DATA COMMUNICATION PROTOCOLS FOR NAVIGATION DEVICES WITH FIRMWARE VERSION 6.74

2021

TABLE OF CONTENTS

1 GENERAL CONCEPT OF GNSS MODULE INTERFACE

Data exchange between the navigation module and processing computer (host) includes the following types of information:

A. From the host to the module: commands for performing specified actions or setting up the module's parameters.

B. From the module to the host: decoded broadcast navigation data, GNSS observables (GNSS raw measurements: pseudoranges, carrier phases, Doppler etc.), responses to host commands containing execution acknowledgements or current values of module's parameters.

To transfer this data, the following data formats are used:

- NMEA 0183
- NTL Binary
- Novatel OEMv7
- RTCM 3.3

Upon turning the navigation module on, the NMEA 0183 (also known as IEC 61162-1 standard) data format is used. The protocol is designed for transferring navigation information in the form of text messages to the host. It provides message transfer only in the direction from the module to the host. Compared with the standard set of NMEA 0183 sentences, there are modifications for outputting additional parameters, which are not specified in NMEA 0183 standard. For more information, see Section 2.

The NTL Binary protocol is designed to output GNSS navigation information and raw measurements in a binary form. In addition, it allows one to configure the module, monitor its status and update the firmware. The format implies two-way messaging between the module and the host.

Messages from the host can have the following form:

- Commands to perform operations. Optionally, the commands may contain some data fields.
- Commands for setting up the module's parameters. These commands contain data.
- Commands for reading current values of module's parameters. Such commands do not contain any data. The module response containing the required data is implied.

Messages from the module are the followings:

- Responses to the module's commands (aperiodic messages).
- GNSS navigation information and raw measurements (periodic messages).

The module sends a response to all host commands. The response can be one of three types:

- Confirmation of the successful execution of the command.
- Error message: the command is not supported, invalid data field values, etc.

- Response containing data, if required by the command.

The delay to module's response to a command may be on the order of 100 milliseconds or greater. The actual execution time of given command depends on its type. Therefore, the recommended operating mode when working with the module is to always wait for the module's response to a command. The response confirms the execution of the command and indicates the readiness of the module to receive the next one.

While operating in NMEA mode, the module is also ready to receive and execute the host commands. However, the module does not send confirmations and responses with data. Thus, there is a possibility to configure the module settings in NMEA mode, but there is no option to control over command processing.

The periodic messages of NTL Binary interface provide access to a full range of information generated by navigation receiver. Some of them duplicate and complement NMEA messages ("sentences" according to NMEA 0183 terminology), others – extend and make additional information available. The details of the protocol implementation are given in Section 3.

The Novatel OEMv7 format is intended for transmission of GNSS observables (raw measurements) and other information related to the consumer equipment. This format is implemented on the basis of the protocol used in Novatel navigation receivers of family 6.x, 7.x.

RTCM messages are generated in accordance with RTCM 10403.3 standard (October, 2016). They are designed for transmission of GNSS raw measurements along with other relevant information. Current firmware version (6.70) supports Legacy message types (1004, 1006, 1012, 1019, 1020) as well as some more up to date (MSM7, 1041, 1042, 1046) messages. The given set of message types allows transmission of GPS, GLONASS, BeiDou, Galileo and NavIC raw measurements and ephemerids.

If RTCM or Novatel OEMv7 format is not a point or interest, raw data can be received in NTL Binary format as well.

All available data messages and commands are grouped in logical data channels (LDC). Following logical channels are available:

A. Master-channel. A bidirectional channel used to control navigation receiver. Utilize only NTL Binary commands and appropriate messages to transmit requests.

B. NavData flow. A set of messages used to transmit navigation products and related information to the host. NTL Binary or NMEA formats may be used.

C. NTLB raw data flow. A set of NTL Binary messages used to transmit raw ranging measurements and related information to the host.

D. RTCM data flow. A set of RTCM3.3 messages used to transmit raw ranging measurements and related information to the host.

E. Novatel raw data flow. A set of messages in Novatel OEM format used to transmit raw ranging measurements and related information to the host.

It is expected, that all messages united in one LDC have some common properties: common purpose, data format, transmission trigger, physical channel to be transmitted through and so on.

Using the configuration commands, the correspondence between logical and physical data channels can be established. The total number and type of available physical data channels are determined by the type and hardware configuration of the device. Up to four UART, CAN etc. modules may be available. Configuration in which two logical channels use one physical channel is allowed. In such case different data formats may be mixed in one physical port (ex. NTL Binary and RTCM) what can impede proper data processing at the host side. It is not recommended to work in such mode but if the only physical channel is a requirement, some precautions can be made.

The first method assumes using the NTL Binary message as a shell for RTCM3 or Novatel message. In such mode a message will get a prefix and checksum defined in accordance with NTL Binary protocol. The host will receive BMT_RAW_DATA message that includes completed Novatel or RTCM message as a data field. Utilization of such transport message excludes data format misalignment but supposes some additional actions at the host processor side to refine original data.

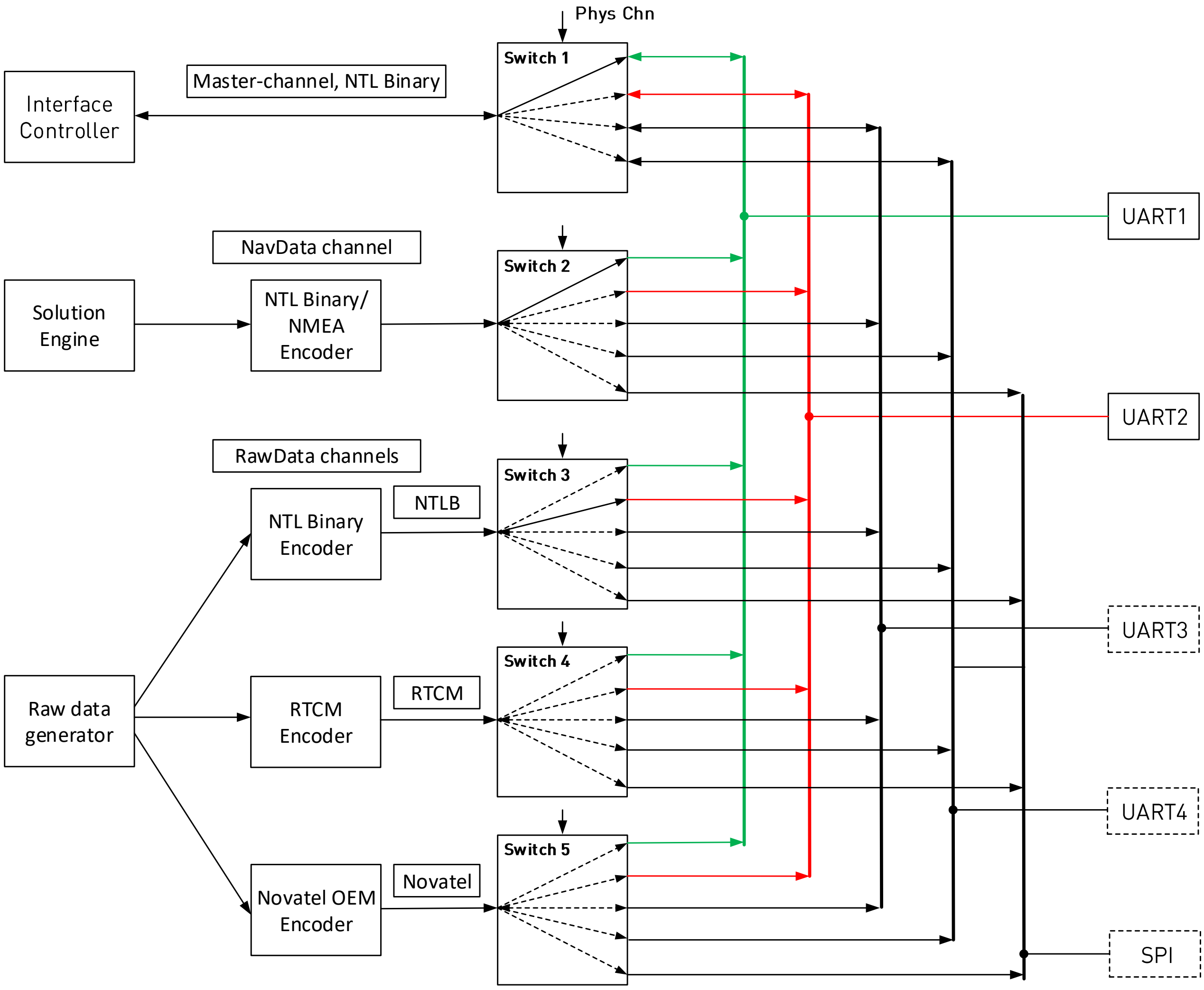
The second method supposes time division to be used to transmit not relevant data through the common data channel. While executing the commands of module's configuration, generation of RTCM or Novatel data should be disabled. When configuration is over, generation of RTCM or Novatel data may be restored. Mixing up of responses to NTL Binary commands and raw measurements may impede correct commands processing at the host side. Time multiplexing may be used to prevent such situations and guarantee stable interaction with the receiver.

Physical data channel is active if corresponding hardware circuitry is switched on and configuration is done. Unused channels can be switched off to decrease power consumption. Any active physical data channel can be switched to Master mode if a special sequence of bytes is sent to this channel externally (see Section 3, BMT_CONFIG.SET_MASTER command). Having such a sequence obtained, the module performs following actions:

- Commutates Master-channel to the corresponding physical port;
- Commutates NavData flow to the same port and makes NTL Binary a data format to transmit navigation products.
- If RTCM or Novatel raw data have been transmitted through this physical channel, then raw data generation would have been disabled to prevent data formats discrepancy.

Thus, after correct processing of the BMT_CONFIG.SET_MASTER command, the navigation module enters the ready mode to receive upcoming commands for configuration.

In general, the navigation module is ready to receive and execute the next binary protocol command, if the response to the previous command has been obtained. The module executes the received command first, then sends a response. Exceptions are the commands that affect the interface operation mode: changing the UART baud rate, changing NMEA/NTL Binary/Raw Data modes, switching physical interface channels. In these cases, upon receiving such a command, the module first completes sending all the data, which are already available in the transmitter buffer, outputs a response to the command and only after that performs a change in the interface settings. The command BMT_CONFIG.LOOP_BACK allows checking whether the interface is ready for use or not. When this command has been received, the module sends RequestCompleted response but does not perform any other actions.



2 NMEA 0183

There are two versions of NMEA 0183 standard, implemented in receiver firmware. Version 4.11 (December, 2018) strictly corresponds to official standard. Refer to official documentation for details. Version 2.3 is based on IEC 61162-1 specification (year 2000) and has some proprietary modifications. Modifications were implemented to adopt this data format to actual environments, when multiple GNSS systems and signals are available. The implementation details are given in Tables 2.1, 2.2, 2.3, 2.4 below. NMEA v2.3 will be supported further only for the reason on backward compatibility with earlier firmware versions.

