-UPD-POS	0x09 0x15	• Enable PLL during safeboot (Command)
UBX-UPD-RBOOT	0x09 0x0e	• Perform a watchdog reset (Command)
UBX-UPD-ROM	0x09 0x25	• ROM CRC (Polled)
UBX-UPD-SAFE	0x09 0x07	• Boot in safe environment from ROM or RAM (Command) • Start flash loader task (Command)
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-CFG (0x06 0x09)

3.10.1.1 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. The receiver replies with a single UBX-ACK-ACK or UBX-ACK-NAK. A UBX-ACK-ACK indicates that all operations were successful. A UBX-ACK-NAK indicates that at least one of the configured operations was unsuccessful. It is recommended to send individual commands for a more comprehensive monitoring of the success or not of the individual operations. ☞ 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					

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

3.10.2.1 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

3.10.2.2 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)

3.10.2.3 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

3.10.3 UBX-CFG-OTP (0x06 0x41)

3.10.3.1 Poll OTP memory content

Message	**UBX-CFG-OTP** **Poll OTP memory content**
Type	Poll request
Comment	See section OTP memory in the integration manual for details.

Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x06	0x41	0	*see below*	CK_A CK_B

Payload	This message has no payload.

3.10.3.2 Write OTP memory content

Message	**UBX-CFG-OTP** **Write OTP memory content**
Type	Set
Comment	Writes content to the OTP meomory. A UBX-ACK-ACK message will be returned when the content was successfully written, a UBX-ACK-NAK message when there was a problem.

See section OTP memory in the integration manual for details.

Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x06	0x41	[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	`payload`	-	-	Payload. Use u-center to compose the message.
End of repeated group (N times)					

3.10.3.3 Write file 0x20: USB vendor ID

Message	**UBX-CFG-OTP** **Write file 0x20: USB vendor ID**
Type	Set
Comment	Writes the USB vendor ID to the OTP memory. A UBX-ACK-ACK message will be returned when the content was successfully written, a UBX-ACK-NAK message when there was a problem. See section OTP memory in the integration manual for details.

Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x06	0x41	14	*see below*	CK_A CK_B

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	U1	`version`	-	-	Message version (0x04 for this version)
1	U1	`operation`	-	-	Operation • 0x01 - write OTP memory file to chip 0
2	U1	`fileID`	-	-	File identifier (0x20 for this file)
3	U1[11]	`data`	-	-	File data: use u-center to compose the message

3.10.3.4 Write file 0x21: USB vendor string

Message	**UBX-CFG-OTP** **Write file 0x21: USB vendor string**
Type	Set
Comment	Writes the USB vendor string to the OTP memory. A UBX-ACK-ACK message will be returned when the content was successfully written, a UBX-ACK-NAK message when there was a problem. See section OTP memory in the integration manual for details.

Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x06	0x41	12 + [0..n]	*see below*	CK_A CK_B

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	U1	`version`	-	-	Message version (0x04 for this version)
1	U1	`operation`	-	-	Operation • 0x01 - write OTP memory file to chip 0
2	U1	`fileID`	-	-	File identifier (0x21 for this file)
3	U1[9]	`data`	-	-	File data: use u-center to compose the message
Start of repeated group (N times)					
12 + n	U1	`data2`	-	-	Optional file data

End of repeated group (N times)

3.10.3.5 Write file 0x22: USB product ID

Message	**UBX-CFG-OTP** **Write file 0x22: USB product ID**
Type	Set
Comment	Writes the USB product ID to the OTP memory. A UBX-ACK-ACK message will be returned when the content was successfully written, a UBX-ACK-NAK message when there was a problem. See section OTP memory in the integration manual for details.

Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x06	0x41	14	*see below*	CK_A CK_B

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	U1	`version`	-	-	Message version (0x04 for this version)
1	U1	`operation`	-	-	Operation • 0x01 - write OTP memory file to chip 0
2	U1	`fileID`	-	-	File identifier (0x22 for this file)
3	U1[11]	`data`	-	-	File data: use u-center to compose the message

3.10.3.6 Write file 0x23: USB product string

Message	**UBX-CFG-OTP** **Write file 0x23: USB product string**
Type	Set
Comment	Writes the USB product string to the OTP memory. A UBX-ACK-ACK message will be returned when the content was successfully written, a UBX-ACK-NAK message when there was a problem. See section OTP memory in the integration manual for details.

Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x06	0x41	12 + [0..n]	*see below*	CK_A CK_B

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	U1	`version`	-	-	Message version (0x04 for this version)
1	U1	`operation`	-	-	Operation • 0x01 - write OTP memory file to chip 0
2	U1	`fileID`	-	-	File identifier (0x23 for this file)
3	U1[9]	`data`	-	-	File data: use u-center to compose the message
Start of repeated group (N times)					
12 + n	U1	`data2`	-	-	Optional file data
End of repeated group (N times)					

3.10.3.7 Write file 0x36: oscillator offset calibration

Message	**UBX-CFG-OTP** **Write file 0x36: oscillator offset calibration**
Type	Set
Comment	Writes the oscillator offset calibration to the OTP memory. A UBX-ACK-ACK message will be returned when the content was successfully written, a UBX-ACK-NAK message when there was a problem.

See section OTP memory in the integration manual for details.

☞ This message is not applicable to TCXO.

Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x06	0x41	17	*see below*	CK_A CK_B

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	U1	version	-	-	Message version (0x04 for this version)
1	U1	operation	-	-	Operation: 0x01 - write OTP memory file to chip 0
2	U1	fileID	-	-	File identifier (0x36 for this file)
3	X1	lengthMask	-	-	File length and mask (0x85 for this file)
bits 6...0	U:7	length	-	-	The length of the file (0x05 for this file)
bit 7	U:1	mask	-	-	Reserved: set to 1. Ignored in Monch
4	U1[4]	reserved0	-	-	Reserved
8	U4	key	-	-	A key, which protects against accidental changes (set to 0x0512ef28)
12	X4	oscillatorOffsetCalibration1	-	-	Oscillator offset calibration
bits 13...0	I:14	offset	-	-	Offset of the oscillator [0.1ppm]
bits 17...14	U:4	precision	-	-	Precision of the offset [1ppm]
bit 18	U:1	offsetInvalid	-	-	Flag to indicate if the offset (and precision) is invalid
bit 19	U:1	maxCalibDeviationInvalid	-	-	Flag to indicate if maxCalibDeviation is invalid
bits 24...20	U:5	maxCalibDeviation	-	-	maximum calibration deviation [1ppm]
bits 31...25	U:7	notUsed	-	-	Reserved: set to 0
16	U1	extraByte	-	-	Extra byte: set to 0xFF

3.10.3.8 Write file 0xa4: receiver configuration items

Message	**UBX-CFG-OTP** **Write file 0xa4: receiver configuration items**
Type	Set
Comment	Writes the configuration data (key ID and value) for one or more configuration items to the OTP memory. Any supported configuration item can be set this way, as long as enough free OTP memory is available. It is possible to write multiple files of this type. To minimize usage of OTP memory, it is recommended to combine multiple configuration data in a single write file operation. See section OTP memory in the integration manual for details.

Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x06	0x41	12 + [0..n]	*see below*	CK_A CK_B

Payload description:

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

0		U1	version	-	-	Message version (0x04 for this version)
1		U1	operation	-	-	Operation: 0x01 - write OTP memory file to chip 0
2		U1	fileID	-	-	File identifier (0xA4 for this file)
3		X1	lengthMask	-	-	File length and mask
	bits 6...0	U:7	length	-	-	File length (max 127). The file length depends on the number and data type of the configuration items used. Available space in OTP is limited and should be checked before writing this file. The items are stored in groups of key ID, value without padding. The key is always stored in 4 bytes; the value is stored in a number of bytes depending on the associated data type, as described in section Configuration data.
	bit 7	U:1	mask	-	-	Reserved: set to 1. Ignored in Monch
4		U4	CRC	-	-	CRC of the message. Calculate an IEEE-802.3 32-bit CRC over the data in the fields version, operation, fileID, lengthMask and cfgData. Create the IEEE-802.3 lookup table CRCTable of 256 unsigned 32-bit integers with polynomial 0xEDB88320. For each byte in the data, set 32-bit unsigned integer index = (crc32 XOR byte) AND 0xFF; crc32 = ((crc32 >> 8) AND 0x00FFFFFF)) XOR CRCTable[index]. Use initial value (seed) of 0xBAADBAAD for the 32 bit unsigned integer crc32. Do not invert/reflect/XOR input, output, or any intermediate values.
8		U4	key	-	-	A key, which protects against accidental changes (set to 0x0512ef28)
Start of repeated group (N times)						
12 + n		U1	cfgData	-	-	Configuration data (key and value pairs)
End of repeated group (N times)						

3.10.4 UBX-CFG-PT2 (0x06 0x59)

3.10.4.1 Production test configuration

Message	**UBX-CFG-PT2** **Production test configuration**					
Type	Set					
Comment	Activate and set configuration for Production test mode. This allows setting a variable number of satellite signal descriptors (no more than the number of RF channels of the receiver). Activating also enables output message UBX-MON-PT2.					
Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x06	0x59	12 + [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	activate	-	-	Production test mode
bit 0	U:1	enable	-	-	1=activate test mode, 0=deactivate test mode

bits 7...6	U:2	lnaMode	-	-	Set to 0x00
2	U1	extint	-	-	Input pin for the optional frequency aiding • 0x00=EXTINT0 • 0x01=EXTINT1 • 0xFF=no frequency aiding
3	U1	reAcqCno	-	dBHz	C/N0 threshold to force re-acquisition. Set to 10-15 dB lower than the expected C/N0 ratio (set a value > 0)
4	U4	refFreq	-	Hz	Reference frequency
8	U4	refFreqAcc	-	ppb	Reference frequency accuracy
Start of repeated group (N times)					
12 + n·4	U1	gnssId	-	-	GNSS identifier (see Satellite numbering)
13 + n·4	U1	svId	-	-	Satellite identifier (see Satellite numbering)
14 + n·4	U1	sigId	-	-	Signal identifier. 0 is the only value currently supported.
15 + n·4	U1	accsId	-	-	Access identifier, used to select frequency channel in range (0-13) for GLONASS (0 = -7, 1 = -6, ..., 12 = +5, 13 = +6). Ignored for all other GNSS.
End of repeated group (N times)					

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

3.10.5.1 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 • 0x0a = Hardware reset (via PWSEQ), retaining BBR contents
3	U1	reserved0	-	-	Reserved

3.10.6 UBX-CFG-USBTEST (0x06 0x58)

3.10.6.1 USB testing

Message	**UBX-CFG-USBTEST** **USB testing**					
Type	Set					
Comment						
Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x06	0x58	2	*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	usbPinState	-	-	Define the USB test state and output • 0 = Test mode disabled, normal pin usage • 1 = Set tristate: DP=Z DM=Z • 2 = Output DP=1 DM=0 • 3 = Output DP=0 DM=1 • 4 = Output DP=0 DM=0

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

3.10.7.1 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, 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.
- The provided keys can be complete key values (group and item specifiers) or wild-card specifications. A complete key value constitutes a deletion request for one key-value pair. A key value with a valid group specifier and 0xffff in the item part of the key value (bits 0-15) constitutes a deletion 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 deletion request for all items known to the receiver in all groups.

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)					

3.10.7.2 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, 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.
- The provided keys can be complete key values (group and item specifiers) or wild-card specifications. A complete key value constitutes a deletion request for one key-value pair. A key value with a valid group specifier and 0xffff in the item part of the key value (bits 0-15) constitutes a deletion 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 deletion request for all items known to the receiver in all groups.

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)

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

3.10.8.1 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 • 3 - Image layer • 4 - OTP layer • 5 - Pin layer • 6 - ROM 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)

3.10.8.2 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 • 3 - Image layer • 4 - OTP layer • 5 - Pin layer • 6 - ROM layer • 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)					

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

3.10.9.1 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)					

3.10.9.2 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.					
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						
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-STRING (0x21 0x04)

3.12.6.1 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]	msgPayload Start	-	-	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 for satellites svId 1..37

Message	**UBX-MGA-BDS-EPH** **BeiDou ephemeris assistance for satellites svId 1..37**					
Type	Input					
Comment	This message allows the delivery of BeiDou D1/D2 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 from D1/D2 ephemeris to a receiver. See section AssistNow online in the integration manual for details. This message allows the delivery of health assistance data for all satellites with svId 1 to 30.					
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 D2 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 ZYX

Message	**UBX-MGA-INI-POS_XYZ** **Initial position assistance ZYX**
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 LLH

Message	**UBX-MGA-INI-POS_LLH** **Initial position assistance LLH**
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 UTC

Message	**UBX-MGA-INI-TIME_UTC** **Initial time assistance UTC**
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	X1	bitfield0	-	-	bitfield:
bit 0	U:1	trustedSource	-	-	Time is provided from a trusted source. Potentially usable for replay attack detection • 0: Unknown • 1: Time source can be trusted for spoofing detection
12	U4	ns	-	ns	Nanoseconds, from 0 to 999,999,999
16	U2	tAccS	-	s	Seconds part of time accuracy
18	U1[2]	reserved0	-	-	Reserved
20	U4	tAccNs	-	ns	Nanoseconds part of time accuracy, from 0 to 999,999,999

3.13.7.4 Initial time assistance GNSS

Message	**UBX-MGA-INI-TIME_GNSS** **Initial time assistance GNSS**					
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	X1	bitfield0	-	-	bitfield:
bit 0	U:1	trustedSource	-	-	Time is provided from a trusted source. Potentially usable for replay attack detection • 0: Unknown • 1: Time source can be trusted for spoofing detection
5	U1	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*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x13	0x05	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	-	-	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*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x13	0x05	12	*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[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)
bits 4...2	U:3	outputPort	-	-	Output port: Reports the port from which this message was output from. • 0 = N/A • 1 = I2C • 2 = UART1 • 3 = UART2 • 4 = USB • 5 = SPI
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 OTP memory 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-HW3 (0x0a 0x37)

3.14.3.1 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	U1	`reserved1`	-	-	Reserved
23 + n·6	U1	`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
bits 11...10	U:2	`testModeStatus`	-	-	Testmode status: 0=Unknown, 1=Pin in testmode, 2=Pin not in testmode
26 + n·6	U1	`VP`	-	-	Virtual pin mapping
27 + n·6	U1	`reserved2`	-	-	Reserved
End of repeated group (`nPins` times)					

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

3.14.4.1 Poll request for installed patches

Message	**UBX-MON-PATCH** **Poll request for installed patches**
Type	Poll request

Comment						
Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x0a	0x27	0	*see below*	CK_A CK_B
Payload	This message has no payload.					

3.14.4.2 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*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x0a	0x27	4 + nEntries·16	*see below*	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: OTP, 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)					

3.14.5 UBX-MON-PT2 (0x0a 0x2b)

3.14.5.1 Multi-GNSS production test monitor

Message	**UBX-MON-PT2** **Multi-GNSS production test monitor**					
Type	Periodic/polled					
Comment						
Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x0a	0x2b	24 + numRfChn·28 + numSvSigDesc·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	`testMode`	-	-	currently active test mode
2	U1	`numRfChn`	-	-	number of RF channels reported in this message

3	U1	numSvSigDesc	-	-	number of satellite signal descriptors reported in this message
4	U4	testRunTime	-	ms	test runtime since channel assignment
8	I4	clkDriftAid	-	ppb	clock drift of receiver clock relative to extint source (with an offset of 1e9: 1000000000 means 'zero doppler')
12	I4	clkDriftTrk	-	ppb	clock drift of receiver clock relative to tracked GNSS signals (without offset: 0 means 'zero doppler')
16	U4	rtcFreq	-	Hz	RTC frequency
20	U4	postStatus	-	-	Power On Self Test status mask
Start of repeated group (numRfChn times)					
24 + n·28	U1	rfPga	-	-	RF gain amplifier setting
25 + n·28	U1[27]	reserved0	-	-	Reserved
End of repeated group (numRfChn times)					
Start of repeated group (numSvSigDesc times)					
24 + numRfChn·28 + n·36	U1	gnssId	-	-	GNSS identifier (see Satellite numbering)
25 + numRfChn·28 + n·36	U1	svId	-	-	GNSS identifier (see Satellite numbering)
26 + numRfChn·28 + n·36	U1	sigId	-	-	Signal identifier. 0 is the only value currently supported.
27 + numRfChn·28 + n·36	U1	accsId	-	-	Access identifier, used to indicate frequency channel in range (0-13) for GLONASS (0 = -7, 1 = -6, ..., 12 = +5, 13 = +6). The value should be ignored for all other GNSS.
28 + numRfChn·28 + n·36	U2	cnoMin	2^-8	dBHz	minimum CNo across all channels tracking this satellite signal
30 + numRfChn·28 + n·36	U2	cnoMax	2^-8	dBHz	maximum CNo across all channels tracking this satellite signal
32 + numRfChn·28 + n·36	U1[14]	reserved1	-	-	Reserved
46 + numRfChn·28 + n·36	U1	carrPhDevMax	2^-8	cycles	carrier phase measurement deviation maximum across all associated channels (1 cycle = 360 deg)
47 + numRfChn·28 + n·36	X1	signalInfo	-	-	signal information
bit 0	U:1	ifChnIdValid	-	-	Flag to show if channel input number (ifChnId) is valid
bits 3...1	U:3	ifChnId	-	-	Channel input number (0,1..numRfChn-1) for this signal corresponding to rfChannels above
48 + numRfChn·28 + n·36	U1	codeLock Success	-	%	percentage of channels codelocked

49 + numRfChn·28 + n·36	U1	phaseLock Success	-	%	percentage of channels codelocked
50 + numRfChn·28 + n·36	U2	minCodeLock Time	-	ms	minimum codelock time across all associated channels
52 + numRfChn·28 + n·36	U2	maxCodeLock Time	-	ms	maximum codelock time across all associated channels
54 + numRfChn·28 + n·36	U2	minPhaseLock Time	-	ms	minimum phaselock time across all associated channels
56 + numRfChn·28 + n·36	U2	maxPhaseLock Time	-	ms	maximum phaselock time across all associated channels
58 + numRfChn·28 + n·36	U1[2]	reserved2	-	-	Reserved
End of repeated group (numSvSigDesc times)					

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

3.14.6.1 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*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x0a	0x38	4 + nBlocks·24	*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	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 or flag unavailable, 1 = ok - no significant jamming, 2 = warning - interference visible but fix OK, 3 = critical - interference visible and no fix). This flag is deprecated in protocol versions that support UBX-SEC-SIG (version 0x02) and always reported as 0; instead jammingState in UBX-SEC-SIG should be monitored.
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, as percentage of maximum gain, range 0 to 8191 (100%)
20 + n·24	U1	cwSuppression	-	-	CW interference suppression level, 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)					

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

3.14.7.1 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*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x0a	0x21	1	*see below*	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

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

3.14.8.1 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`	2^-2	dB	Spectrum data (number of points = span/res) [Uuu.ff dB]
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)					

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

3.14.9.1 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 system 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 11-System reset
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

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

3.14.10.1 Poll receiver and software version

Message	**UBX-MON-VER** **Poll receiver and software version**					
Type	Poll request					
Comment						
Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x0a	0x04	0	*see below*	CK_A CK_B
Payload	This message has no payload.					

3.14.10.2 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	

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-POSECEF (0x01 0x01)

3.15.10.1 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

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

3.15.11.1 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

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

3.15.12.1 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 (applicable to heading products only)

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
bit 13	U:1	authTime	-	-	Flag that indicates if the output time has been validated against an external trusted time source • 0 = Time is not authenticated • 1 = Time is authenticated
bit 14	U:1	nmaFixStatus	-	-	Flag assigned to a fix that has been computed mixing satellites with data authenticated through Navigation Message Authentication (NMA) methods and satellites using unauthenticated data. The fix is flagged as Verified when internal cross-checks validates the unauthenticated signals against the authenticated ones. Note that Not Verified status does not imply directly spoofing attacks, to identify spoofing alerts refer to UBX-SEC-SIG. • 0 = Not Verified: The mixed solution does not agree with the NMA authenticated data or the comparison could not be performed, e.g., not enough authenticated SVs to extrapolate the result or cryptographic data not decoded yet • 1 = Verified: The mixed solution agrees with the NMA authenticated data Currently, the only existing NMA method is Galileo Open Service Navigation Message Authentication (OSNMA) protocol.
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.

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

3.15.13.1 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	`relPosHP Length`	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

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

3.15.14.1 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.					

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

3.15.15.1 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)					

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

3.15.16.1 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)						

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

3.15.17.1 Signal information

Message	**UBX-NAV-SIG** **Signal information**
Type	Periodic/polled
Comment	This message displays information about signals currently tracked or searched by the receiver.

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
bit 9	U:1	authStatus	-	-	Authentication status of the navigation data used to compute the satellite's position in current navigation epoch. If the authentication fails, the navigation data is not used so the authentication status in this message can take only two values: • 0 = Unknown • 1 = Authenticated Note that currently the only data authentication function is provided by Galileo Open Service Navigation Message Authentication (OSNMA) protocol for E1 I/NAV message.
20 + n·16	U1[4]	reserved1	-	-	Reserved
End of repeated group (numSigs times)					

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

3.15.18.1 Receiver navigation status

Message	**UBX-NAV-STATUS** **Receiver navigation status**					
Type	Periodic/polled					
Comment	See important comments concerning the validity of the 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. For details, see section iTOW timestamps in the integration manual.
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 (for details, see section Time validity in the Integration manual)
	bit 3	U:1	towSet	-	-	1 = Time of Week valid (for details, see section Time validity in the integration manual)
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

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

3.15.19.1 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*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x01	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

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

3.15.20.1 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

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

3.15.21.1 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

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

3.15.22.1 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

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

3.15.23.1 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

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

3.15.24.1 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. If the value is 0, then the amount of leap seconds did not change and the event should be ignored.
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	dateOfLsGps Wn	-	-	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	dateOfLsGps Dn	-	-	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	validTimeToLs Event	-	-	1 = Valid time to next leap second event or from the last leap second event if no future event scheduled.

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

3.15.25.1 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

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

3.15.26.1 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
bit 3	U:1	authStatus	-	-	Indicates if the parameters used to convert GNSS time into UTC time have been authenticated. • 0 = Unknown • 1 = Authenticated Note that currently the only data authentication function is provided by Galileo Open Service Navigation Message Authentication (OSNMA) protocol for E1 I/NAV message. Systems other than EU UTC can be authenticated indirectly only using the above information.
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

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

3.15.27.1 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

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

3.15.28.1 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-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.

3.16.1 UBX-RXM-COR (0x02 0x34)

3.16.1.1 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	reserved0	-	-	Reserved
3	U1	reserved1	-	-	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: Decrypted
8	U2	msgType	-	-	Message type
10	U2	msgSubType	-	-	Message subtype

3.16.2 UBX-RXM-MEASX (0x02 0x14)

3.16.2.1 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)					

3.16.3 UBX-RXM-RAWX (0x02 0x15)

3.16.3.1 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.					
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)					

3.16.4 UBX-RXM-RLM (0x02 0x59)

3.16.4.1 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

3.16.4.2 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

3.16.5 UBX-RXM-RTCM (0x02 0x32)

3.16.5.1 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

3.16.6 UBX-RXM-SFRBX (0x02 0x13)

3.16.6.1 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 16, 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)					

3.16.7 UBX-RXM-SPARTN (0x02 0x33)

3.16.7.1 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

3.16.8 UBX-RXM-SPARTNKEY (0x02 0x36)

3.16.8.1 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.					