Table 2.1 – NMEA v2.3 modifications and extensions

Type	Details of implementation	Update rate (by default)
	Legacy NMEA 0183 sentences	
GPTXT	It is a text message. It is displayed only once upon turning the navigation module on or after software reset. It contains information about the firmware version and build date.	
xxGGA	Possible values of xx prefix are: GP, GL, GN. "Geoidal separation" data field is left empty. "Antenna altitude" data field represents the altitude above WGS 84 ellipsoid.	1 Hz
xxGSA	Possible values of xx prefix are: GP, GL, GA, GB, GI, GN. Update rate corresponds to the position update rate. If tracking of the given navigation system is disabled, the message with the corresponding prefix is not output. The satellite with the number "n" will be available in the list of active satellites, if, at least, one of its frequency bands signals is used for computing of navigation solution.	1 Hz
xxZDA	Possible values of xx prefix are: GP, GL, GN. Update rate corresponds to the position update rate	1 Hz
xxVTG[1]	Possible values of xx prefix are: GP, GL, GN. Update rate corresponds to the position update rate	1 Hz
xxRMC[1]	Possible values of xx prefix are: GP, GL, GN. Update rate corresponds to the position update rate	1 Hz
xxGST	Possible values of xx prefix are: GP, GL, GN. Update rate corresponds to the position update rate	Off
xxGSV	Possible values of xx prefix are: GP, GL, GA, GB, GI, GS. If tracking of the given navigation system is disabled, the message with the corresponding prefix is not outputted.	0.2 Hz
	Proprietary extensions	
xxHDG[1]	Possible values of xx prefix are: GP, GL, GN. This sentence contains information about heading determination (for navigation modules with dual antenna input) and relevant data.	1 Гц

Notes

1. Information about Course Over Ground, which is available as contents of RMC and VTG sentences, is determined by the vector of velocity of the navigation module. As such, it may not correspond to the information about heading and pitch/roll from HDG sentence, which is determined by high accurate differential methods of positioning. These methods are available only for multi antenna systems having the heading determination function activated.

Legacy NMEA 2.3 messages suppose only GP, GL, GN prefixes to start messages with. To deal with multiple GNSS systems the set was extended in accordance with Table 2.2.

Table 2.2 – Prefixes of NMEA v2.3 sentences

Prefix	Description	
GP	Data obtained from GPS satellites	NMEA 0183 v2.3
GL	Data obtained from GLONASS satellites	
GN	Data obtained from GNSS systems	
GA	Data obtained from Galileo satellites	Proprietary extensions
GB	Data obtained from BeiDou satellites	
GI	Data obtained from NavIC (IRNSS) satellites	
GS	Data obtained from SBAS satellites	

To identify signals coming from multiple GNSS on multiple frequencies signals encoding was implemented in accordance with the Table 2.3.

Table 2.3 – Satellites IDs vs. frequency bands (GSV messages). Valid for NMEA v2.3 mode.

GNSS	Satellites IDs		
	1..38	(1..37)+64	(1..38) +128
	Frequency bands		
GPS	L1	L2	-
GLONASS	L1	L2	-
Galileo	E1	E5b	E5a
Beidou	B1	B2	-
NavIC	L5	S	-
SBAS	L1	-	

Enumeration of SBAS satellites implemented in accordance with Table 2.4 (only for NMEA v2.3 mode).

Table 2.4 – SBAS satellites IDs. Valid for NMEA v2.3 mode

GSGSV number	Satellite number	SBAS subsystem	GSGSV number	Satellite number	SBAS subsystem
1	120	-	20	139	-
2	121	EGNOS	21	140	SDCM
3	122	-	22	141	SDCM
4	123	EGNOS	23	142	-
5	124	-	24	143	-
6	125	SDCM	25	144	-
7	126	EGNOS	26	145	-
8	27	GAGAN	27	146	-
9	128	GAGAN	28	147	-
10	129	MSAS	29	148	-
11	130	-	30	149	-
12	131	WAAS	31	150	-
13	132	GAGAN	32	151	-
14	133	WAAS	33	152	-
15	134	-	34	153	-
16	135	WAAS	35	154	-
17	136	EGNOS	36	155	-
18	137	MSAS	37	156	-
19	138	WAAS	38	157	-
Note: GSGSV satellite numbers have constant shift (+128) while tracked on L5					

Both NMEA v2.3 and v4.11 implementations have some proprietary sentences used to transmit additional information. Messages are listed in Table 2.5. Check Appendix C for detailed message structure.

Table 2.5 – Proprietary sentences implemented both in NMEA v2.3 and v4.11.

Prefix	Description	Update rate
$PELE110	Indian regional time	1 Hz
$PELE111	GAGAN messaging service information	Aperiodic. Trigged by MT63 reception.
$PELE112	GAGAN-1A messaging service information	Aperiodic. Trigged by valid data subframe reception.
$PELE113	NavIC MT18 messaging service information	Aperiodic. Trigged by valid MT18 reception.

3 NTL_BINARY PROTOCOL

3.1 Message structure

Length of the NTL Binary message may range from 8 to 4104 bytes depending on the size of data field. The message structure is shown in the Table 3.1 below.

Table 3.1 – NTL Binary message structure

Message = {SB1 SB2 MSGT ID LL LH DATA CSA CSB}		
Offset	**Field**	**Description**
0	SB1	First byte of synchronization. Contains value 0x21 (ASCII '!').
1	SB2	Second byte of synchronization. Contains value 0x4E (ASCII 'N').
2	MSGT	Message type. Defines the functional purpose.
3	ID	The most significant bit of this field determines the type of message: ID[7] == '1': message type set_ , command to perform desired operations. ID[7] == '0': message type get_ , request data transfer. The less significant bits ID [6…0] identify the command.
4	LL	The less significant bit of the data field size.
5	LH	The most significant bit of the data field size.
6	DATA	Data field (may be zero length).
6-4102	CSA	The first byte of the checksum.
7-4103	CSB	The second byte of the checksum.

Table 3.2 – Command types by functional purpose

Name	MSGT	Description	Comment
BMT_STATUS	0	The module response to the host command (all aperiodic messages).	Module to host
BMT_DATA	1	Navigation data (periodic messages).	Module to host
BMT_RAW_DATA	2	NTL binary messages to transmit raw data.	Module to host
BMT_CONFIG	3	Commands allowing configuration of various receiver parameters related to the interface, positioning mode, etc.	
BMT_RFIC	4	Access to the configuration registers of RF front-end. May be unavailable for some firmware implementations.	
	5	Not used	Delated
BMT_FIRMWARE	6	Commands for flash memory access. Applied for navigation module's firmware update as well as for update control.	
BMT_RESTART	7	Commands for restarting the module.	
BMT_RAW_CTRL	8	Enables/disables/configures raw data generation.	
	9	Not used	Delated
BMT_HEALTH	10	Module self-checking data request	
BMT_NT1057_CTRL	11	Reserved	
BMT_CONST_CTRL	12	Constellation control and low level firmware settings.	
BMT_PPP_CTRL	13	Commands to set up parameters of PP Positioning mode.	

3.2 Algorithm for receiving and parsing the receiver commands

When decoding NTL binary messages, it is recommended to follow the steps below.

1. Detect 2 bytes of synchronization.
2. Obtain the data length field and check it for the maximum value. If it exceeds the maximum value, proceed to step 7.
3. Obtain the specified data field length.
4. Obtain two bytes of checksum.
5. Calculate and verify the checksum. If the checksum is not correct, proceed to step 7.
6. Decode the message type and perform required operations.
7. Flush the incoming command buffer and proceed to wait for the new command, step 1.

The checksum is calculated using the Fletcher algorithm (two bytes at the output) for all the command fields except the first two (sync bytes). Example of optimized implementation in C:

```
void fletcher16(unsigned char* checkA, unsigned char* checkB, unsigned char* data, unsigned int len)
{
  unsigned int tlen;
  unsigned short sum1, sum2;

  sum1 = sum2 = 0xFF;

  while (len)
  {
          tlen = len > 21 ? 21 : len;
          len -= tlen;

          do
          {
                  sum1 += *data++;
                  sum2 += sum1;
          }
          while (--tlen);

          sum1 = (sum1 & 0xFF) + (sum1 >> 8);
          sum2 = (sum2 & 0xFF) + (sum2 >> 8);
  }

  sum1 = (sum1 & 0xFF) + (sum1 >> 8);
  sum2 = (sum2 & 0xFF) + (sum2 >> 8);

  *checkA = (unsigned char) sum1;
  *checkB = (unsigned char) sum2;
}
```

4 BMT_STATUS TYPE MESSAGES

Table 4.1 - BMT_STATUS type messages

BMT_STATUS, MSGT = 0			
Messages which are sent by the module as the response to the binary commands.			
Direction: from the module to the host.			
Message = {0x21,0x4E,0x00,ID,DL,DATA,CSA,CSB}			
ID	DL	DATA	Alias
0x00	0	-	RequestError. The most recent command was not processed properly
0x01	0	-	RequestCompleted. The most recent command was executed successfully
0x02	1… 4096	Data in accordance with the most recent command	DataReturned. The most recent command was executed successfully, and the module returned data.

5 BMT_DATA TYPE MESSAGES

Messages of the given type are intended for transmission of navigation products and auxiliary information to the host (NavData flow). Data format of NavData flow should be set to NTL Binary. Messages are generated periodically. Update rate of messages, containing navigation products (position, velocity, ...) is predefined by position update rate. Other's tempo is programmed via user interface or assigned to a fixed value.

Table 5.1 - BMT_DATA type messages

BMT_DATA, MSGT = 1			
Periodically generated data.			
Direction: from the module to the host.			
Message = {0x21,0x4E,0x01,ID,DL,DATA,CSA,CSB}			
Messages, which are completely analogous to NMEA sentences			
ID	DL	DATA	Alias
0x01	1...2048	ASCII line with text information	TXT_DATA
0x02	-	BIN_GGA_DATA	GGA_DATA[4]
0x03	-	BIN_GSA_DATA	GSA_DATA
0x04	-	BIN_GSV_DATA	GSV_DATA
0x05	-	BIN_ZDA_DATA	ZDA_DATA
0x06	-	BIN_VTG_DATA	VTG_DATA[1]
0x07	-	BIN_RMC_DATA	RMC_DATA[1]
0x08	-	BIN_GST_DATA	GST_DATA
0x09	-	BIN_HDG_DATA	HDG_DATA[2]
0x0a	-	BIN_GNS_DATA	GNS_DATA[3]
0xb	-	BIN_GMS_MSG	GMS_MSG
0xc	-	BIN_NMS_MSG	NMS_MSG
0xd	-	BIN_MT18_MSG	MT18_MSG
Special NTL_Binary messages			
ID	DL	DATA	Alias
0x10	17	BIN_RTIME_RDATE	RTIME_RDATE
0x11	29	BIN_RPOS_ECEF	RPOS_ECEF
0x12	29	BIN_RVEL_ECEF	RVEL_ECEF
0x13	29	BIN_RPOS_LLA	RPOS_LLA
0x14	35	BIN_RVEL_ENU	RVEL_ENU
0x15	71	BIN_RSOL_ATR	RSOL_ATR
0x16	29	BIN_BPOS_ECEF	BPOS_ECEF
0x17	N*8	BIN_GSV_FULL	GSV_FULL
0x18	72	BIN_RSOL_FULL	RSOL_FULL
0x19	42	BIN_RSOL_iFULL	RSOL_iFULL
0x1a	N*10	BIN_GSV2_FULL	GSV2_FULL
0x1b	29	BIN_PACR_ENU	POS_ACR_ENU
0x1c		BIN_YPR_DATA	YPR_DEG
0x1d		BIN_HVECT_DATA	HVECT_DATA
0x1e	29	BIN_PACR_ECEF	POS_ACR_ECEF
0x1f	29	BIN_VACR_ECEF	VEL_ACR_ECEF

Note 1. Information about Course Over Ground, which is available as contents of BIN_RMC and BIN_VTG sentences, is determined by the vector of velocity of the navigation module. As such, it may not correspond to the information about heading and pitch/roll from HDG_DATA, YPR_DEG, HVECT_DATA messages, which is determined by high accurate differential methods of positioning. These methods are available only for multi antenna systems having the heading determination function activated.