3.16.8.2 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 'current'. If the message contains a second key, that key shall be stored as 'next'. • If the receiver has two (2) active keys (current and next), the transferred key(s) shall be stored as 'current' and 'next'. 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)					

3.17 UBX-SEC (0x27)

The messages in the UBX-SEC class are used for security features of the receiver.

3.17.1 UBX-SEC-SIG (0x27 0x09)

3.17.1.1 Signal security information

Message	**UBX-SEC-SIG** **Signal security information**					
Type	Periodic/polled					
Comment	Information related to the security, i.e. availability and integrity, of the signals.					
Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x27	0x09	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[3]	reserved0	-	-	Reserved
4	X1	jamFlags	-	-	Information related to jamming/interference
bit 0	U:1	jamDetEnabled	-	-	Flag indicates whether jamming/interference detection is enabled
bits 2...1	U:2	jammingState	-	-	Jamming/interference state • 0: Unknown • 1: No jamming indicated • 2: Warning; jamming indicated but fix OK • 3: Critical; jamming indicated and no fix
5	U1[3]	reserved1	-	-	Reserved
8	X1	spfFlags	-	-	Information related to GNSS spoofing
bit 0	U:1	spfDetEnabled	-	-	Flag indicates whether spoofing detection is enabled
bits 3...1	U:3	spoofingState	-	-	Spoofing state • 0: Unknown • 1: No spoofing indicated • 2: Spoofing indicated • 3: Spoofing affirmed Note that the spoofing state value only reflects the detector state for the current navigation epoch. 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.
9	U1[3]	reserved2	-	-	Reserved

3.17.1.2 Signal security information

Message	**UBX-SEC-SIG** **Signal security information**					
Type	Periodic/polled					
Comment	Information related to the security, i.e. availability and integrity, of the signals.					
Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x27	0x09	4 + jamNumCentFreqs·4	*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	`sigSecFlags`	-	-	Signal security flags, providing high-level jamming and spoofing detector information
bit 0	$U_{:1}$	`jamDetEnabled`	-	-	Flag indicates whether jamming detection is enabled
bits 2...1	$U_{:2}$	`jamState`	-	-	Jamming state • 0: Unknown • 1: No jamming indicated • 2: Warning; jamming indicated *0: Unknown*, denotes that the currently available information is not sufficient to judge whether the receiver is jammed or not. This may occur at receiver start up (or more generally when the receiver is in a mode, where jamming detection is hindered) or when the jamming indicator is disabled. *1: No jamming indicated*: the jamming indicator is enabled and does not sense any significant jamming. *2: Warning; jamming indicated*: the jamming indicator is indicating jamming which has a significant impact on the signal tracking. (The list *jamPerCentFreq* can be checked to find out which frequency bands are jammed.)
bit 3	$U_{:1}$	`spfDetEnabled`	-	-	Flag indicates whether spoofing detection is enabled
bits 6...4	$U_{:3}$	`spfState`	-	-	Spoofing state • 0: Unknown • 1: No spoofing indicated • 2: Spoofing indicated • 3: Spoofing affirmed
2	U1	`reserved0`	-	-	Reserved
3	U1	`jamNumCentFreqs`	-	-	The number of center frequencies we provide jamming information for (subsequent messages)
Start of repeated group (`jamNumCentFreqs` times)					
4 + n·4	X4	`jamStateCentFreq`	-	-	Jamming state of signals sharing a given center frequency Note that jamming information is only provided for center frequencies related to at least one in-use signal, for which a sufficient amount of information is currently available to judge if it is affected by jamming.
bits 23...0	$U_{:24}$	`centFreq`	-	-	Center frequency in [kHz], floored to the nearest kHz multiple

bit 24	U:1	jammed	-	-	Flag indicates whether signals on the given center frequency are considered jammed
End of repeated group (jamNumCentFreqs times)					

3.17.2 UBX-SEC-SIGLOG (0x27 0x10)

3.17.2.1 Signal security log

Message	**UBX-SEC-SIGLOG** **Signal security log**					
Type	Periodic/polled					
Comment	This message provides a log of past signal security related events, that is, events related to jamming and spoofing. Each event is a combination of a detection type and a event type, where the event type 'indication started' and 'indication stopped' and also the event type 'indication triggered' and 'indication timed-out' form a pair. A maximum of 16 events are logged; after the log is filled, recent events take precedence over past events in the log. Power cycles and restarts of the receiver reset the log, deleting its content. Note: It is advised not to restart the receiver while it's indicating spoofing.					
Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x27	0x10	8 + numEvents·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	numEvents	-	-	Number of events
2	U1[6]	reserved0	-	-	Reserved
Start of repeated group (numEvents times)					
8 + n·8	U4	timeElapsed	-	s	Seconds elapsed since this event Special value 0xFFFFFFFF: more than 45 days
12 + n·8	U1	detectionType	-	-	Type of the spoofing or jamming detection: • 0 = simulated signal • 1 = abnormal signal • 2 = INS/GNSS mismatch • 3 = abrupt changes in GNSS signal • 4 = jamming indicated • 5 = authentication failed • 6 = replayed signals
13 + n·8	U1	eventType	-	-	Type of the event: • 0 = indication started • 1 = indication stopped • 2 = indication triggered • 3 = indication timed-out Note: Single epoch events, caused by abrupt changes due to switching from the real to the spoofing signal or vice versa are handled as time-out events. This means that the time-out event is reported after a certain cool off period which is not related to any observations in the signal. The other detection types make use of 'start' and 'stop' event types.
14 + n·8	U1[2]	reserved1	-	-	Reserved
End of repeated group (numEvents times)					

3.17.3 UBX-SEC-UNIQID (0x27 0x03)

3.17.3.1 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

3.17.3.2 Unique chip ID

Message	**UBX-SEC-UNIQID** **Unique chip ID**					
Type	Output					
Comment	This message is used to retrieve a unique chip identifier (48 bits, 6 bytes).					
Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x27	0x03	10	*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	U1[3]	`reserved0`	-	-	Reserved
4	U1[6]	`uniqueId`	-	-	Unique chip ID

3.18 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.

3.18.1 UBX-TIM-TM2 (0x0d 0x03)

3.18.1.1 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 CFG-TP Configuration Items 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 CFG-TP Configuration Items for tpIdx=0) • 2=Time base is UTC (the variant according to the configuration in CFG-NAVSPG-* configuration items)
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

3.18.2 UBX-TIM-VRFY (0x0d 0x06)

3.18.2.1 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

3.19 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.

3.19.1 UBX-UPD-CERASE (0x09 0x16)

3.19.1.1 Chip erase the connected SQI flash

Message	**UBX-UPD-CERASE** **Chip erase the connected SQI flash**					
Type	Command					
Comment	An UBX-ACK-ACK message is sent if the command was received and is valid. A UBX-ACK-NAK is sent if the payload has a wrong size. This response indicates the success state of the command parsing, but does not give an indication whether the command was added to the work queue or had to be discarded because of queue size or memory limitation. If the command could not be added to the queue, the receiver will not provide an additional answer to the UBX-ACK-ACK message. If the command was added to the queue and after flash erase operation was performed, the erase success state is signalized with an UBX-UPD-CERASE output message. Note that depending on the flash it can take up to 5 minutes until the UBX-UPD-CERASE message is output.					
Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x09	0x16	0	*see below*	CK_A CK_B
Payload	This message has no payload.					

3.19.1.2 Chip erase the connected SQI flash acknowledgement

Message	**UBX-UPD-CERASE** **Chip erase the connected SQI flash acknowledgement**					
Type	Output					
Comment						
Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x09	0x16	1	*see below*	CK_A CK_B

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	U1	success	-	-	1 if success, 0 if chip erase failed

3.19.2 UBX-UPD-CRC (0x09 0x0d)

3.19.2.1 Check CRC of binary

Message	**UBX-UPD-CRC** **Check CRC of binary**
Type	Command

Comment	A UBX-ACK-ACK is sent after after the answer (UBX-UPD-CRC) was sent. A NAK is sent if the payload has a wrong size. ☞ Check of the firmware stored in RAM is performed against CRC checksum received in message					
Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x09	0x0d	18	*see below*	CK_A CK_B

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	U1	`version`	-	-	Always 0x01
1	U1	`region`	-	-	0 - Calculate CRC over the data stored in any memory mapped region 1 - Calculate CRC over the data stored in flash
2	X4	`addr`	-	-	CRC range begin address (has to be 4 byte aligned)
6	U4	`size`	-	-	CRC range size (has to be a multiple of 4)
10	X4	`crcA`	-	-	First word of CRC checksum
14	X4	`crcB`	-	-	Second word of CRC checksum

3.19.3 UBX-UPD-ERASE (0x09 0x0b)

3.19.3.1 Erase flash sector

Message	**UBX-UPD-ERASE** **Erase flash sector**					
Type	Command					
Comment	A UBX-ACK-ACK message is sent if the command was received and is valid. A UBX-ACK-NAK is sent if the payload has a wrong size. This response indicates the success state of the command parsing, but does not give an indication whether the command was added to the work queue or had to be discarded because of queue size or memory limitation. If the command could not be added to the queue, the receiver will not provide an additional answer to the UBX-ACK-ACK message. If the command was added to the queue and after flash erase operation was performed, the erase success state is signalized with an UBX-UPD-ERASE output message. Note that depending on the flash it can take up to 10 seconds until the UBX-UPD-ERASE message is output.					
Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x09	0x0b	4	*see below*	CK_A CK_B

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	U4	`addr`	-	-	Base address of flash sector

3.19.3.2 Erase flash sector acknowledgement

Message	**UBX-UPD-ERASE** **Erase flash sector acknowledgement**					
Type	Output					
Comment						
Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x09	0x0b	5	*see below*	CK_A CK_B

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	X4	`addr`	-	-	Base address of flash sector

4	U1	success	-	-	1 if success, 0 if erase failed

3.19.4 UBX-UPD-FLDET (0x09 0x08)

3.19.4.1 Get the flash manufacturer and device IDs

Message	**UBX-UPD-FLDET** **Get the flash manufacturer and device IDs**					
Type	Poll request					
Comment	A UBX-ACK-ACK is sent after the answer (UBX-UPD-FLDET with payload) was sent. A UBX-ACK-NAK is sent if the payload has a wrong size.					
Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x09	0x08	4	*see below*	CK_A CK_B

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	U4	address	-	-	Base address of Flash

3.19.4.2 Flash manufacturer and device IDs

Message	**UBX-UPD-FLDET** **Flash manufacturer and device IDs**					
Type	Get					
Comment	This is the response from the receiver					
Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x09	0x08	8	*see below*	CK_A CK_B

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	U4	address	-	-	Base address of Flash
4	U2	manId	-	-	Manufacturer ID
6	U2	devId	-	-	Device ID

3.19.5 UBX-UPD-FLWRI (0x09 0x0c)

3.19.5.1 Write flash data (area must be erased before)

Message	**UBX-UPD-FLWRI** **Write flash data (area must be erased before)**					
Type	Command					
Comment	A UBX-ACK-ACK message is sent if the command was received and it is valid. A UBX-ACK-NAK is sent if the payload has a wrong size or the field 'size' does not match the data payload size. This response indicates the success state of the command parsing, but it does not indicate whether the command was added to the work queue or had to be discarded because of the queue size or memory limitation. If the command cannot be added to the queue, the receiver does not provide an additional answer to the UBX-ACK-ACK message. If the command is added to the queue, the write success state is indicated with the UBX-UPD-FLWRI output message after the flash write operation has been performed.					
Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x09	0x0c	8 + size	*see below*	CK_A CK_B

Payload description:

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

0	X4	addr	-	-	Base address of the write block. Must be within the range of 0..71 .
4	U4	size	-	-	Size of data to write. Must be within the range of 1..72-addr.
Start of repeated group (size times)					
8 + n	U1	data	-	-	Data to write
End of repeated group (size times)					

3.19.5.2 Write flash data success indication

Message	**UBX-UPD-FLWRI** **Write flash data success indication**					
Type	Output					
Comment	Success report for write command					
Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x09	0x0c	5	*see below*	CK_A CK_B

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	X4	addr	-	-	Base address of write block
4	U1	success	-	-	1 if success, 0 if write failed

3.19.6 UBX-UPD-IDEN (0x09 0x06)

3.19.6.1 Identify flash loader version

Message	**UBX-UPD-IDEN** **Identify flash loader version**					
Type	Poll request					
Comment	A UBX-ACK-ACK is sent after the answer (UBX-UPD-IDEN including payload) was sent. A UBX-ACK-NAK is sent if the payload has a wrong size.					
Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x09	0x06	0	*see below*	CK_A CK_B
Payload	This message has no payload.					