Note 2. Message is not native for NMEA 0183 standard. Such proprietary extensions are described in Appendix C.

Note 3. Available only when NMEA mode is 4.11.

Note 4. Available only when NMEA mode is 2.3.

Messages listed in the Table 5.2 are completely analogous to NMEA messages.

Table 5.2 – Messages completely analogous to NMEA

<table>
<tr><th>Data field</th><th>Description</th></tr>
<tr><td>BIN_TXT_DATA</td><td rowspan="13">These messages contain an ASCII string inside. The content of this string is completely analogous to NMEA sentence of corresponding type. The same contents of messages can be obtained directly by means of generation of NMEA sentences. Detailed information about NMEA sentences can be found in NMEA 0183 standard except for sentences belonging to proprietary extensions of NMEA 0183 standard (See Appendix C).</td></tr>
<tr><td>BIN_GGA_DATA</td></tr>
<tr><td>BIN_GSA_DATA</td></tr>
<tr><td>BIN_GSV_DATA</td></tr>
<tr><td>BIN_ZDA_DATA</td></tr>
<tr><td>BIN_VTG_DATA</td></tr>
<tr><td>BIN_RMC_DATA</td></tr>
<tr><td>BIN_GST_DATA</td></tr>
<tr><td>BIN_HDG_DATA</td></tr>
<tr><td>BIN_GNS_DATA</td></tr>
<tr><td>BIN_GMS_MSG</td></tr>
<tr><td>BIN_NMS_MSG</td></tr>
<tr><td>BIN_MT18_MSG</td></tr>
</table>

NTL Binary messages with ID = [0x01..0x0a] implement a transport layer to transmit textual NMEA data to the host. May be useful when host application in interested in NMEA data and there is the only physical channel to transmit it. In such case binary transport messages prevent interfacing process of format misalignment with Master-channel data flow. While there are two available versions of NMEA format – 4.11 and 2.3 – content of BIN_XXX_DATA message depends on NMEA version parameter (controlled by

BMT_CONFIG.NMEA_VER command). BIN_GNS_DATA message is available only while NMEA is 4.11. BIN_GGA_DATA message is available only while NMEA is 2.3.

The following types of binary data are used in the binary messages structures:

Table 5.3 – Data types

Symbol	Description
intN	N-bit length two's complement signed integer
uintN	N-bit length unsigned integer
SGL	Single-precision floating-point number
DBL	Double-precision floating point number

The byte is little-endian: from the least significant bit to the most significant bit. The data transmission is started with LSB.

Further text includes the description of periodical NTL Binary messages. These messages include unified data fields. Description of unified the data fields is given in the Table 5.4.

Table 5.4 – Data fields of binary messages

<table>
<tr><th>Data field</th><th>Data type</th><th colspan="2">Description</th></tr>
<tr><td>TAG</td><td>uint32</td><td colspan="2">Data synchronization tag is a data field containing integer value which is incremented upon generation of a new set of navigation products. All messages having the same data synchronization tag are referenced to the same time epoch.</td></tr>
<tr><td rowspan="6">TD_STS</td><td rowspan="6">uint8</td><td colspan="2">Status of data set containing information about date and time</td></tr>
<tr><td>Value</td><td>Description</td></tr>
<tr><td>0</td><td>Information about date and time is unavailable</td></tr>
<tr><td>1</td><td>Information about time is available</td></tr>
<tr><td>2</td><td>Information about date is available</td></tr>
<tr><td>3</td><td>Information about date and time is available</td></tr>
<tr><td rowspan="4">VEL_STS</td><td rowspan="4">uint8</td><td colspan="2">Status of data set containing information about computed velocity</td></tr>
<tr><td>Value</td><td>Description</td></tr>
<tr><td>0</td><td>There is no velocity solution. Data fields related to velocity are invalid</td></tr>
<tr><td>1</td><td>There is a velocity solution. Data fields related to velocity are valid</td></tr>
<tr><td rowspan="9">POS_STS</td><td rowspan="9">uint8</td><td colspan="2">Status of data set ("Solution Status") containing information about computed position</td></tr>
<tr><td>Value</td><td>Description</td></tr>
<tr><td>0</td><td>There is no position solution. Data fields related to position are invalid</td></tr>
<tr><td>1</td><td>Autonomous (stand-alone) solution (SINGLE)</td></tr>
<tr><td>2</td><td>Code differential solution (CDDIFF)</td></tr>
<tr><td>3</td><td>Converging PPP solution</td></tr>
<tr><td>4</td><td>RTK solution with fixed ambiguities (FIX)</td></tr>
<tr><td>5</td><td>RTK solution with float ambiguities (FLOAT) or converged PPP solution.</td></tr>
<tr><td>6</td><td>Mode of extrapolation of the most recent valid position</td></tr>
</table>

Table 5.5 – Type BIN_RTIME_RDATE

<table>
<tr><th>Offset</th><th>Data field</th><th>Data type</th><th>Units</th><th>Description</th></tr>
<tr><td>0x00</td><td>TAG</td><td>uint32</td><td>-</td><td>Data synchronization tag</td></tr>
<tr><td>0x04</td><td>YEAR</td><td>uint16</td><td>year</td><td rowspan="3">Year/month/day at the moment of generation of raw measurements</td></tr>
<tr><td>0x06</td><td>MONTH</td><td>uint8</td><td>month</td></tr>
<tr><td>0x07</td><td>DAY</td><td>uint8</td><td>day</td></tr>
<tr><td>0x08</td><td>UTC</td><td>DBL</td><td>second</td><td>UTC time since the beginning of the day</td></tr>
<tr><td>0x10</td><td>TD_STS</td><td>uint8</td><td>-</td><td>Solution Status, see Table 5.4</td></tr>
<tr><td></td><td>Total number</td><td>17</td><td>bytes</td><td>Contains information about the epoch to which raw measurements are referenced.</td></tr>
</table>

Table 5.6 – Type BIN_RPOS_ECEF

Offset	Data field	Data type	Units	Description
0x00	TAG	uint32	-	Data synchronization tag
0x04	X	DBL	meters	X-coordinate of the antenna (ECEF)
0x0c	Y	DBL	meters	Y-coordinate of the antenna (ECEF)
0x14	Z	DBL	meters	Z-coordinate of the antenna (ECEF)
0x1c	POS_STS	uint8	-	Solution Status, see Table 5.4
	Total number	29	bytes	Coordinates of the antenna (ECEF)

Table 5.7 – Type BIN_BPOS_ECEF

Offset	Data field	Data type	Units	Description
0x00	TAG	uint32	-	Data synchronization tag
0x04	X	DBL	meters	Coordinates (ECEF) of the reference (base) station. Valid when POS_STS equal to CDDIF, FLOAT or FIX (differential mode).
0x0c	Y	DBL	meters	
0x14	Z	DBL	meters	
0x1c	POS_STS	uint8	-	Solution Status, see Table 5.4. Represents the status of differential solution, based on this reference station position.
	Total number	29	bytes	Coordinates (ECEF) of the reference (base) station

Table 5.8 – Type BIN_RVEL_ECEF

Offset	Data field	Data type	Units	Description
0x00	TAG	uint32	-	Data synchronization tag
0x04	VX	DBL	meters/sec	Components of the vector of antenna velocity (ECEF)
0x0c	VY	DBL	meters/sec	
0x14	VZ	DBL	meters/sec	
0x1c	VEL_STS	uint8	-	Solution Status, see Table 5.4
	Total number	29	bytes	Components of the vector of antenna velocity (ECEF)

Table 5.9 – Type BIN_RPOS_LLA

Offset	Data field	Data type	Units	Description
0x00	TAG	uint32	-	Data synchronization tag
0x04	LAT	DBL	degrees	Geodetic latitude (WGS 84)
0x0c	LON	DBL	degrees	Geodetic longitude (WGS 84)
0x14	HGT	DBL	meters	Height above WGS 84 ellipsoid
0x1c	POS_STS	uint8	-	Solution Status, see Table 5.4
	Total number	29	bytes	Geodetic coordinates of the antenna (WGS 84)

Table 5.10 – Type BIN_RVEL_ENU

Offset	Data field	Data type	Units	Description
0x00	TAG	uint32	-	Data synchronization tag
0x04	VE	DBL	meters/sec	Components of the vector of antenna velocity (ENU). The center of ENU (East-North-Up) local coordinate system corresponds to the current position of the antenna.
0x0c	VN	DBL	meters/sec	
0x14	VU	DBL	meters/sec	
0x1c	VEN	DBL	meters/sec	Horizontal Speed (Speed Over Ground)
0x24	COG	DBL	degrees	True Course
0x2c	VEL_STS	uint8	-	Solution Status, see Table 5.4
	Total number	45	bytes	Components of the vector of antenna velocity (ENU).

Table 5.11 – Type BIN_RSOL_ATR

Offset	Data field	Data type	Units	Description
0x00	TAG	uint32	-	Data synchronization tag
0x04	VDOP	SGL	-	Geometric Dilutions of Precision: VDOP, HDOP, PDOP
0x08	HDOP	SGL	-	
0x0c	PDOP	SGL	-	
0x10	GPS_SAM	uint64	-	Bit masks reflecting which satellites were used in the current navigation solution. The used bits: {1..32} for GPS, {1..24} for GLO, {5..36} for BDO, {1..36} for GAL, {1..22} for NavIC. Zero bit is not used.
0x18	GLO_SAM	uint64	-	
0x20	BDO_SAM	uint64	-	
0x28	GAL_SAM	uint64	-	
0x30	IRN_SAM	uint64	-	
0x38	TOTAL_SA	uint32	-	Total number of used satellites
0x3c	ARF	SGL	-	Indicator of ambiguity resolution in RTK mode
0x40	BD_AGE	uint32	seconds	Age of data received from the reference (base) station
0x44	TD_STS	uint8	-	Solution Status (Time), see Table 5.4
0x45	POS_STS	uint8	-	Solution Status (Position), see Table 5.4
0x46	VEL_STS	uint8	-	Solution Status (Velocity). see Table 5.4
	Total number	71	bytes	Attributes of the current navigation solution

Table 5.12 – BIN_GSV_FULL structure

Offset	Data field	Data type	Units
0x00	Sat_Data_0	BIN_GSV_SINGLE	–
0x08	Sat_Data_1	BIN_GSV_SINGLE	–
...	...		–
(N-1)*8	Sat_Data_N	BIN_GSV_SINGLE	–
	Total number	N*8, Nmax = 80	bytes

The BIN_GSV_FULL data structure contains information about tracked satellites and satellites used for position computation. It has got a periodic structure, consists of N fields of BIN_GSV_SINGLE type. Each field of BIN_GSV_SINGLE type contains information about one satellite. The data field size of the corresponding command may be of variable length Len = N * 8, where N is the number of tracked satellites, 8 is the size of the BIN_GSV_SINGLE structure in bytes. The maximum value of N is 80. It corresponds to maximum length equal to 640 bytes (80 * 8 = 640).