3.19.6.2 Flash loader version

Message	**UBX-UPD-IDEN** **Flash loader version**					
Type	Get					
Comment	This is the version response from the receiver.					
Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x09	0x06	1	*see below*	CK_A CK_B

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	X1	Version	-	-	Major.Minor (each 4 Bit)

3.19.7 UBX-UPD-POS (0x09 0x15)

3.19.7.1 Enable PLL during safeboot

Message	**UBX-UPD-POS** **Enable PLL during safeboot**					
Type	Command					
Comment	The host can send this message in order to enable precise clock. Clock configuration needs to be loaded beforehand using the UBX-CFG-VALSET message. A UBX-ACK-NAK is sent if the message has wrong size or enabling PLL failed. Host should wait for UBX-ACK-ACK, which is issued after external oscillator and PLL are started. It might take a few hundreds of ms before it is completed in case of XTO auto tuning and few dozens of milliseconds in other cases.					
Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x09	0x15	2	*see below*	CK_A CK_B

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	U1	`version`	-	-	Message version (1 for this message)
1	U1	`skipOsc`	-	-	If true OSC initialization is skipped (it has to be already running!)

3.19.8 UBX-UPD-RBOOT (0x09 0x0e)

3.19.8.1 Perform a watchdog reset

Message	**UBX-UPD-RBOOT** **Perform a watchdog reset**					
Type	Command					
Comment	Performs a watchdog reset after disconnecting USB (if connected). The type of the reset can be compared to a hot start with an additional operating system reboot. This message is not acknowledged as the system is being reset immediately.					
Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x09	0x0e	0	*see below*	CK_A CK_B
Payload	This message has no payload.					

3.19.9 UBX-UPD-ROM (0x09 0x25)

3.19.9.1 ROM CRC

Message	**UBX-UPD-ROM** **ROM CRC**					
Type	Polled					
Comment						
Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x09	0x25	12	*see below*	CK_A CK_B

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	U1	`version`	-	-	Version (0x01 for this version)
1	U1	`reserved0`	-	-	Reserved
2	U1[2]	`reserved1`	-	-	Reserved
4	U4	`romCrcLsw`	-	-	Least significant word of ROM CRC (ROM size - 8 bytes)

8	U4	romCrcMsw	-	-	Most significant word of ROM CRC (ROM size - 4 bytes)

3.19.10 UBX-UPD-SAFE (0x09 0x07)

3.19.10.1 Boot in safe environment from ROM or RAM

Message	**UBX-UPD-SAFE** **Boot in safe environment from ROM or RAM**					
Type	Command					
Comment	Boot receiver in a safe environment from ROM or RAM. A UBX-ACK-ACK is sent after receiving the command. A UBX-ACK-NAK is sent if the payload has a wrong size.					
Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x09	0x07	0	*see below*	CK_A CK_B
Payload	This message has no payload.					

3.19.10.2 Start flash loader task

Message	**UBX-UPD-SAFE** **Start flash loader task**					
Type	Command					
Comment	If already running the firmware from ROM, the flash loader task has to be started prior to sending update messages (especially flash-write and erase). The receiver does not need to be started in safe environment. A UBX-ACK-ACK is sent after receiving the command. A UBX-ACK-NAK is sent if the payload has a wrong size.					
Message structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xb5 0x62	0x09	0x07	1	*see below*	CK_A CK_B

Payload description:

Byte offset	*Type*	*Name*	*Scale*	*Unit*	*Description*
0	X1	flags	-	-	flags
bit 0	$U_{:1}$	ldrStart	-	-	start flash loader task (flash write and erase routines)

3.19.11 UBX-UPD-SOS (0x09 0x14)

3.19.11.1 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.					

3.19.11.2 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.

	Header	Class	ID	Length (Bytes)	Payload	Checksum
Message structure	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

3.19.11.3 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.

	Header	Class	ID	Length (Bytes)	Payload	Checksum
Message structure	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

3.19.11.4 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.

	Header	Class	ID	Length (Bytes)	Payload	Checksum
Message structure	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

3.19.11.5 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 system. The host should clear the backup file after receiving this message. If the UBX-UPD-SOS message is polled, this message is 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 10403.4 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 3.4 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) (Input/output)

4.4 RTCM 3.4 messages

For details see RTCM protocol and the RTCM Standard 10403.4 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 10403.4 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 10403.4 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 10403.4 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 10403.4 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 10403.4 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 10403.4 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 10403.4 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 10403.4 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 10403.4 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 10403.4 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 10403.4 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 10403.4 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 10403.4 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 10403.4 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 10403.4 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 10403.4 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 10403.4 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 10403.4 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 10403.4 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 10403.4 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 10403.4 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 10403.4 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 10403.4 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 10403.4 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 10403.4 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	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 (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 2.0.2, February 2022*.

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-OCB_BDS	0xf6 0x04	Message type 0, sub-type 3 • BeiDou 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-HPAC_BDS	0xf6 0x0d	Message type 1, sub-type 3 • BeiDou 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 2.0.2, February 2022 available from https://www.spartnformat.org.

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 2.0.2, February 2022 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 2.0.2, February 2022 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 2.0.2, February 2022 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 0, sub-type 3

5.4.4.1 BeiDou orbit, clock, bias (OCB)

Message	**SPARTN-1X-OCB_BDS** **BeiDou orbit, clock, bias (OCB)**
Type	Input
Comment	This message carries the data for BeiDou 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 2.0.2, February 2022 for a detailed message specification.
Information	*Class/ID:* 0xf6 0x04, *Message Type:* 0 (0x00), *Sub-type:* 3 (0x3), *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 0

5.4.5.1 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 2.0.2, February 2022 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)					

5.4.6 Message type 1, sub-type 1

5.4.6.1 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 2.0.2, February 2022 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)					

5.4.7 Message type 1, sub-type 2

5.4.7.1 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 2.0.2, February 2022 for a detailed message specification.				
Information	*Class/ID:* 0xf6 0x0c, *Message Type:* 1 (0x01), *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)					

5.4.8 Message type 1, sub-type 3

5.4.8.1 BeiDou high-precision atmosphere correction (HPAC)

Message	**SPARTN-1X-HPAC_BDS** **BeiDou high-precision atmosphere correction (HPAC)**
Type	Input
Comment	This message contains high-precision atmosphere data for BeiDou, 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 2.0.2, February 2022 for a detailed message specification.
Information	*Class/ID:* 0xf6 0x0d, *Message Type:* 1 (0x01), *Sub-type:* 3 (0x3), *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)					

5.4.9 Message type 2, sub-type 0

5.4.9.1 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 2.0.2, February 2022 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

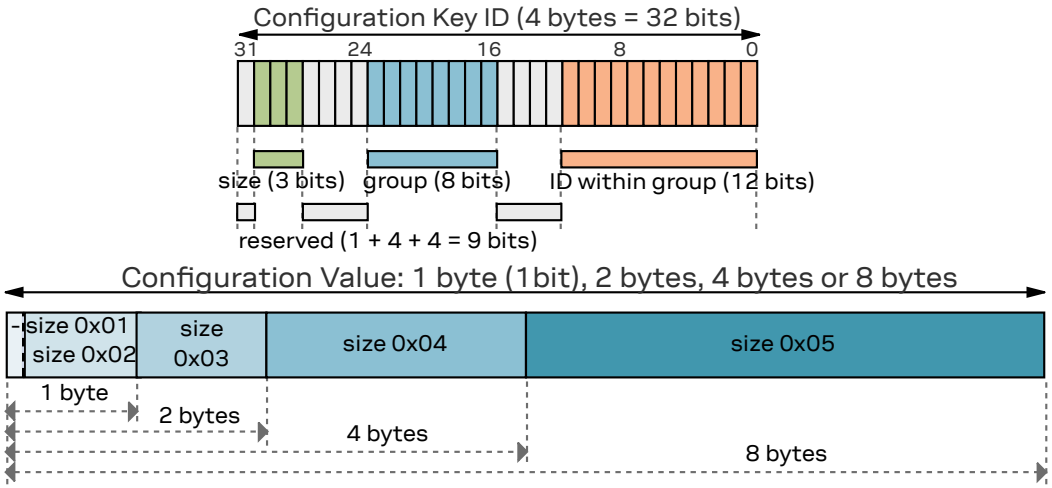
This chapter describes the receiver configuration interface.

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

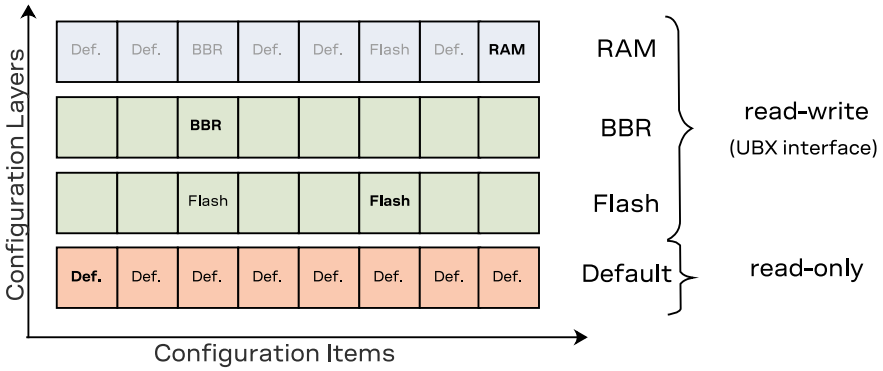
6.3 Configuration layers

The receiver has several *Configuration Layers*. 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 replace values stored in a low-priority layer. At startup, the receiver reads all configuration layers and stacks up the items 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 is 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 becomes effective when the receiver is restarted.
- **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 becomes effective when the receiver is restarted.
- **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.



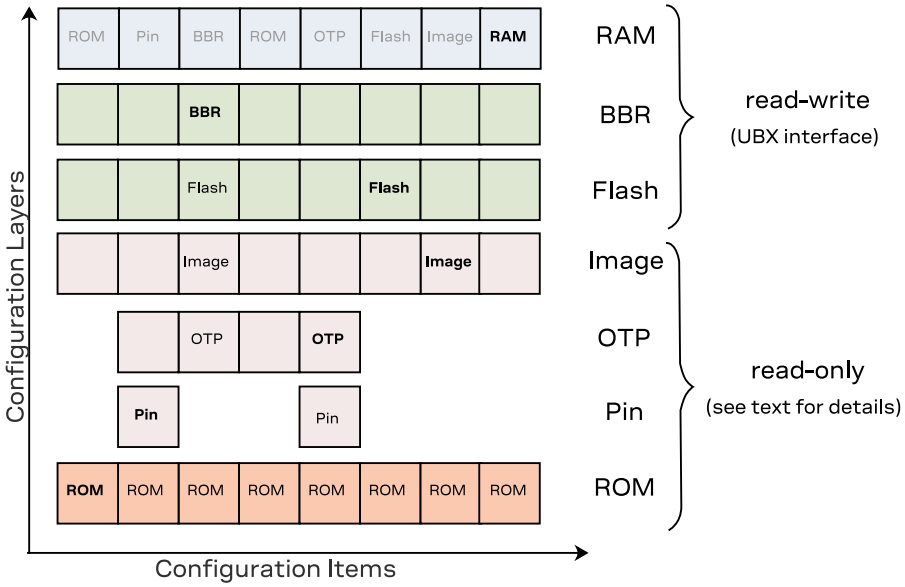
In the example figure above several items (e.g. the first item) are only set in the Default Layer and hence the default value ends up in Current Configuration in the RAM Layer. The third item is present in the Default, Flash and BBR Layers. The value from the BBR Layer has the highest priority and therefore 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.

6.3.1 Default layer composite

The Default Layer is a composite of the following four layers. Some of these are writable by special means different from the UBX protocol interface used for the read-write layers listed above.

- **Image:** This layer contains items appended to an external flash firmware image. It is not modifiable by the user. It is used to provide firmware images that differ in their default (factory) configuration but not in their software or the items in the ROM layer.
- **OTP:** This layer contains items from the contents of the OTP memory (one-time programmable memory). See OTP layer configuration for details.
- **Pin:** This layer contains items derived from configuration pins. See Pin layer configuration for details.
- **ROM:** This layer defines all items known to the running receiver software and their hard-coded default value. Data in this layer is not writeable.

The figure below shows all seven layers. An empty space indicates that the item cannot be stored in that layer.



In the example figure above, the first and fourth items are only present in the ROM Layer. Hence the value from the ROM Layer ends up in the RAM Layer. The second item is also present in the Pin Layer and hence that value ends up in the RAM Layer. The third item is present in the ROM, OTP, Image, Flash and BBR Layers. Since the BBR Layer has the highest priority, this value will end up in the RAM Layer. The seventh item is present in the ROM and Image Layers. There is no corresponding item in the Flash or BBR Layers and so the value from the Image Layer ends up in the RAM Layer. The last item is present in the ROM and the RAM Layers. Upon startup the value in the RAM Layer was the value from the ROM Layer. But here 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

6.4.2 Pin layer configuration

Some configuration items are available in the Pin Layer. See section Configuration pins in the integration manual for or details on how to use configuration pins and how their state affects the values of these items in the Pin Layer.

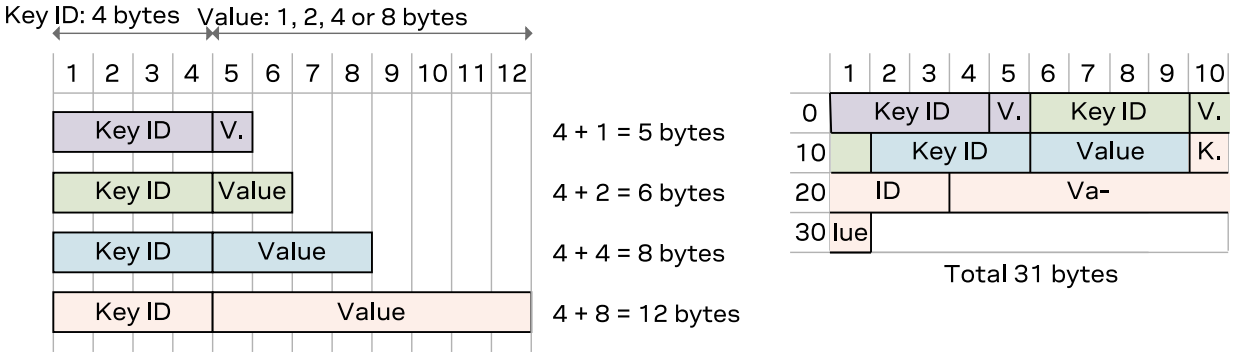
6.4.3 OTP layer configuration

Some configuration items are available in the OTP Layer. They can be set or changed by changing the values in the OTP memory. See section OTP layer configuration in the integration manual for details.

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.



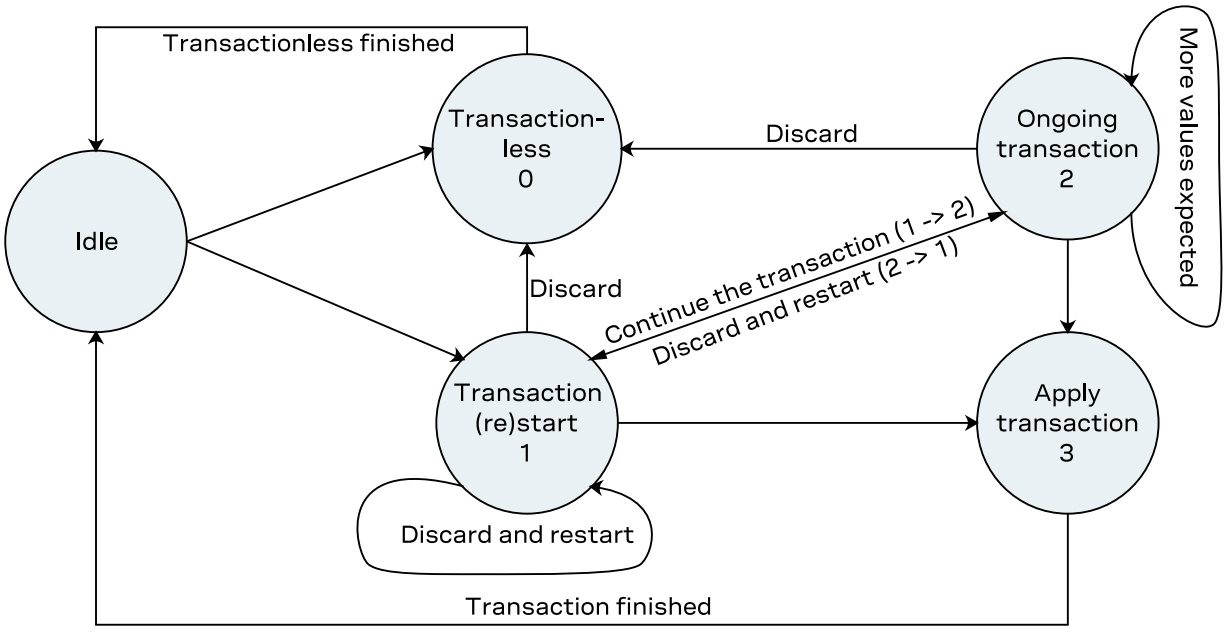
Note that this is an arbitrary example and any number of items of any value storage size can be concatenated the same way.

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, specify the layer(s) to apply the changes to. This list of configuration layer(s) must be observed throughout the transaction states. Modifying the configuration layer(s) mid-transaction causes the transaction to be aborted and consequently, no queued changes will be applied.

In the start transaction state, the receiver locks 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 are queued waiting to be applied.

In the ongoing state, a configuration key and value must be sent. The receiver aborts the transaction and does not apply any changes if this condition is violated. Key-value pairs sent in the ongoing state are queued waiting to be applied.

In the apply state, the receiver collectively checkes the queued changes and applied them to the requested configuration layer(s). Note that any additional key-value pairs sent within the apply state are 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 aborts the current transaction and the queued changes are not 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-CLOCK	System clock 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-LOGFILTER	Data logger configuration
CFG-MOT	Motion detector configuration
CFG-MSGOUT	Message 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-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
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	Time pulse 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 5: CFG-BDS configuration items

6.9.2 CFG-CLOCK: System clock configuration

Configuration of system clock tree.

Configuration item	Key ID	Type	Scale	Unit	Description
CFG-CLOCK-OSC_FREQ	0x40a4000d	U4	-	Hz	Oscillator speed

Table 6: CFG-CLOCK configuration items

6.9.3 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 8 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 9 below for a list of possible constants for this item.					
CFG-GEOFENCE-PIN	0x20240014	U1	-	-	PIO pin number
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

Configuration item	Key ID	Type	Scale	Unit	Description
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 7: 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 8: 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 9: Constants for CFG-GEOFENCE-PINPOL

6.9.4 CFG-HW: Hardware configuration

Hardware configuration settings.

Note that not all settings are available for all products. See the applicable data sheet for supported features.

Configuration item	Key ID	Type	Scale	Unit	Description
CFG-HW-SINGLE_CLK	0x10a30019	L	-	-	Single-clock system
If set to true, the main oscillator is used to maintain time in backup or standby mode. A false value indicates that a dedicated RTC crystal is used if present.					
CFG-HW-BYPASS_LDO_DIS	0x10a30020	L	-	-	Bypass LDO_C disable
Disable the LDO_C bypass. When this setting is true the LDO_C is in use. If this setting is false (LDO_C is bypassed), V_CORE must be supplied at 1 V. Set to true if V_CORE = 1.8 V (nominal) and false if V_CORE = 1 V (nominal).					
CFG-HW-CLK_OFFSET	0x40a30028	I4	-	ppb	Clock offset
CFG-HW-CLK_OFFSET_VALID	0x10a30029	L	-	-	Clock offset valid
CFG-HW-CLK_PRECISION	0x40a3002a	U4	-	ppb	Precision of the clock offset
CFG-HW-CLK_MAX_CALIB_DEV	0x40a3002b	U4	-	ppb	Maximum calibration deviation
CFG-HW-CLK_MAX_CALIB_DEV_VALID	0x10a3002c	L	-	-	Max calibration deviation valid
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.					

Configuration item	Key ID	Type	Scale	Unit	Description
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.					
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	-	-	Antenna switch PIO number
Antenna switch PIO number. Used by EXT and MADC engines.					
CFG-HW-ANT_SUP_SHORT_PIN	0x20a30037	U1	-	-	Antenna short detection PIO number
Antenna short detection PIO number. Used by EXT engine.					
CFG-HW-ANT_SUP_OPEN_PIN	0x20a30038	U1	-	-	Antenna open detection PIO number
Antenna open detection PIO number. Used by EXT engine.					
CFG-HW-ANT_ON_SHORT_US	0x30a3003c	U2	-	-	ANT on->short timeout[us]
Delay in microseconds between turning the antenna power supply on and enabling the antenna short circuit detection.					
CFG-HW-CLK_IS_TCXO	0x10a30047	L	-	-	Oscillator type indicator
True if clock is a TCXO, false otherwise.					
CFG-HW-OSC_VOLTAGE	0x40a30052	U4	-	mV	Oscillator voltage indicator
CFG-HW-OSC_CURRENT	0x40a30053	U4	-	uA	Oscillator current indicator
CFG-HW-ANT_SUP_ENGINE	0x20a30054	E1	-	-	Antenna supervisor engine selection
Select the engine used to evaluate antenna state. The EXT engine uses an external comparator for current measurement. The MADC engine uses built-in measurement ADC and requires only a shunt resistor for current measurement. The MADC engine is supported only in selected u-blox generation 9 receivers. See Table 11 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.					
CFG-HW-TCXO_DC_BIAS_ENABLE	0x10a3005e	L	-	-	Enable DC bias for TCXO
Set to true if the TCXO is connected through a DC block (i.e. is AC coupled), set to false if TCXO is DC coupled.					
CFG-HW-FLASH_REFRESH	0x10a30068	L	-	-	Enable flash refresh
Set true to periodically rewrite the flash contents to increase the flash retention time.					