Table 5.13 – BIN_GSV_SINGLE structure

Offset	Data field	Data type	Units of measurements	Description
0x00	GNSS	uint8	–	Navigation system: ‘0’ – GPS; ‘4’ – NavIC; ‘1’ – GLO; ‘5’ – SBAS. ‘2’ – BDO; ‘3’ – GAL;
0x01	SAT	uint8	–	Satellite ID GPS – 1..32; NavIC – 1..22; GLO – 1..24; SBAS – 120..158; BDO – 5..37; GAL – 1..36;
0x02	Azimuth	uint16	Degrees	Azimuth
0x04	Elevation	uint8	Degrees	Elevation angle
0x05	SNR_L1	uint8	dB/Hz	The C/N corresponding to the primary frequency band (L1, B1, E1. L5 for NavIC). Zero value in this field indicates that for the given frequency band, the signal from the corresponding satellite is not tracked.
0x06	SNR_L2	uint8	dB/Hz	The C/N corresponding to the secondary frequency band (L2, B2, E5b/E5a. S-band for NavIC). Zero value in this field indicates that for the given frequency band, the signal from the corresponding satellite is not tracked.
0x07	Active	uint8	-	A non-zero value indicates that the satellite is used for position computation.
	Total number	8	bytes	Contains information about tracked and used satellites

Table 5.14 – Type BIN_RSOL_FULL

Offset	Data field	Data type	Units	Description
0x00	TAG	uint32	-	Data synchronization tag
0x04	POS_STS	uint8	-	Solution Status (Position, Velocity, Time), see Table 5.4
0x05	VEL_STS	uint8	-	
0x06	TD_STS	uint8	-	
0x07	SVN	uint8	-	Total number of used satellites
0x08	LAT	DBL	degrees	Geodetic latitude (WGS 84)
0x10	LON	DBL	degrees	Geodetic longitude (WGS 84)
0x18	HGT	DBL	meters	Height above WGS 84 ellipsoid
0x20	SPEED	DBL	meters/sec	Horizontal Speed (Speed Over Ground)
0x28	COG	DBL	degrees	True Course
0x30	CLMB	DBL	meters/sec	Vertical Speed
0x38	TIME	DBL	seconds	UTC time since the beginning of the day
0x40	HDOP	SGL	-	Horizontal Dilution Of Precision
0x44	YEAR	uint16	Year	Date
0x46	MONTH	uint8	Month	
0x47	DAY	uint8	Day	
	Total number	72	bytes	Navigation products in floating point format

Table 5.15 – Type BIN_RSOL_iFULL

Offset	Data field	Data type	Units	Description
0x00	TAG	uint32	-	Data synchronization tag
0x04	POS_STS	uint8	-	Solution Status (Position, Velocity, Time), see Table 3.7
0x05	VEL_STS	uint8	-	
0x06	TD_STS	uint8	-	
0x07	SVN	uint8	-	Total number of used satellites
0x08	X	int32	centimeters	X-coordinate of the antenna (ECEF)
0x0c	Y	int32	centimeters	Y-coordinate of the antenna (ECEF)
0x10	Z	int32	centimeters	Z-coordinate of the antenna (ECEF)
0x14	VX	int32	mm/sec	X-component of the vector of antenna velocity (ECEF)
0x18	VY	int32	mm/sec	Y-component of the vector of antenna velocity (ECEF)
0x1c	VZ	int32	mm/sec	Z-component of the vector of antenna velocity (ECEF)
0x20	TIME	int32	milliseconds	UTC time since the beginning of the day
0x24	HDOP	uint16	*10	Horizontal Dilution Of Precision
0x26	YEAR	uint16	year	Date
0x28	MONTH	uint8	month	
0x29	DAY	uint8	day	
	Total number	42	bytes	Navigation products in integer format

Table 5.16 – Type BIN_YPR_DATA

Offset	Data field	Data type	Units	Description
0x00	TAG	uint32	-	Data synchronization tag
0x04	POS_STS	uint8	-	Solution Status (Position, Time) for MS[1] vector. See Table 5.4
0x05	TD_STS	uint8	-	
0x06	TOT_SA	uint8	-	Total number of satellites that were used for computing MS[1] vector.
0x07	HMODE	uint8	-	Mode: 0x02: there are two antennas; Pitch and Yaw have been computed 0x03: there are three antennas; Pitch, Yaw and Roll have been computed
0x08	ROLL	DBL	degrees	Roll
0x10	PITCH	DBL	degrees	Pitch
0x18	YAW	DBL	degrees	Yaw
0x20	TIME	DBL	seconds	UTC time since the beginning of the day
	Total number	40	bytes	Information about attitude of platform

[1]MS – vector connecting the phase centers of the Master and Slave antennas (SA => MA). Components of MS vector are used for determination of the attitude of platform.

Table 5.17 – Type BIN_HVECT_DATA

Offset	Data field	Data type	Units of measurements	Description
0x00	TAG	uint32	-	Data synchronization tag
0x04	POS_STS	uint8	-	Solution Status (Position, Time) for MS[3] vector. See Table 5.4
0x05	TD_STS	uint8	-	
0x06	TOT_SA	uint8	-	Total number of satellites that were used for computing MS[3] vector.
0x07	ARF	uint8	*10	Indicator of ambiguity resolution in RTK mode for MS[3] vector.
0x08	SX	DBL	meters	X-coordinate of SA[1] (ECEF)
0x10	SY	DBL	meters	Y-coordinate of SA[1] (ECEF)
0x18	SZ	DBL	meters	Z-coordinate of SA[1] (ECEF)
0x20	MX	DBL	meters	X-coordinate of MA[2] (ECEF)
0x28	MY	DBL	meters	Y-coordinate of MA[2] (ECEF)
0x30	MZ	DBL	meters	Z-coordinate of MA[2] (ECEF)
0x38	TIME	DBL	seconds	UTC time since the beginning of the day
	Total number	64	bytes	Coordinates of the phase centers of antennas of the attitude system

[1]SA – Slave Antenna: additional antenna;
[2]MA – Master Antenna: main antenna;
[3]MS – vector connecting the phase centers of Slave and Master antennas (SA–>MA).

Table 5.18 – BIN_GSV2_FULL structure

Offset	Data field	Data type	Units
0x00	Sat_Data_0	BIN_GSV2_SINGLE	–
0x0a	Sat_Data_1	BIN_GSV2_SINGLE	–
...	...		–
(N-1)*10	Sat_Data_N	BIN_GSV2_SINGLE	–
	Total number	N*10, Nmax = 80	bytes

The BIN_GSV2_FULL is an analog of BIN_GSV_FULL structure but contains additional information about tracked signals. It is periodical, consists of BIN_GSV2_SINGLE blocks. The data field size of the corresponding command may be of variable length Len = N * 10, where N is the number of tracked satellites, 10 is the size of the BIN_GSV2_SINGLE structure in bytes. The maximum value of N is 80, what corresponds to maximum data field length of 800 bytes.

Table 5.19 – BIN_GSV2_SINGLE structure

Offset	Data field	Data type	Units of measurements	Description
0x00	GNSS	uint8	–	Navigation system: '0' – GPS; '4' – NavIC; '1' – GLO; '5' – SBAS. '2' – BDO; '3' – GAL;
0x01	SAT	uint8	–	Satellite ID GPS – 1..32; NavIC – 1..22; GLO – 1..24; SBAS – 120..158; BDO – 5..37; GAL – 1..36;
0x02	SGN_L1	uint8	-	Signal Identifier of primary frequency. Refer to Section 13 for DCP signal encoding tables.
0x03	SGN_L2	uint8		Signal Identifier of secondary frequency. Refer to Section 13 for DCP signal encoding tables.
0x04	Azimuth	uint16	Degrees	Azimuth
0x06	Elevation	uint8	Degrees	Elevation angle
0x07	SNR_L1	uint8	dB/Hz	The C/N corresponding to the primary frequency band (L1, B1, E1. L5 for NavIC). Zero value in this field indicates that for the given frequency band, the signal from the corresponding satellite is not tracked.
0x08	SNR_L2	uint8	dB/Hz	The C/N corresponding to the secondary frequency band (L2, B2, E5b/E5a. S-band for NavIC). Zero value in this field indicates that for the given frequency band, the signal from the corresponding satellite is not tracked.
0x09	Active	uint8	-	A non-zero value indicates that the satellite is used for position computation.
	Total number	10	bytes	Contains information about tracked and used satellites

Table 5.20 – Type BIN_PACR_ENU

Offset	Data field	Data type	Units	Description
0x00	TAG	uint32	-	Data synchronization tag
0x04	STD_EW	DBL	meters	Standard deviation of longitude error
0x0c	STD_NS	DBL	meters	Standard deviation of latitude error
0x14	STD_UD	DBL	meters	Standard deviation of altitude error
0x1c	POS_STS	uint8	-	Solution Status, see Table 5.4
	Total number	29	bytes	Estimates of ENU coordinate errors

Table 5.21 – Type BIN_PACR_ECEF

Offset	Data field	Data type	Units	Description
0x00	TAG	uint32	-	Data synchronization tag
0x04	STD_X	DBL	meters	Standard deviation of X coordinate
0x0c	STD_Y	DBL	meters	Standard deviation of Y coordinate
0x14	STD_Z	DBL	meters	Standard deviation of Z coordinate
0x1c	POS_STS	uint8	-	Solution Status, see Table 5.4
	Total number	29	bytes	Estimates of coordinate errors in ECEF coordinate system

Table 5.22 – Type BIN_VACR_ECEF

Offset	Data field	Data type	Units	Description
0x00	TAG	uint32	-	Data synchronization tag
0x04	STD_VX	DBL	meters	Standard deviation of X component of velocity vector
0x0c	STD_VY	DBL	meters	Standard deviation of Y component of velocity vector
0x14	STD_VZ	DBL	meters	Standard deviation of Z component of velocity vector
0x1c	VEL_STS	uint8	-	Solution Status, see Table 5.4
	Total number	29	bytes	Estimates of XYZ-components of velocity error vector in ECEF coordinate system

6 BMT_CONFIG TYPE MESSAGES

Table 6.1 – Structure of BMT_CONFIG messages

BMT_CONFIG, MSGT = 3					
Receiver setup commands.					
Direction: from the module to the host.					
Message = {0x21,0x4E,0x03,ID,DL,DATA,CSA,CSB}.					
			Interface		
ID	DL	DATA	Type	Purpose	Alias
0x00	4	{0x6F,0x70,0x65,0x6E}MSB	Action	Switches physical channel to Master-channel mode	SET_MASTER
0x01	0	--	Action	Connection check	LOOP_BACK
0x82	4	BIN_URATE_DATA	Write	UART1 baud rate	UART1_CTRL
0x02	0	--	Read		
0x83	4	BIN_URATE_DATA	Write	UART2 baud rate	UART2_CTRL
0x03	0	--	Read		
0x84	4	BIN_URATE_DATA	Write	UART3 baud rate	UART3_CTRL
0x04	0	--	Read		
0x85	4	BIN_URATE_DATA	Write	UART4 baud rate	UART4_CTRL
0x05	0	--	Read		
0x86	1	BIN_CHN_DATA	Write	Physical port chosen for Master-channel	MAST_CHN
0x06	0	--	Read		
0x87	1	BIN_CHN_DATA	Write	Physical port chosen for NTLB raw data flow	NTLB_RD_CHN
0x07	0	--	Read		
0x88	1	BIN_ND_FORMAT	Write	NavData data protocol type (NMEA/NTL Binary)	ND_FORMAT
0x08	0	--	Read		
0x89	4	BIN_PMSG_MASK	Write	Controls set of periodical messages	PM_MASK
0x09	0	--	Read		
0x8a	1	BIN_GSV_RATE	Write	GSV messages update rate	GSV_RATE
0x0a	0	--	Read		
0x8b	1	BIN_BD_CHN	Write	Channel for receiving data from the reference (base) station	BD_CHN
0x0b	0	--	Read		
0x8c	1	BIN_BD_TYPE	Write	Data format of the reference (base) station	BD_TYPE
0x0c	0	--	Read		
0x8d	2	BIN_BD_LAT	Write	Threshold of the transport delay for raw data message synchronization	BD_LAT
0x0d	0	--	Read		
0x0e	0	--	Read	Physical port chosen for NavData flow	ND_CHN
0x8e	1	BIN_CHN_DATA	Write		
0x0f	0	--	Read	Physical port chosen for RTCM raw data flow	RTCM_RD_CHN
0x8f	1	BIN_CHN_DATA	Write		
0x10	0	--	Read	Physical port chosen for NOVATEL OEM raw data flow	NOV_RD_CHN
0x90	1	BIN_CHN_DATA	Write		

0x11	0		Read	Version of NMEA format: pure 4.11 or modified 2.3	NMEA_TYPE
0x91	1	BIN_NMEA_TYPE	Write		
0x12	0	--	Read	Glue NavData channel to Master channel	GLUE_ND2MAST
0x92	1	BIN_GLUE_ND_TYPE	Write		
0x13	0	--	Read	Channel for reception of SSR corrections	SSR_CHN
0x93	1	BIN_CHN_DATA	Write		
Positioning attributes					
ID	DL	DATA	Type	Purpose	Alias
0xa0	1	BIN_GNSS_DATA	Write	Navigation systems selection	GNSS_SET
0x20	0	--	Read		
0xa1	4	BIN_SBAS_MASK	Write	SBAS satellites selection	SBAS_MASK
0x21	0	--	Read		
0xa2	1	BIN_DNM_DATA	Write	Receiver motion model	DNM_CTRL
0x22	0	--	Read		
0xa3	1	BIN_SRATE_DATA	Write	Navigation solutions update rate	SRATE_CTRL
0x23	0	--	Read		
0xa4	1		Write	RESERVED	RESERVED
0x24	0	--	Read		
0xa5	3	BIN_SMASK_DATA	Write	Elevation angle, C/No, HDOP limits control	SMASK_CTRL
0x25	0	--	Read		
0xa6	1	BIN_RTK_DNM	Write	Mode of motion in RTK mode	RTK_DNM
0x26	0	--	Read		
0xa7	1	BIN_RTK_QLT	Write	Method of positioning in RTK mode	RTK_QLT
0x27	0	--	Read		
0xa8	1	BIN_FIX_SYS	Write	Mask of navigation systems for ambiguity resolution	FIX_SYS
0x28	0	--	Read		
0xa9	1	BIN_ARF_THLD	Write	ARF threshold	ARF_THLD
0x29	0	--	Read		
0xaa	1	BIN_BD_AGE	Write	Maximum age of data received from the reference (base) station	BD_AGE
0x2a	0	--	Read		
0xab	4	BIN_REFBL_LEN	Write	Length of baseline for heading determination system	BL_LEN
0x2b	0	--	Read		
0xac	1	BIN_SOL_MODE	Write	Positioning mode	SOL_MODE
0x2c	0	--	Read		
0xad	32	BIN_IONO_MODEL	Write	Parameters for user defined ionospheric model	IONO_MODEL
0x2d	0	--	Read		
0xae	1	BIN_TROPO_CTRL	Write	Control of tropospheric error corrections	TROPO_CTRL
0x2e	0	--	Read		
0xaf	4	BIN_IONO_CTRL	Write	Control of ionospheric error corrections	IONO_CTRL
0x2f	0	--	Read		
0xb0	1	BIN_SBAS_CTRL	Write	SBAS corrections control	SBAS_CTRL
0x30	0	--	Read		

Others					
ID	DL	DATA	Type	Purpose	Alias
0xc0	5	BIN_PPS_DATA	Write	Calibration value for PPS rising front position	PPS_CTRL
0x40	0	--	Read		
0xc1	3	BIN_DATE_DATA	Write	Calendar data for GPS and GLONASS navigation systems	DATE_CTRL
0x41	0	--	Read		
0x42	1	BIN_STG_DATA	Action	Save current setting to FLASH block	STG_SAVE
0xc3	1	BIN_STGM_DATA	Write	Select and activate one of preset blocks from FLASH.	STGM_CTRL
0x43	0	--	Read		
0xc4	32	BIN_USER_TAG	Write	Set/Read 32-byte tag to identify device or set of controls	USER_TAG
0x44	0	--	Read		

The BMT_CONFIG commands of the "write" type are intended for setting up the module operation parameters. These commands have got a non-zero length data field and the most significant bit of the ID field equal to "1". The data field content is defined by the command type (its ID) and described further in this section. If the command is successfully executed, the module responds with the BMT_STATUS.RequestCompleted message. The commands of the "read" type are not intended to perform the data transmission and they have got a zero-length data field. The module responds to the given command with a message of the BMT_STATUS.ReturnedData type. This message has got BIN_XXXX_DATA format and corresponds to the previous request. The commands of the "action" type force the module to perform certain action. Data fields of such commands may contain some parameters. The module sends BMT_STATUS.RequestCompleted acknowledge after successful command execution.

Table 6.2 – Type BIN_URATE_DATA

Structure of DATA field of URATE_DATA command		
UART module baud rate setup.		
Byte	**Type**	**Description**
3..0	uint32	Nominal UART baud rate in the range from 0 to 921600. '0' value turns the module off.

Table 6.3 – Type BIN_CHN_DATA

Structure of DATA field of CHN_DATA command			
Codes for physical data channel selection			
Byte	**Type**	**Value**	**Description**
0	uint8	0x01	UART1
		0x02	UART2
		0x03	UART3
		0x04	UART4
		0x05	SPI1 (only receiver output)
		0x06	CAN

Table 6.4 – Type BIN_BD_TYPE

Structure of DATA field of BD_TYPE command			
RTK base station data format			
Byte	**Type**	**Value**	**Description**
0	uint8	0x00	Not used
		0x01	Novatel
		0x02	RTCM3

Table 6.5 – Type BIN_ND_FORMAT

Structure of DATA field of ND_FORMAT command			
Selection of the interface type.			
Byte	**Type**	**Value**	**Description**
0	uint8	0x00	Not used
		0x01	NTL Binary
		0x02	NMEA

Table 6.6 – Type BIN_PMSG_MASK

Structure of DATA field of PM_MASK command				
Setting the mask enables periodic messages generation. “1” – “enabled”				
Byte	**Bit**	**Description**		**Notice**
3..0		NTL_Binary	NMEA	
	0	Not used		
	1	TXT_DATA	GPTXT	
	2	GGA_DATA	xxGGA	NMEA_FORTMAT=2.3
	3	GSA_DATA	xxGSA	
	4	GSV_DATA	xxGSV	
	5	ZDA_DATA	xxZDA, PELE110	
	6	VTG_DATA	xxVTG	
	7	RMC_DATA	xxRMC	
	8	GST_DATA	xxGST	
	9	HDG_DATA	xxHDG	
	10	GNS_DATA	xxGNS	NMEA_FORTMAT=4.11

	11	GMS_MSG	PELE111	
	12	NMS_MSG	PELE112	
	13	MT18_MSG	PELE113	
	15..14	Not used		
	16	RTIME_RDATE		
	17	RPOS_ECEF		
	18	RVEL_ECEF		
	19	RPOS_LLA		
	20	RVEL_ENU		
	21	RSOL_ATR		
	22	BPOS_ECEF		
	23	GSV_FULL		
	24	RSOL_FULL		
	25	RSOL_iFULL		
	26	GSV2_FULL		
	27	RPOS_ACR_ENU		
	28	YPR_DATA		
	29	HVECT_DATA		
	30	RPOS_ACR_ECEF		
	31	RVEL_ACR_ECEF		

Table 6.7 – Type BIN_GSV_RATE

Structure of DATA field of GSV_RATE command		
Setting of the GSV message transmission update rate		
Byte	**Type**	**Description**
0	uint8	GSV messages update rate in seconds

Table 6.8 – Type BIN_BD_CHN

Structure of DATA field of BD_CHN command			
Codes for RTK base station input channel selection			
Byte	**Type**	**Value**	**Description**
0	uint8	0x01	UART1
		0x02	UART2
		0x03	UART3
		0x04	UART4

Table 6.9 – Type BIN_BD_LAT

Structure of DATA field of BD_LAT command		
Maximum value of transport delay used during raw data message synchronization.		
Byte	**Type**	**Description**
1..0	uint16	Range from 0 to 1000. Units of measurements: milliseconds.

Table 6.10 – Type BIN_GNSS_DATA

Structure of DATA field of GNSS_DATA command		
Constellation mask that enables the use of various navigation systems		
Byte	Bit	Description
0	0	1 – enable GPS
	1	1 – enable GLONASS
	2	1 – enable BDO
	3	1 – enable GAL
	4	1 – enable NavIC
	7..5	Not used

Table 6.11 – Type BIN_SBAS_MASK

Structure of DATA field of SBAS_MASK command		
SBAS satellites mask. SBAS satellites constellation to be tracked is defined with command BMT_CNSTL_CTRL.SBAS_CTRL (refer Section 13). Up to 6 satellites is available. SBAS_MASK allows manual control over satellites to be used in navigation solution. If more than one satellite is enabled, the only satellite to receive data from will be selected automatically in accordance with internal logic.		
Byte	Bit	Description
0	0	'1' – SBAS_SAT_1 enabled
	1	'1' – SBAS_SAT_2 enabled
	2	'1' – SBAS_SAT_3 enabled
	3	'1' – SBAS_SAT_4 enabled
	4	'1' – SBAS_SAT_5 enabled
	5	'1' – SBAS_SAT_6 enabled
	6..7	Reserved
1..3	0..7	Reserved
Refer to Table 12.4 for {SAT_1..SAT_6} identifiers.		

Table 6.12 – Type BIN_DNM_DATA

Structure of DATA field of DNM_DATA command			
Selection of the object motion mode.			
Byte	Type	Value	Description
0	uint8	0x00	Static & OCXO
		0x01	Static & TCXO
		0x02	Mid. Dynamic
		0x03	Max. Dynamic

Table 6.13 – Type BIN_SRATE_DATA

Structure of DATA field of SRATE_DATA command		
Navigation solutions update rate.		
Byte	Type	Frequency
0	uint8	Available values are 1, 2, 5, 10 Hz.