Table 10: CFG-HW configuration items

Constant	Value	Description
EXT	0	Use the EXT engine.

Constant	Value	Description
MADC	1	Use the MADC engine.

Table 11: Constants for CFG-HW-ANT_SUP_ENGINE

6.9.5 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 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
CFG-I2C-REMAP	0x10510004	L	-	-	I2C remapping

Table 12: CFG-I2C configuration items

6.9.6 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
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 13: CFG-I2CINPROT configuration items

6.9.7 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 14: CFG-I2COUTPROT configuration items

6.9.8 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 16 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 16 below for a list of possible constants for this item.					

Configuration item	Key ID	Type	Scale	Unit	Description
CFG-INFMSG-UBX_UART2	0x20920003	X1	-	-	Information message enable flags for the UBX protocol on the UART2 interface
See Table 16 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 16 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 16 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 16 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 16 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 16 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 16 below for a list of possible constants for this item.					
CFG-INFMSG-NMEA_SPI	0x2092000a	X1	-	-	Information message enable flags for the NMEA protocol on the SPI interface
See Table 16 below for a list of possible constants for this item.					

Table 15: 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 16: 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

6.9.9 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-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 17: 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	Static hold speed threshold, below which the receiver is considered to be stationary
Set this parameter to 0 to enable the default firmware value or behavior.					
CFG-MOT-GNSSDIST_THRS	0x3025003b	U2	1.0	m	Static hold distance threshold, within which the receiver is considered to be stationary
Set this parameter to 0 to enable the default firmware value or behavior.					

Table 18: 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

Configuration item	Key ID	Type	Scale	Unit	Description
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
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

Configuration item	Key ID	Type	Scale	Unit	Description
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
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

Configuration item	Key ID	Type	Scale	Unit	Description
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-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

Configuration item	Key ID	Type	Scale	Unit	Description
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
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

Configuration item	Key ID	Type	Scale	Unit	Description
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
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

Configuration item	Key ID	Type	Scale	Unit	Description
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
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_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

Configuration item	Key ID	Type	Scale	Unit	Description
CFG-MSGOUT-UBX_MON_PT2_I2C	0x20910209	U1	-	-	Output rate of the UBX-MON-PT2 message on port I2C
CFG-MSGOUT-UBX_MON_PT2_SPI	0x2091020d	U1	-	-	Output rate of the UBX-MON-PT2 message on port SPI
CFG-MSGOUT-UBX_MON_PT2_UART1	0x2091020a	U1	-	-	Output rate of the UBX-MON-PT2 message on port UART1
CFG-MSGOUT-UBX_MON_PT2_UART2	0x2091020b	U1	-	-	Output rate of the UBX-MON-PT2 message on port UART2
CFG-MSGOUT-UBX_MON_PT2_USB	0x2091020c	U1	-	-	Output rate of the UBX-MON-PT2 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_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
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_NAV_CLOCK_I2C	0x20910065	U1	-	-	Output rate of the UBX-NAV-CLOCK message on port I2C

Configuration item	Key ID	Type	Scale	Unit	Description
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
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

Configuration item	Key ID	Type	Scale	Unit	Description
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
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_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
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

Configuration item	Key ID	Type	Scale	Unit	Description
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
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

Configuration item	Key ID	Type	Scale	Unit	Description
CFG-MSGOUT-UBX_NAV_SIG_USB	0x20910348	U1	-	-	Output rate of the UBX-NAV-SIG 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
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

Configuration item	Key ID	Type	Scale	Unit	Description
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
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

Configuration item	Key ID	Type	Scale	Unit	Description
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
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

Configuration item	Key ID	Type	Scale	Unit	Description
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_SEC_SIGLOG_I2C	0x20910689	U1	-	-	Output rate of the UBX-SEC-SIGLOG message on port I2C
CFG-MSGOUT-UBX_SEC_SIGLOG_SPI	0x2091068d	U1	-	-	Output rate of the UBX-SEC-SIGLOG message on port SPI
CFG-MSGOUT-UBX_SEC_SIGLOG_UART1	0x2091068a	U1	-	-	Output rate of the UBX-SEC-SIGLOG message on port UART1
CFG-MSGOUT-UBX_SEC_SIGLOG_UART2	0x2091068b	U1	-	-	Output rate of the UBX-SEC-SIGLOG message on port UART2
CFG-MSGOUT-UBX_SEC_SIGLOG_USB	0x2091068c	U1	-	-	Output rate of the UBX-SEC-SIGLOG message on port USB
CFG-MSGOUT-UBX_SEC_SIG_I2C	0x20910634	U1	-	-	Output rate of the UBX-SEC-SIG message on port I2C
CFG-MSGOUT-UBX_SEC_SIG_SPI	0x20910638	U1	-	-	Output rate of the UBX-SEC-SIG message on port SPI
CFG-MSGOUT-UBX_SEC_SIG_UART1	0x20910635	U1	-	-	Output rate of the UBX-SEC-SIG message on port UART1
CFG-MSGOUT-UBX_SEC_SIG_UART2	0x20910636	U1	-	-	Output rate of the UBX-SEC-SIG message on port UART2
CFG-MSGOUT-UBX_SEC_SIG_USB	0x20910637	U1	-	-	Output rate of the UBX-SEC-SIG 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

Configuration item	Key ID	Type	Scale	Unit	Description
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
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 19: CFG-MSGOUT configuration items

6.9.12 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 21 below for a list of possible constants for this item.					

Table 20: CFG-NAVHPG configuration items

Constant	Value	Description
RTK_FLOAT	2	No attempts made to fix ambiguities
RTK_FIXED	3	Ambiguities are fixed whenever possible
RTK_CAR	5	Conservative ambiguity resolution

Table 21: Constants for CFG-NAVHPG-DGNSSMODE

6.9.13 CFG-NAVSPG: Standard precision navigation configuration

This group contains configuration items related to the operation of the receiver at standard precision, including configuring position 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 23 below for a list of possible constants for this item.					
CFG-NAVSPG-INIFIX3D	0x10110013	L	-	-	Initial fix must be a 3D fix

Configuration item	Key ID	Type	Scale	Unit	Description
CFG-NAVSPG-WKNROLLOVER	0x30110017	U2	-	-	GPS week rollover number
GPS week numbers are set correctly from this week up to 1024 weeks after this week. The 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 24 below for a list of possible constants for this item.					
CFG-NAVSPG-DYNMODEL	0x20110021	E1	-	-	Dynamic platform model
See Table 25 below for a list of possible constants for this item.					
CFG-NAVSPG-DCMMODE	0x20110023	E1	-	-	Delta carrier measurements mode
See Table 26 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
User specified geodetic system can be used instead of the receiver's default WGS84 ellipsoid. All of the CFG-NAVSPG-USERDAT_* user geodetic datum parameters listed here must be configured before enabling the user specified geodetic system.					
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					
CFG-NAVSPG-USRDAT_FLAT	0x50110063	R8	-	-	Geodetic datum 1.0 / flattening
Accepted range is 0.0 to 500.0.					
CFG-NAVSPG-USRDAT_DX	0x40110064	R4	-	m	Geodetic datum X axis shift at the origin
Accepted range is +/- 5000.0 meters.					
CFG-NAVSPG-USRDAT_DY	0x40110065	R4	-	m	Geodetic datum Y axis shift at the origin
Accepted range is +/- 5000.0 meters.					
CFG-NAVSPG-USRDAT_DZ	0x40110066	R4	-	m	Geodetic datum Z axis shift at the origin
Accepted range is +/- 5000.0 meters.					
CFG-NAVSPG-USRDAT_ROTX	0x40110067	R4	-	arcsec	Geodetic datum rotation about the X axis
Accepted range is +/- 20.0 milli arc seconds.					
CFG-NAVSPG-USRDAT_ROTY	0x40110068	R4	-	arcsec	Geodetic datum rotation about the Y axis ()
Accepted range is +/- 20.0 milli-arc seconds.					
CFG-NAVSPG-USRDAT_ROTZ	0x40110069	R4	-	arcsec	Geodetic datum rotation about the Z axis
Accepted range is +/- 20.0 milli-arc seconds.					
CFG-NAVSPG-USRDAT_SCALE	0x4011006a	R4	-	ppm	Geodetic datum scale factor
Accepted range is 0.0 to 50.0 parts per million.					
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)

Configuration item	Key ID	Type	Scale	Unit	Description
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

Table 22: CFG-NAVSPG configuration items

Constant	Value	Description
2DONLY	1	2D only
3DONLY	2	3D only
AUTO	3	Auto 2D/3D

Table 23: 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
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
NICT	9	UTC as operated by the National Institute of Information and Communications Technology, Japan (NICT); derived from QZSS time

Table 24: 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 25: Constants for CFG-NAVSPG-DYNMODEL

Constant	Value	Description
DEFAULT	0	Use the default delta carrier measurement mode defined by the dynamic model
DONTUSE	1	Don't use delta carrier measurements
EPOCH	2	One-sided full epoch delta carrier measurements
1S100	3	One-sided 100 ms delta carrier measurements
1S200	4	One-sided 200 ms delta carrier measurements
1S400	5	One-sided 400 ms delta carrier measurements
2S100	6	Two-sided 100 ms delta carrier measurements
2S200	7	Two-sided 200 ms delta carrier measurements
2S400	8	Two-sided 400 ms delta carrier measurements

Table 26: Constants for CFG-NAVSPG-DCMMODE

6.9.14 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 28 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 29 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 affects the way the used satellite count in NMEA output is calculated. If set, also considered but rejected 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
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 30 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-FILT_NAVIC	0x10930018	L	-	-	Disable reporting of NavIC satellites
CFG-NMEA-OUT_INVFIX	0x10930021	L	-	-	Enable position output for failed or invalid fixes

Configuration item	Key ID	Type	Scale	Unit	Description
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 31 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 32 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 receiver uses the default BeiDou Talker ID.					

Table 27: 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 28: Constants for CFG-NMEA-PROTVER

Constant	Value	Description
UNLIM	0	Unlimited
8SVS	8	8 SVs
12SVS	12	12 SVs
16SVS	16	16 SVs

Table 29: Constants for CFG-NMEA-MAXSVS

Constant	Value	Description
STRICT	0	Strict - satellites are not output
EXTENDED	1	Extended - use proprietary numbering

Table 30: 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)

Constant	Value	Description
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 31: 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 32: Constants for CFG-NMEA-GSVTALKERID

6.9.15 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 34 below for a list of possible constants for this item.
CFG-ODO-COGMAXSPEED	0x20220021	U1	1e-1	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 33: CFG-ODO configuration items

Constant	Value	Description
RUN	0	Running
CYCL	1	Cycling
SWIM	2	Swimming
CAR	3	Car
CUSTOM	4	Custom

Table 34: Constants for CFG-ODO-PROFILE

6.9.16 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

Configuration item	Key ID	Type	Scale	Unit	Description
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 35: CFG-QZSS configuration items

6.9.17 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 is 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.					
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 37 below for a list of possible constants for this item.					

Table 36: 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
NAVIC	5	Align measurements to NavIC time

Table 37: Constants for CFG-RATE-TIMEREF

6.9.18 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.					

Configuration item	Key ID	Type	Scale	Unit	Description
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 39 below for a list of possible constants for this item.					