Table 6.14 – Reserved

Byte	Bit	Description

Table 6.15 – Type BIN_SMASK_DATA

Structure of DATA field of SMASK_DATA command		
Setting of thresholds for elevation angle, PDOP and C/No.		
Byte	Type	Description
0	uint8	The maximum PDOP value for which the solution is available.
1	uint8	The minimum value of the elevation angle of satellites used in the solution.
2	uint8	The minimum signal-to-noise ratio (C/N0) of satellites used in the solution.

Table 6.16 – Type BIN_RTK_DNM

Structure of DATA field of RTK_DNM command			
Mode of motion in RTK mode			
Byte	Type	Value	Description
0	uint8	0x00	RTK_STATIC
		0x01	RTK_KINEMATIC
		0x03	RTK_MOVING_BASE

Table 6.17 – Type BIN_RTK_QLT

Structure of DATA field of RTK_QLT command			
RTK mode			
Byte	Type	Value	Description
0	uint8	0x02	DIFF
		0x05	FLOAT
		0x04	FIX

Table 6.18 – Type BIN_FIX_SYS

Structure of DATA field of FIX_SYS command		
Mask of navigation systems for ambiguity resolution		
Byte	**Bit**	**Description**
0	0	1 – enable GPS
	1	1 – enable GLONASS
	2	1 – enable BDO
	3	1 – enable GAL
	4	1 – enable NavIC
	7..5	Not used

Table 6.19 – Type BIN_ARF_THLD

Structure of DATA field of ARF_THLD command		
ARF threshold. It defines the threshold for making decision on correct ambiguity resolution.		
Byte	**Type**	**Description**
0	uint8	Dimensionless value in the range from 0 to 100.

Table 6.20 – Type BIN_BD_AGE

Structure of DATA field of BD_AGE command		
Maximum age of data received from the reference (base) station		
Byte	**Type**	**Description**
0	uint8	Integer value in the range from 0 to 100 seconds.

Table 6.21 – Type BIN_PPS_DATA

Structure of DATA field of PPS_DATA command			
Synchronization of PPS mark position.			
Byte	**Type**	**Description**	
3..0	int32	Calibration value of the PPS mark position. Least significant bit value is 1 ns.	
4	uint8	Code of reference time system to which the rising edge of PPS mark is synchronized	
		Code	System
		0	Automatically select
		1	GPS
		2	GLONASS
		3	BeiDou
		4	Galileo
		5	NavIC
		6	UTC
		7	No synchronization

Table 6.22 – Type BIN_DATE_DATA

Structure of DATA field of DATE_DATA command		
Setting the calendar data.		
Byte	Type	Description
0	uint8	gpsCycle – Current number of GPS 1024-week cycle
1	uint8	gpsUTCshift – a shift between GPS system time and UTC, seconds
2	uint8	glnN4 – Current number of GLONASS four-year period

Table 6.23 – Type BIN_STG_DATA

Structure of DATA field of STG_DATA command		
Forces the receiver to save current settings to flash memory		
Byte	Value	Description
0	0x01	Save as set #1
	0x02	Save as set #2

Table 6.24 – Type BIN_STGM_DATA

Structure of DATA field of STGM_DATA command		
Enabling/disabling the use of module's settings recorded into flash memory		
Byte	Value	Description
0	0x00	Default settings will be used upon turning the power on
	0x01	Set #1 will be used upon turning the power on
	0x02	Set #2 will be used upon turning the power on

Table 6.25 – Type BIN_REFBL_LEN

Structure of DATA field of REFBL_LEN command		
Length of baseline for heading determination system		
Byte	Type	Description
3..0	uint32	Unsigned integer of uint32 type measured in centimeters. Approximate value of the length of baseline is used in heading determination engine for integrity checking of obtained solution. Zero turns the integrity checking off.

Table 6.26 – Type BIN_SOL_MODE

Structure of DATA field of SOL_MODE command			
Positioning mode			
Byte	Type	Value	Description
0	uint8	0x01	SINGLE
		0x02	RTK
		0x03	SINGLE + HEADING
		0x04	RTK + HEADING

Table 6.27 – Type BIN_IONO_MODEL

Structure of DATA field of IONO_MODEL command				
Programs values for Klobuchar model of ionospheric error.				
Offset, Byte	**Type**	**Field**	**Description**	**Default value**
0x00	SGL	a0	Scaling factor SCL is 1e-8. $a_n^{model} = a_n*SCL$	0.46566
0x04	SGL	a1		1.49011
0x08	SGL	a2		-5.96046
0x0c	SGL	a3		-5.96046
0x10	SGL	b0		79872.0
0x14	SGL	b1		65536.0
0x18	SGL	b2		-65536.0
0x1c	SGL	b3		-393216.0
Total : 32 Bytes				

Table 6.28 – Type BIN_IONO_CTRL

Structure of DATA field of IONO_CTRL command		
Programs mode of correction of ionospheric errors.		
Byte	**Value**	**Description**
0		Mode:
	0	OFF – corrections not applied
	1	USER – Klobuchar model for all satellites, parameters are user-defined (See BIN_IONO_MODEL)
	2	BDO – Klobuchar model for all satellites, parameters are acquired from BeiDou navigation data flow
	3	IRN – Klobuchar model for all satellites, parameters are acquired from IRNSS navigation data flow
	4	GPS – Klobuchar model for all satellites, parameters are acquired from GPS navigation data flow
	5	GAL – corrections in accordance with Galileo ICD for all satellites, parameter acquired from Galileo data flow
	6	AUTO_GENERIC – Klobuchar model. Model parameters selected automatically in accordance with generic navigation system (GPS model for GPS measurements, IRNSS model for IRNSS measurements and so on). When generic parameters are not available, they will be substituted with any alternative GNSS. GNSS priority is: GPS -> BDO-> IRNSS -> USER_MODEL.
	7	AUTO_REGIONAL – Klobuchar model. Model parameters selected automatically in accordance with current geographical region. Available data sources are GPS, BDO, IRNSS navigation data flow or user-defined model. Priority list depends of current latitude L: L<30 : IRN -> BDO -> USER 25<L<30 : BDO -> IRN -> GPS -> USER 30<L<40 : BDO -> GPS-> USER L>40: GPS->BDO -> USER
	8	IRN_GRID – IRNSS grid point used as the only source of ionospheric corrections for all measurements
	9	SBAS_GRID – SBAS grid point used as the only source of ionospheric corrections for all measurements

	10	DUAL_FRQ – ionosphere free combination used for dual frequency measurements.
	11	AUTO – mode of corrections for each navigation system selected automatically in accordance with the following priority list: DUAL_FRQ -> SBAS_GRID -> IRN_GRID -> AUTO_REGIONAL -> AUTO_GENERIC -> GAL -> GPS ->IRN -> BDO -> USER. Any point can be excluded from this list if appropriate bit set '1' in EXCUDE_MASK.
1..2	Uint16	EXCUDE_MASK: Bit 0 - not used Bit 1 - exclude user defined model from automatic selection Bit 10 – exclude SBAS grid point from automatic selection
3	Uint8	TRASHOLD. A situation is possible for DUAL_FRQ, SBAS_GRID, IRN_GRID modes, when only part of available measurements can be corrected (N of Ntot). (Nmin = Ntot*TRASHOLD/100). If N < Nmin no one correction value will be applied. In AUTO mode next correction mode will be examined.

Table 6.29 – Type BIN_TROPO_CTRL

Structure of DATA field of TROPO_CTRL command		
Programs mode of correction of tropospheric errors.		
Byte	**Value**	**Description**
0	0x00	Corrections not applied
	0x01	SBAS standard model (DO-229E)
	0x02	Saastamoinen model

Table 6.30 – Type BIN_SBAS_CTRL

Structure of DATA field of SBAS_CTRL command		
Programs mode of correction of tropospheric errors.		
Byte	**Value**	**Description**
0	0x00	SBAS OFF.
	0x01	SBAS ON. SBAS corrections applied to GPS L1 measurements. No one GNSS and/or frequency band used except GPS L1.
	0x02	SBAS+ ON. SBAS corrections applied to GPS L1 measurements. GPS L2/L5 measurements are not used. Other GNSS measurements are used and corrected in accordance with IONO_CTRL and TROPO_CTRL settings.

Table 6.31 – Type BIN_NMEA_TYPE

Structure of DATA field of NMEA_TYPE command		
Controls version of NMEA format to be used for NavData flow.		
Byte	**Value**	**Description**
0	0x00	NMEA 2.3 – modified version of NMEA 2.3
	0x01	Pure NMEA 4.11

Table 6.32 – Type BIN_GLUE_ND_TYPE

Structure of DATA field of GLUE_ND2MAST command		
Controls behavior of NavData flow channel.		
Byte	**Value**	**Description**
0	0x00	Physical port to transmit NavData chosen by ND_CHN command
	0x01	Physical port to transmit NavData is the same as for Master-channel and controlled by MAST_CHN command

7 BMT_RFIC TYPE MESSAGES

Table 7.1 - BMT_RFIC message type

BMT_RFIC, MSGT = 4					
Commands for RF front-end registers access					
Direction: from the host to the module.					
Message = {0x21,0x4E,0x04,ID,DL,DATA,CSA,CSB}.					
ID	DL	DATA	Type	Description	Legend
0x00	1	BIN_RFIC_RD	Read	RF front-end register read	RFIC_RD
0x80	2	BIN_RFIC_WR	Write	RF front-end register write	RFIC_WR

RFIC_RD. The command to read the register. The register address is transmitted in the data field. If the command is successfully executed, the module responds with the BMT_STATUS.ReturnedData message, which includes one byte of information in DATA field.

RFIC_WR. The command to write the register. The register address and its new value are sent in the BIN_RFIC_WR data field. The successful operation is confirmed by the BMT_STATUS.RequestCompleted message.

Table 7.2 - BIN_RFIC_WR type

Structure of DATA field of RFIC_WR command		
Write into RF front-end register		
Byte	Type	Description
0	uint8	Register address.
1	uint8	Value to be written.

8 BMT_FIRMWARE TYPE MESSAGES

Messages of this type are intended to work with flash memory of the navigation module and allow updating the module’s firmware.