Table 38: 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
STRICT	2	Strict RTCM input filter; input messages allowed must contain a DF003 data field matching the CFG-RTCM-DF003 value

Table 39: Constants for CFG-RTCM-DF003_IN_FILTER

6.9.19 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 uses only GPS satellites for which integrity information is available					
CFG-SBAS-ACCEPT_NOT_IN_PRNMASK	0x30360008	X2	-	-	Accept corrections from SBAS SV, even if not self included in PRN MASK (Message Type 1)
If enabled, the receiver will still use the SBAS data, even when the SBAS SV itself is not included in its PRN MASK. This is only useful for BDSBAS and not compatible whith current EGNOS implementation. See Table 41 below for a list of possible constants for this item.					
CFG-SBAS-USE_IONOONLY	0x10360007	L	-	-	Use SBAS ionosphere correction only
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 42 below for a list of possible constants for this item.					

Table 40: CFG-SBAS configuration items

Constant	Value	Description
WAAS	0x01	WAAS bit
1 = Use WAAS provider Id.		
EGNOS	0x02	EGNOS bit
1 = Use EGNOS provider Id.		
MSAS	0x04	MSAS bit
1 = Use MSAS provider Id.		

Constant	Value	Description
GAGAN	0x08	GAGAN bit
1 = Use GAGAN provider Id.		
SDCM	0x10	SDCM bit
1 = Use SDCM provider Id.		
BDSBAS	0x20	BDSBAS bit
1 = Use BDSBAS provider Id.		
KASS	0x40	KASS bit
1 = Use KASS provider Id.		

Table 41: Constants for CFG-SBAS-ACCEPT_NOT_IN_PRNMASK

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
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

Constant	Value	Description
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 42: Constants for CFG-SBAS-PRNSCANMASK

6.9.20 CFG-SEC: Security configuration

Security configuration.

Configuration item	Key ID	Type	Scale	Unit	Description
CFG-SEC-CFG_LOCK	0x10f60009	L	-	-	Configuration lockdown
When set, the 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 enables writing to the specified group even 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 enables writing to the specified group even after the configuration lockdown has been enabled.					
CFG-SEC-SPOOFDET_SIM_SIG_DIS	0x10f6005d	L	-	-	Disabling the simulated signal spoofing detection.
CFG-SEC-JAMDET_SENSITIVITY_HI	0x10f60051	L	-	-	When set, go for a more sensitive jamming detection (at the cost of increased false alarm rate).

Table 43: CFG-SEC configuration items

6.9.21 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 triggers a reset to the GNSS subsystem. The reset takes some time, so wait first for the acknowledgement from the receiver and then 0.5 seconds before sending the next command.

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-GPS_L5_ENA	0x10310004	L	-	-	GPS L5
CFG-SIGNAL-SBAS_ENA	0x10310020	L	-	-	SBAS enable
CFG-SIGNAL-SBAS_L1CA_ENA	0x10310005	L	-	-	SBAS L1C/A

Configuration item	Key ID	Type	Scale	Unit	Description
CFG-SIGNAL-GAL_ENA	0x10310021	L	-	-	Galileo enable
CFG-SIGNAL-GAL_E1_ENA	0x10310007	L	-	-	Galileo E1
CFG-SIGNAL-GAL_E5A_ENA	0x10310009	L	-	-	Galileo E5a
CFG-SIGNAL-GAL_E5B_ENA	0x1031000a	L	-	-	Galileo E5b
CFG-SIGNAL-GAL_E6_ENA	0x1031000b	L	-	-	Galileo E6
CFG-SIGNAL-BDS_ENA	0x10310022	L	-	-	BeiDou Enable
CFG-SIGNAL-BDS_B1_ENA	0x1031000d	L	-	-	BeiDou B1I
CFG-SIGNAL-BDS_B1C_ENA	0x1031000f	L	-	-	BeiDou B1C
CFG-SIGNAL-BDS_B2_ENA	0x1031000e	L	-	-	BeiDou B2I
CFG-SIGNAL-BDS_B2A_ENA	0x10310028	L	-	-	BeiDou B2a
CFG-SIGNAL-BDS_B3_ENA	0x10310010	L	-	-	BeiDou B3I
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-QZSS_L5_ENA	0x10310017	L	-	-	QZSS L5
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
CFG-SIGNAL-NAVIC_ENA	0x10310026	L	-	-	NavIC enable
CFG-SIGNAL-NAVIC_L5_ENA	0x1031001d	L	-	-	NavIC L5

Table 44: CFG-SIGNAL configuration items

6.9.22 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 46 below for a list of possible constants for this item.					

Table 45: 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 46: Constants for CFG-SPARTN-USE_SOURCE

6.9.23 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

Configuration item	Key ID	Type	Scale	Unit	Description
CFG-SPI-CPOLARITY	0x10640002	L	-	-	Clock polarity select: 0: Active Hight Clock, SCLK idles low, 1: Active Low Clock, SCLK idles high
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 47: CFG-SPI configuration items

6.9.24 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 48: CFG-SPIINPROT configuration items

6.9.25 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 49: CFG-SPIOUTPROT configuration items

6.9.26 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 51 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 52 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.

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_Z	0x40030005	I4	-	cm	ECEF Z coordinate of the ARP position.
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 50: 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 51: Constants for CFG-TMODE-MODE

Constant	Value	Description
ECEF	0	Position is ECEF

Constant	Value	Description
LLH	1	Position is Lat/Lon/Height

Table 52: Constants for CFG-TMODE-POS_TYPE

6.9.27 CFG-TP: Time pulse configuration

Use this group to configure the generation of time pulses.

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 54 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 55 below for a list of possible constants for this item.					
CFG-TP-ANT_CABLEDELAY	0x30050001	I2	1e-9	s	Antenna cable delay in [ns]
CFG-TP-PERIOD_TP2	0x4005000d	U4	1e-6	s	Time pulse period (TP2) in [us]
This is used only if CFG-TP-PULSE_DEF=PERIOD.					
CFG-TP-PERIOD_LOCK_TP2	0x4005000e	U4	1e-6	s	Time pulse period when locked to GNSS time (TP2) in [us]
Only used if CFG-TP-PULSE_DEF=PERIOD and CFG-TP-USE_LOCKED_TP2 is set.					
CFG-TP-FREQ_TP2	0x40050026	U4	-	Hz	Time pulse frequency (TP2)
Only used if CFG-TP-PULSE_DEF=FREQ.					
CFG-TP-FREQ_LOCK_TP2	0x40050027	U4	-	Hz	Time pulse frequency when locked to GNSS time (TP2) in [Hz]
Only used if CFG-TP-PULSE_DEF=FREQ and CFG-TP-USE_LOCKED_TP2 is set.					
CFG-TP-LEN_TP2	0x4005000f	U4	1e-6	s	Time pulse length (TP2) in [us]
Only used if CFG-TP-PULSE_LENGTH_DEF=LENGTH is set.					
CFG-TP-LEN_LOCK_TP2	0x40050010	U4	1e-6	s	Time pulse length when locked to GNSS time (TP2) in [us]
Only used if CFG-TP-PULSE_LENGTH_DEF=LENGTH and CFG-TP-USE_LOCKED_TP2 is set.					
CFG-TP-DUTY_TP2	0x5005002c	R8	-	%	Time pulse duty cycle (TP2) in [%]
Only used if CFG-TP-PULSE_LENGTH_DEF=RATIO is set.					
CFG-TP-DUTY_LOCK_TP2	0x5005002d	R8	-	%	Time pulse duty cycle when locked to GNSS time (TP2)
Only used if CFG-TP-PULSE_LENGTH_DEF=RATIO and CFG-TP-USE_LOCKED_TP2 are set.					
CFG-TP-USER_DELAY_TP2	0x40050011	I4	1e-9	s	User-configurable time pulse delay (TP2) in [ns]
CFG-TP-TP2_ENA	0x10050012	L	-	-	Enable the time pulse (TP2)
CFG-TP-SYNC_GNSS_TP2	0x10050013	L	-	-	Sync time pulse to GNSS time or local clock (TP2)
If set, sync to GNSS if GNSS time is valid. Otherwise, use local clock.					
This flag can be unset only in Timing product variants.					
CFG-TP-USE_LOCKED_TP2	0x10050014	L	-	-	Use locked parameters when possible (TP2)
If set, use CFG-TP-PERIOD_LOCK_TP2 and CFG-TP-LEN_LOCK_TP2 as soon as GNSS time is valid. Otherwise, use CFG-TP-PERIOD_TP2 and CFG-TP-LEN_TP2.					
CFG-TP-ALIGN_TO_TOW_TP2	0x10050015	L	-	-	Align time pulse to top of second (TP2)
To use this feature, CFG-TP-SYNC_GNSS_TP2 must be set.					
Time pulse period must be an integer fraction of 1 second.					
CFG-TP-POL_TP2	0x10050016	L	-	-	Set time pulse polarity (TP2)

Configuration item	Key ID	Type	Scale	Unit	Description
false (0) : falling edge at top of second. true (1) : rising edge at top of second.					
CFG-TP-TIMEGRID_TP2	0x20050017	E1	-	-	Time grid to use (TP2)
Only relevant if CFG-TP-SYNC_GNSS_TP2 is 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 attempts 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-*. No TP is generated if the selected GNSS constellation is not configured. See Table 56 below for a list of possible constants for this item.					
CFG-TP-DRSTR_TP2	0x20050036	E1	-	-	Set drive strength of TP2
Time Pulse pin 2 (TP2) can support 4 possible drive strength cases: 2, 4, 8 and 12 mA See Table 57 below for a list of possible constants for this item.					

Table 53: CFG-TP configuration items

Constant	Value	Description
PERIOD	0	Time pulse period [us]
FREQ	1	Time pulse frequency [Hz]

Table 54: Constants for CFG-TP-PULSE_DEF

Constant	Value	Description
RATIO	0	Time pulse ratio
LENGTH	1	Time pulse length

Table 55: 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 56: Constants for CFG-TP-TIMEGRID_TP2

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 57: Constants for CFG-TP-DRSTR_TP2

6.9.28 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

Configuration item	Key ID	Type	Scale	Unit	Description
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
The value is amount of 8-byte chunks. For example, value of 250 sets the trigger to 2000 bytes.					
CFG-TXREADY-INTERFACE	0x20a20005	E1	-	-	Interface where the TX ready feature should be linked to
See Table 59 below for a list of possible constants for this item.					

Table 58: CFG-TXREADY configuration items

Constant	Value	Description
I2C	0	I2C interface
SPI	1	SPI interface

Table 59: Constants for CFG-TXREADY-INTERFACE

6.9.29 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 61 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 62 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 63 below for a list of possible constants for this item.					
CFG-UART1-ENABLED	0x10520005	L	-	-	Flag to indicate if the UART1 should be enabled
CFG-UART1-REMAP	0x10520006	L	-	-	UART1 Remapping

Table 60: 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 61: Constants for CFG-UART1-STOPBITS

Constant	Value	Description
EIGHT	0	8 databits
SEVEN	1	7 databits

Table 62: 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 63: Constants for CFG-UART1-PARITY

6.9.30 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 64: CFG-UART1INPROT configuration items

6.9.31 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 65: CFG-UART1OUTPROT configuration items

6.9.32 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 67 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 68 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 69 below for a list of possible constants for this item.					
CFG-UART2-ENABLED	0x10530005	L	-	-	Flag to indicate if the UART2 should be enabled
CFG-UART2-REMAP	0x10530006	L	-	-	UART2 Remapping

Table 66: CFG-UART2 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 67: Constants for CFG-UART2-STOPBITS

Constant	Value	Description
EIGHT	0	8 databits
SEVEN	1	7 databits

Table 68: 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 69: Constants for CFG-UART2-PARITY

6.9.33 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 70: CFG-UART2INPROT configuration items

6.9.34 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 71: CFG-UART2OUTPROT configuration items

6.9.35 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

Configuration item	Key ID	Type	Scale	Unit	Description
CFG-USB-SELFPOW	0x10650002	L	-	-	Self-powered device
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 72: CFG-USB configuration items

6.9.36 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 73: CFG-USBINPROT configuration items

6.9.37 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 74: CFG-USBOUTPROT 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.