Table 8.1 – Type BMT_FIRMWARE

BMT_FIRMWARE, MSGT = 6					
Update of the module firmware.					
Direction: from the host to the module.					
Message = {0x21,0x4E,0x06,ID,DL,DATA,CSA,CSB}.					
ID	**DL**	**DATA**	**Type**	**Description**	**Legend**
0x00	0	-	Read	Request for current firmware version	FRM_DEV_INFO1
0x81	1	BIN_SECT_NUM	Write	Erase flash memory sector	FRM_SECT_ERS[1]
0x02	2	BIN_PAGE_NUM	Read	Read flash memory page	FRM_PAGE_RD[1]
0x82	1024+2	BIN_PAGE_WR	Write	Write flash memory page	FRM_PAGE_WR[1]
0x83	4	BIN_FLASH_EN1	Write	Enable flash memory access (ver.< 6.60)	FRM_ENABLE1
0x84	0	--	Write	Erase USER firmware file	FRM_UFILE_ERS
0x05	2	BIN_UPAGE_RD	Read	Read USER firmware file page	FRM_UPAGE_RD
0x85	1024+2	BIN_UPAGE_WR	Write	Write USER firmware file page	FRM_UPAGE_WR
0x06	0	-	Read	Read load marker	FRM_LMARK_RD
0x86	1	BIN_LMARK	Write	Write load marker	FRM_LMARK_WR
0x07	0	-	Read	Request for current firmware version	FRM_DEV_INFO2
0x88	4	BIN_FLASH_EN2	Write	Enable flash memory access (ver. > 6.60)	FRM_ENABLE2
Note 1. Commands to be used only by authorized users					

FRM_DEV_INFO1. The request of information about device hardware and firmware configuration. The data is returned in the BMT_STATUS.ReturnedData message. The data field in the module’s response has got the FRM_DEV_INFO1 structure.

Table 8.2 – The FRM_DEV_INFO1 structure

Offset	Data field	Data type	Units of measurements	Description
0x00	MajorVersion	1U	–	Major firmware version number
0x01	MinorVersion	1U	–	Minor firmware version number
0x02	ReleaseYear	2U	–	Year of firmware release
0x04	ReleaseMonth	1U	–	Month of firmware release
0x05	ReleaseDay	1U	–	Day of firmware release
0x06	HardwareId	2U	–	Hardware ID
	Total number	8	bytes	
Notes	If device returns MajorVersion = 6 and MinorVersion = 60 use another command with ID = 0x07 to request device info in more up to date format. The FRM_DEV_INFO1 structure is supported for reasons of compatibility and does not contain any actual information about firmware and hardware version. Information from new data structure (FRM_DEV_INFO2) is mandatory for firmware update if your current version is 6.60 or higher. If firmware version is below 6.60, FRM_DEV_INFO1 contains actual information, command with ID = 0x07 will be ignored.			

FRM_DEV_INFO2. The request of information about device hardware and firmware configuration. The data is returned in the BMT_STATUS.ReturnedData message. The data field in the module's response has got the FRM_DEV_INFO2 structure.

Table 8.3 – The FRM_DEV_INFO2 structure

Offset	Data field	Data type	Units	Description
0x00	MajorHdwType	uint8	–	Hardware version
0x01	MinorHdwType	uint8	–	
0x02	MajorFwrVer	uint8	–	Firmware version
0x03	MinorFwrVer	uint8	–	
0x04	FwrOption	uint8	–	Firmware build option
0x05	BuildDay	uint8	–	Day of firmware release
0x06	BuildMonth	uint8	–	Month of firmware release
0x07	BuildYear	uint8	–	Year of firmware release (2000-based)
0x08	DeviceID_0	uint32	–	Personal device ID
0x0c	DeviceID_1	uint32		
0x10	DeviceID_2	uint32		
0x14	MinorBldrVer	uint8	–	BootLoaderVersion
0x15	MajorBldrVer	uint8	–	
0x16	...	...		Reserved
0x17	...	...		Reserved
	Total number	24	bytes	

FRM_SECT_ERS. Erases flash memory sector. The sector size is 64 KB. The sector number is declared in the BIN_SECT_NUM data field (uint8, values 0 to 15 are acceptable). The successful operation is confirmed by the BMT_STATUS.RequestCompleted message.

FRM_PAGE_RD. A request to read a page of flash memory. Command's DATA field should contain page number (uint16, values 0 to 1023 acceptable). The page size is 1024 bytes. If the command successful, the module returns the BMT_STATUS.ReturnedData message, which includes the contents of the page of flash memory as data field.

FRM_PAGE_WR. A request to write the flash memory page. The BIN_PAGE_WR data field contains the number and contents of the page. The successful operation is confirmed by the module BMT_STATUS.RequestCompleted message.

Table 8.4 – Type BIN_PAGE_WR

Structure of DATA field of FRM_PAGE_WR command	
Byte	**Description**
1..0	Number of the memory page (0..1023)
1025..2	Contents of the 1024-byte flash memory page

FRM_ENABLE1, FRM_ENABLE2. Commands to perform flash memory access enable. The BIN_FLASH_EN1 (or BIN_FLASH_EN2) field contains a request to enable/disable flash memory access, as well as some additional information necessary to verify the compatibility of hardware with the expected firmware version. If the command is successfully executed, the module responds with a BMT_STATUS.RequestCompleted message.

If current firmware version is below 6.60, command FRM_ENABLE1 should be used.

Table 8.5 – Type BIN_FLASH_EN1

Structure of DATA field of FRM_ENABLE1 command			
Enable/disable flash memory programming mode (fw. version < 6.60).			
Byte	**Type**	**Field**	**Description**
0	uint8	Operation	0x01 – Programming mode ON; 0x00 – Programming mode OFF.
1	uint8	RFIC_ID	This information should be extracted from firmware binary file and provided during Programming mode ON operation to validate expected firmware and current hardware compatibility.
2	uint8	BaseBand_ID	
3	uint8	FLASH_ID	

Required information should be extracted from binary file header. Header structure is in Appendix D. Both early and actual versions of firmware have RFIC_ID at 0x2a offset, BaseBand_ID at 0x2b, and FLASH_ID at 0x2c.

If current firmware version if above 6.60, both commands are available. Use FRM_ENABLE1 if you want to step back to version earlier than 6.60 (such binary file does not contain information mandatory for FRM_ENABLE2). Otherwise use FRM_ENABLE2.

Table 8.6 – Type BIN_FLASH_EN2

Structure of DATA field of FRM_ENABLE2 command			
Enable/disable flash memory programming mode (fw. version >= 6.60).			
Byte	**Type**	**Field**	**Description**
0	uint8	Operation	0x01 – Programming mode ON; 0x00 – Programming mode OFF.
1	uint8	MajorHdwType	This information should be extracted from firmware binary file and provided during Programming mode ON operation to validate expected firmware and current hardware compatibility.
2	uint8	MinorHdwType	
3	uint8	BldrMajorVer	

Extract DevAttribitesTable pointer at address 0x48. Then get MajorHdwType at [DevAtrPtr + 12], MinorHdwType at [DevAtrPtr + 13], BldrMajorVer at [DevAtrPtr + 9]. Check Appendix D for boot header structure. There are no DevAttribitesTable in binary files for firmware versions below 6.60. New firmware files may be identified by BootHeader marker at 0x0060 offset.

FRM_UFILE_ERS. A request to erase the USER firmware file. USER firmware file is a 512 kByte area in the flash memory containing alternative firmware version (contrary to FACTORY firmware version). Consists of 512 pages, available for reading and writing. User firmware file should be erased before writing. Successful erasing operation is confirmed by the BMT_STATUS.RequestCompleted message.

FRM_UPAGE_RD. A request to read a USER firmware file page. A page number is transmitted in the command DATA field. The page size is 1024 bytes, page number should be in the rang 0 to 511. If the command is successful, the module returns a BMT_STATUS.ReturnedData message, which includes the required page content as DATA field.

FRM_UPAGE_WR. A request to write a USER firmware file page. The BIN_UPAGE_WR DATA field contains the number and content of the page. If the command is successfully executed, the module responds with the BMT_STATUS.RequestCompleted message.

Table 8.7 – Type BIN_UPAGE_WR

BIN_UPAGE_WR	
Structure of DATA field of FRM_UPAGE_WR command	
Byte	**Description**
1..0	Number of the memory page (0..511).
1025..2	Content of the 1024-byte flash memory page.

Load marker is an indicator stored in flash memory which defines what firmware file to be loaded from (FACTORY default or USER).

FRM_LMARK_RD. A request to read the load marker. The module responds with the BMT_STATUS.ReturnedData message. The load marker is transferred in the response DATA field.

FRM_LMARK_WR. A command to set up a load marker. If the command is successfully executed, the module responds with the BMT_STATUS.RequestCompleted message.

Table 8.8 - BIN_LMARK type

BIN_LMARK	
Structure of Data field of FRM_LMARK_WR command	
Byte	**Description**
0	Load marker value: 0x00 – start up with factory default firmware; non-zero – start up with user-uploaded firmware.

Warning! IF you switch load marker to USER value while firmware is not uploaded or corrupted module may be badly damaged. If current FACTORY firmware is below version 6.60 module will not work. Otherwise module will detect mismatch and start FACTORY firmware, load marker will be ignored.

9 BMT_RESTART MESSAGE TYPE

Table 9.1 – Type BMT_RESTART

BMT_RESTART, MSGST = 7					
Navigation module restart.					
Direction: from the host to the module.					
Message = {0x21,0x4E,0x07,ID,DL,DATA,CSA,CSB}.					
ID	**DL**	**DATA**	**Type**	**Description**	**Legend**
0x80	1	BIN_RST_TYPE	Write	Software reset	PRG_RST
0x81	0	--	Write	Hardware reset similar to power-up	HDW_RST

PRG_RST. A command to trigger a software reset. The parameter in the BIN_RST_TYPE data field determines the reset mode. Before performing the reset, the module responds to the command with the BMT_STATUS.RequestCompleted message. The reset is executed only after the response has been transmitted.

Table 9.2 – Type BIN_RST_TYPE

Structure of DATA field of RRG_RST command		
Restart modes		
Byte	**Type**	**Description**
0	uint8	Restart mode: 0x01 – hot; 0x02 – warm (erase of ephemeris data); 0x03 – cold (erase of all navigation data); 0x04 – restart with default settings.

HDW_RST. A command to trigger hardware restart. The module is restarted similar to power-up, with the full firmware reboot. The command does not affect the stored navigation data (ephemeris/almanac data, most recent coordinates, time).

10 BMT_RAW_CTRL MESSAGE TYPE

Commands of this type are intended for setting up the parameters related to generation of GNSS raw measurements.