These values assume that the defaults have not been changed using the OTP layer configuration or the Pin layer configuration (see Default layer composite).

Configuration item	Key ID	Type	Scale	Unit	Default value
CFG-BDS-USE_GEO_PRN	0x10340014	L	-	-	0 (false)

Table 75: CFG-BDS configuration defaults

Configuration item	Key ID	Type	Scale	Unit	Default value
CFG-CLOCK-OSC_FREQ	0x40a4000d	U4	-	Hz	26000000

Table 76: CFG-CLOCK 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	-	-	19
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 77: CFG-GEOFENCE configuration defaults

Configuration item	Key ID	Type	Scale	Unit	Default value
CFG-HW-SINGLE_CLK	0x10a30019	L	-	-	0 (false)
CFG-HW-BYPASS_LDO_DIS	0x10a30020	L	-	-	1 (true)
CFG-HW-CLK_OFFSET	0x40a30028	I4	-	ppb	0
CFG-HW-CLK_OFFSET_VALID	0x10a30029	L	-	-	0 (false)
CFG-HW-CLK_PRECISION	0x40a3002a	U4	-	ppb	0
CFG-HW-CLK_MAX_CALIB_DEV	0x40a3002b	U4	-	ppb	0

Configuration item	Key ID	Type	Scale	Unit	Default value
CFG-HW-CLK_MAX_CALIB_DEV_VALID	0x10a3002c	L	-	-	0 (false)
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	-	-	12
CFG-HW-ANT_SUP_SHORT_PIN	0x20a30037	U1	-	-	13
CFG-HW-ANT_SUP_OPEN_PIN	0x20a30038	U1	-	-	11
CFG-HW-ANT_ON_SHORT_US	0x30a3003c	U2	-	-	500
CFG-HW-CLK_IS_TCXO	0x10a30047	L	-	-	1 (true)
CFG-HW-OSC_VOLTAGE	0x40a30052	U4	-	mV	1800
CFG-HW-OSC_CURRENT	0x40a30053	U4	-	uA	0
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
CFG-HW-TCXO_DC_BIAS_ENABLE	0x10a3005e	L	-	-	0 (false)
CFG-HW-FLASH_REFRESH	0x10a30068	L	-	-	0 (false)

Table 78: 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)
CFG-I2C-REMAP	0x10510004	L	-	-	0 (false)

Table 79: 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 80: 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 81: 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 82: CFG-INFMSG configuration defaults

Configuration item	Key ID	Type	Scale	Unit	Default value
CFG-LOGFILTER-RECORD_ENA	0x10de0002	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 83: 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	1.0	m	0

Table 84: 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

Configuration item	Key ID	Type	Scale	Unit	Default value
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
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

Configuration item	Key ID	Type	Scale	Unit	Default value
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
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-PUBX_ID_POLYP_I2C	0x209100ec	U1	-	-	0
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

Configuration item	Key ID	Type	Scale	Unit	Default value
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
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

Configuration item	Key ID	Type	Scale	Unit	Default value
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
CFG-MSGOUT-UBX_MON_COMMS_UART2	0x20910351	U1	-	-	0
CFG-MSGOUT-UBX_MON_COMMS_USB	0x20910352	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_PT2_I2C	0x20910209	U1	-	-	0
CFG-MSGOUT-UBX_MON_PT2_SPI	0x2091020d	U1	-	-	0
CFG-MSGOUT-UBX_MON_PT2_UART1	0x2091020a	U1	-	-	0
CFG-MSGOUT-UBX_MON_PT2_UART2	0x2091020b	U1	-	-	0
CFG-MSGOUT-UBX_MON_PT2_USB	0x2091020c	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_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
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

Configuration item	Key ID	Type	Scale	Unit	Default value
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_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_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
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

Configuration item	Key ID	Type	Scale	Unit	Default value
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_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
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

Configuration item	Key ID	Type	Scale	Unit	Default value
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
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

Configuration item	Key ID	Type	Scale	Unit	Default value
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_SEC_SIGLOG_I2C	0x20910689	U1	-	-	0
CFG-MSGOUT-UBX_SEC_SIGLOG_SPI	0x2091068d	U1	-	-	0
CFG-MSGOUT-UBX_SEC_SIGLOG_UART1	0x2091068a	U1	-	-	0
CFG-MSGOUT-UBX_SEC_SIGLOG_UART2	0x2091068b	U1	-	-	0
CFG-MSGOUT-UBX_SEC_SIGLOG_USB	0x2091068c	U1	-	-	0
CFG-MSGOUT-UBX_SEC_SIG_I2C	0x20910634	U1	-	-	0
CFG-MSGOUT-UBX_SEC_SIG_SPI	0x20910638	U1	-	-	0
CFG-MSGOUT-UBX_SEC_SIG_UART1	0x20910635	U1	-	-	0
CFG-MSGOUT-UBX_SEC_SIG_UART2	0x20910636	U1	-	-	0
CFG-MSGOUT-UBX_SEC_SIG_USB	0x20910637	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 85: CFG-MSGOUT configuration defaults

Configuration item	Key ID	Type	Scale	Unit	Default value
CFG-NAVHPG-DGNSSMODE	0x20140011	E1	-	-	3 (RTK_FIXED)

Table 86: CFG-NAVHPG configuration defaults

Configuration item	Key ID	Type	Scale	Unit	Default value
CFG-NAVSPG-FIXMODE	0x20110011	E1	-	-	2 (3DONLY)
CFG-NAVSPG-INIFIX3D	0x10110013	L	-	-	0 (false)
CFG-NAVSPG-WKNROLLOVER	0x30110017	U2	-	-	2326
CFG-NAVSPG-UTCSTANDARD	0x2011001c	E1	-	-	0 (AUTO)
CFG-NAVSPG-DYNMODEL	0x20110021	E1	-	-	0 (PORT)
CFG-NAVSPG-DCMMODE	0x20110023	E1	-	-	0 (DEFAULT)
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	-	-	298.25722356300002502
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

Table 87: 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)

Configuration item	Key ID	Type	Scale	Unit	Default value
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-FILT_NAVIC	0x10930018	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 88: 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	1e-1	m/s	10
CFG-ODO-COGMAXPOSACC	0x20220022	U1	-	-	50
CFG-ODO-VELLPGAIN	0x20220031	U1	-	-	153
CFG-ODO-COGLPGAIN	0x20220032	U1	-	-	76

Table 89: 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	350

Table 90: 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

Configuration item	Key ID	Type	Scale	Unit	Default value
CFG-RATE-TIMEREF	0x20210003	E1	-	-	1 (GPS)

Table 91: CFG-RATE 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-ACCEPT_NOT_IN_PRNMASK	0x30360008	X2	-	-	0x0000
CFG-SBAS-USE_IONOONLY	0x10360007	L	-	-	0 (false)
CFG-SBAS-PRNSCANMASK	0x50360006	X8	-	-	0x000000000003ab88 (ALL | PRN123 | PRN127 | PRN128 | PRN129 | PRN131 | PRN133 | PRN135 | PRN136 | PRN137)

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
CFG-SEC-SPOOFDET_SIM_SIG_DIS	0x10f6005d	L	-	-	0 (false)
CFG-SEC-JAMDET_SENSITIVITY_HI	0x10f60051	L	-	-	1 (true)

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-GPS_L5_ENA	0x10310004	L	-	-	1 (true)
CFG-SIGNAL-SBAS_ENA	0x10310020	L	-	-	1 (true)
CFG-SIGNAL-SBAS_L1CA_ENA	0x10310005	L	-	-	1 (true)
CFG-SIGNAL-GAL_ENA	0x10310021	L	-	-	1 (true)
CFG-SIGNAL-GAL_E1_ENA	0x10310007	L	-	-	1 (true)
CFG-SIGNAL-GAL_E5A_ENA	0x10310009	L	-	-	1 (true)
CFG-SIGNAL-GAL_E5B_ENA	0x1031000a	L	-	-	0 (false)
CFG-SIGNAL-GAL_E6_ENA	0x1031000b	L	-	-	1 (true)
CFG-SIGNAL-BDS_ENA	0x10310022	L	-	-	1 (true)
CFG-SIGNAL-BDS_B1_ENA	0x1031000d	L	-	-	0 (false)
CFG-SIGNAL-BDS_B1C_ENA	0x1031000f	L	-	-	1 (true)

Configuration item	Key ID	Type	Scale	Unit	Default value
CFG-SIGNAL-BDS_B2_ENA	0x1031000e	L	-	-	0 (false)
CFG-SIGNAL-BDS_B2A_ENA	0x10310028	L	-	-	1 (true)
CFG-SIGNAL-BDS_B3_ENA	0x10310010	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	-	-	0 (false)
CFG-SIGNAL-QZSS_L5_ENA	0x10310017	L	-	-	0 (false)
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)
CFG-SIGNAL-NAVIC_ENA	0x10310026	L	-	-	1 (true)
CFG-SIGNAL-NAVIC_L5_ENA	0x1031001d	L	-	-	0 (false)

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_TP2	0x4005000d	U4	1e-6	s	1000000
CFG-TP-PERIOD_LOCK_TP2	0x4005000e	U4	1e-6	s	1000000
CFG-TP-FREQ_TP2	0x40050026	U4	-	Hz	1
CFG-TP-FREQ_LOCK_TP2	0x40050027	U4	-	Hz	1
CFG-TP-LEN_TP2	0x4005000f	U4	1e-6	s	0
CFG-TP-LEN_LOCK_TP2	0x40050010	U4	1e-6	s	100000
CFG-TP-DUTY_TP2	0x5005002c	R8	-	%	0
CFG-TP-DUTY_LOCK_TP2	0x5005002d	R8	-	%	10
CFG-TP-USER_DELAY_TP2	0x40050011	I4	1e-9	s	0
CFG-TP-TP2_ENA	0x10050012	L	-	-	1 (true)
CFG-TP-SYNC_GNSS_TP2	0x10050013	L	-	-	1 (true)
CFG-TP-USE_LOCKED_TP2	0x10050014	L	-	-	1 (true)
CFG-TP-ALIGN_TO_TOW_TP2	0x10050015	L	-	-	1 (true)
CFG-TP-POL_TP2	0x10050016	L	-	-	1 (true)
CFG-TP-TIMEGRID_TP2	0x20050017	E1	-	-	0 (UTC)
CFG-TP-DRSTR_TP2	0x20050036	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)
CFG-UART1-REMAP	0x10520006	L	-	-	0 (false)

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)
CFG-UART2-REMAP	0x10530006	L	-	-	0 (false)

Table 106: CFG-UART2 configuration defaults

Configuration item	Key ID	Type	Scale	Unit	Default value
CFG-UART2INPROT-UBX	0x10750001	L	-	-	1 (true)
CFG-UART2INPROT-NMEA	0x10750002	L	-	-	1 (true)
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)

Configuration item	Key ID	Type	Scale	Unit	Default value
CFG-UART2OUTPROT-RTCM3X	0x10760004	L	-	-	0 (false)

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	-	-	427
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-X20P-00B Data sheet, UBXDOC-963802114-12690

[2] ZED-X20P integration manual, UBXDOC-963802114-12900

[3] RTCM Standard 10403.4 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 2.0.2, February 2022

☞ 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	Status / Comments
R01	09-Sep-2024	HPG 2.00 release

Contact

u-blox AG

Address: Zürcherstrasse 68
8800 Thalwil
Switzerland

For further support and contact information, visit us at www.u-blox.com/support.