Table 10.1 – Type BMT_RAW_CTRL

BMT_RAW_CTRL, MSGST = 8					
Parameters of the raw measurements generator					
Direction: from the host to the module.					
Message = {0x21,0x4E,0x08,ID,DL,DATA,CSA,CSB}.					
Common Controls					
ID	DL	DATA	Type	Description	Legend
0x00	0	--	Read	Defines raw measurements update rate	RAW_RATE
0x80	1	BIN_RAW_RATE	Write		
0x01	0	--	Read	Defines the content of the message containing receiver position	RAW_UPOS
0x81	25	BIN_RAW_UPOS	Write		
0x02				Reserved	
0x82					
0x03	0	--	Read	Defines masks related to elevation angle, C/No, navigation systems and RAIM for generation of raw measurements	RAW_FILTER
0x83	4	BIN_RAW_FILTER	Write		
0x04	0	--	Read	Reference clock mode	RAW_CLK
0x84	1	BIN_RAW_CLK	Write		
0x05	0	--	Read	Defines maximum offset of the receiver time scale	RAW_TLR
0x85		BIN_RAW_TLR	Write		
0x06	0	--	Read	L1 band phase-correcting slope	RAW_TAU1
0x86	4	BIN_RAW_TL1	Write		
0x07	0	--	Read	L2 band phase-correcting slope	RAW_TAU2
0x87	4	BIN_RAW_TL2	Write		
0x08				Reserved	
0x88					
0x09	0	--	Read	Reference station ID.	REFST_ID
0x89	2	BIN_REFST_ID	Write		
0x0a	0	--	Read	Defines whether smoothed or not smoothed measurements should be transmitted	RAW_SMTH_OPT
0x8a	1	BIN_SMTH_OPT	Write		
0x0b	0	--	Read	Defines whether single sol. coordinate should be extrapolated to correspond range measurements rate.	BSPOS_EXT
0x8b	1	BIN_BSPOS_EXT	Write		

RTCM data flow generation					
0x10	0	--	Read	Enable/disable generation of RTCM data flow	RTCM_FLOW_EN
0x90	1	BIN_RTCM_EN	Write		
0x11	0	--	Read	RTCM message mask. Defines a set of RTCM messages to be generated	RTCM_MASK
0x91	4	BIN_RTCM_MASK	Write		
0x12	0	--	Read	Defines whether NTL Binary message should be used over RTCM transport layer	RTCM2NTLB
0x92	1	BIN_RTCM2NTLB	Write		
NTLB raw data flow generation					
0x20	0	--	Read	Enable/Disable generation of NTL Binary raw data flow	NTLRD_EN
0xa0	1	BIN_NTLRD_EN	Write		
0x21	0	--	Read	Defines a set of NTL Binary raw data messages to be generated	NTLRD_MASK
0xa1	4	BIN_NTLRD_MASK	Write		
NOVATEL OEM					
0x30	0	--	Read	Enable/disable generation of Novatel OEM raw data flow	NOVRD_EN
0xb0	1	BIN_NOVRD_EN	Write		
0x31	0	--	Read	Novatel OEM message mask. Defines a set of messages to be generated	NOVRD_MASK
0xb1	4	BIN_NOVRD_MASK	Write		
0x32	0	--	Read	Defines whether NTL Binary message should be used over Novatel OEM transport layer	NOV2NTLB
0xb2	1	BIN_NOV2NTLB	Write		

The BMT_RAW_CTRL commands of the "write" type have got a non-zero length data field. The data fields content is defined by the type of specified command (its ID) and described in this section below. If the command is successfully executed, the module responds with the BMT_STATUS.RequestCompleted message. The commands of the "read" type do not perform the data transmission, have a zero-length data field and the most significant bit of the ID field equal to '1'. The module responds to the given command with a message BMT_STATUS.ReturnedData type. Data field of this message has got BIN_RAW_XXX format corresponding to the specified request.

Table 10.2 – Type BIN_RAW_RATE

Structure of DATA field of RAW_RATE command		
Update rate of raw measurements		
Byte	Type	Description
0	uint8	Raw measurements update rate: 0 – disabled. A non-zero value sets the update rate of raw measurements. Valid values are: 1, 2, 4, 5, 10, 20 Hz.

Table 10.3 – Type BIN_RAW_UPOS

Structure of DATA field of RAW_UPOS command		
Coordinates for the message, containing receiver position (BESTPOS/1006/1005).		
Byte	Type	Description
0	uint8	Data source: 0 – own receiver solution; 1 – user-entered value.
8..1	DBL	Geodetic latitude (WGS 84), degrees
16..9	DBL	Geodetic longitude (WGS 84), degrees
24..17	DBL	Altitude (WGS 84), meters

Table 10.4 – Type BIN_RAW_FILTER

Structure of DATA field of RAW_FILTER command		
Mask for raw data generation		
Byte	Bit	Description
0		Navigation systems permission mask
	0	1 – enable GPS
	1	1 – enable GLONASS
	2	1 – enable BeiDou
	3	1 – enable Galileo
	4	1 – enable NavIC
	7..5	Not used
1	7..0	Threshold for minimum C/No value, dB/Hz
2	7..0	Threshold for elevation angle, degrees
3	7..0	0 – RAIM results will be ignored; Non-zero – satellites, which are detected by RAIM, will be excluded from positioning

Table 10.5 – Type BIN_RAW_CLK

Structure of DATA field of RAW_CLK command		
Defines source of reference time to generate raw measurements relatively to.		
Byte	**Type**	**Description**
0	uint8	0x00 – system time estimate (clock steering mode) 0x01 – receiver time (limited free running clock)

Table 10.6 – Type BIN_RAW_TLR

Structure of DATA field of RAW_TLR command		
Maximum allowable offset of the receiver time with respect to system time		
Byte	**Type**	**Description**
3..0	uint32	Specifies maximum allowable offset of the receiver time (nanoseconds) with respect to system time. Upon reaching the specified value, the receiver clock is adjusted to the system time value (as a jump-like correction).

Table 10.7 – Type BIN_RAW_TL1

Structure of DATA field of RAW_TL1 command		
GLONASS L1phase-correcting slope		
Byte	**Type**	**Description**
3..0	int32	Phase-correction slope (nanoseconds).

Table 10.8 – Type BIN_RAW_TL2

Structure of DATA field of RAW_TL2 command		
GLONASS L2 phase-correcting slope		
Byt e	**Type**	**Description**
3..0	int32	Phase-correction slope (nanoseconds).

Table 10.9 – BIN_REFST_ID

Structure of DATA field of REFST_ID command		
Byte	**Type**	**Description**
1..0	uint16	Defines value to be transmitted in 1005/ 1006/ BESTPOS messages as a reference station identifier

Table 10.10 – BIN_SMTH_OPT

Structure of DATA field of RAW_SMTH_OPT command		
Defines whether smoothed measurements should be transmitted or not		
Byte	Type	Description
0	uint8	0x00 – smoothing is ON 0x01 – smoothing is OFF

Table 10.11 – BSPOS_EXT

Structure of DATA field of BIN_SMTH_OPT command		
Defines whether single sol. coordinate should be extrapolated to correspond range measurements rate.		
Byte	Type	Description
0	uint8	0x00 – extrapolation disabled 0x01 – extrapolation enabled

Table 10.12 – BIN_RTCM_EN

Structure of DATA field of RTCM_FLOW_EN command		
Defines whether RTCM data generation enabled or not		
Byte	Type	Description
0	uint8	0x00 – disable 0x01 – enable

Table 10.13 – Type BIN_RTCM_MASK

Structure of DATA field of RTCM_MASK command		
Bitmask which controls generation of RTCM3 messages.		
Byte	Bit	Description
Novatel messages		
RTCM 3.1 messages		
3..0	0	1004 & 1012 enable
	1	MSM7 enable (1077, 1087, 1107, 1127, 1137)
	2	1 – 1019 enable
	3	1 – 1020 enable
	4	Not used
	5	1 – 1006 enable
	6	Not used
	7	1042
	8	1046
	9	1041
	31..10	Not used

Table 10.14 – BIN_RTCM2NTLB

Structure of DATA field of RTCM2NTLB command		
Defines, whether NTL Binary message should be used as a transport layer over data message generated in RTCM3 format.		
Byte	Type	Description
0	uint8	0x00 – disable 0x01 – enable

Table 10.15 – BIN_NTLRD_EN

Structure of DATA field of NTLRD_EN command		
Defines, whether NTLB raw data flow enabled or not		
Byte	Type	Description
0	uint8	0x00 – disable 0x01 – enable

Table 10.16 – Type BIN_NTLRD_MASK

Structure of DATA field of RAW_SET command		
Byte	Bit	Description
3..0	A bit mask, which defines a set of NTL Binary messages used to transmit raw data measurements, ephemerids and auxiliary information. '1' – enables the message.	
	0	Reserved
	1	RAW_RTK_DATA
	2	RAW_RINEX_DATA
	3	RAW_EXT1_DATA
	4	RAW_EXT2_DATA
	..	Reserved
	16	NTL_GLO_EPH
	17	NTL_GPS_EPH
	18	NTL_GAL_EPH
	19	NTL_BDO_EPH
	20	NTL_IRN_EPH
	..	...
	31	...

Table 10.17 – BIN_NOVRD_EN

Structure of DATA field of NOVRD_EN command		
Defines, whether Novatel OEM raw data flow enabled or not		
Byte	Type	Description
0	uint8	0x00 – disable 0x01 – enable

Table 10.18 – Type BIN_NOVRD_MASK

Structure of DATA field of NOVRD_MASK command		
A bitmask, which controls generation of NOVATEL OEM messages.		
Byte	**Bit**	**Description**
3..0	0	1 – RANGEB enable
	1	1 – RANGECMPB enable
	2	1 – GPSEPHEMB enable
	3	1 – GLOEPHEMERISB enable
	4	1 – GLOCLOCKB enable
	5	1 – BESTPOSB enable
	6	Not used
	7	1 – BDSEPHEMERIS enable
	8	1 – GALINAVEPHB enable
	9	1 – IRNEPHEMERISB enable
	10	1– IONUTC enable
	31..10	Not used

Table 10.19 – BIN_NOV2NTLB

Structure of DATA field of NOV2NTLB command		
Defines, whether NTL Binary message should be used as a transport layer over data message generated in Novatel OEM format.		
Byte	**Type**	**Description**
0	uint8	0x00 – disable 0x01 – enable

11 BMT_HEALTH MESSAGE TYPE

The messages of this type are intended for self-checking of the navigation module.

Table 11.1 – Type BMT_HEALTH

BMT_HEALTH, MSGST = 0x0a					
Self-checking of the navigation module					
Direction: from the host to the module.					
Message = {0x21,0x4E,0x0a,ID,DL,DATA,CSA,CSB}.					
ID	**DL**	**DATA**	**Type**	**Description**	**Alias**
0x00	0	--	Read	Request for self-checking data	HLTH_STS

HLTH_STS: this command reads the receiver health status. The module responds with the BMT_STATUS.ReturnedData message. The data field has got the BIN_HLTH_STS structure.

Table 11.2 – Type BIN_HLTH_STS

<table>
<tr><th colspan="3">BIN_HLTH_STS</th></tr>
<tr><td colspan="3">Information about receiver health status</td></tr>
<tr><th>Byte</th><th>Bit</th><th>Description</th></tr>
<tr><td rowspan="5">0</td><td>0</td><td>Indication of UART1 buffer overflow of navigation module: ‘1’ - buffer overflow detected.</td></tr>
<tr><td>1</td><td>Indication of UART2 buffer overflow of navigation module: ‘1’ - buffer overflow detected.</td></tr>
<tr><td>2</td><td>Indication of UART3 buffer overflow of navigation module: ‘1’ - buffer overflow detected.</td></tr>
<tr><td>3</td><td>Indication of UART4 buffer overflow of navigation module: ‘1’ - buffer overflow detected.</td></tr>
<tr><td>7..4</td><td>Not used.</td></tr>
<tr><td rowspan="2">1</td><td>0</td><td>Real time clock status
0 – OK;
1 – Fault detected;</td></tr>
<tr><td>1</td><td>Digital correlator status:
0 – OK;
1 – Fault detected;</td></tr>
<tr><td>3..2</td><td></td><td>Not used</td></tr>
</table>

12 BMT_CONSTL_CTRL TYPE MESSAGES

This block of commands allows control over some constellation specific parameters, such us list of satellites to be tracked, amount of satellites to be tracked, frequency range to be tracked and so on.

Table 12.1 – Type BMT_CNSTL_CTRL

BMT_CONST_CTRL, MSGT = 0x0c					
Control over satellites signals to be processed					
Direction: from the host to the module.					
Message = {0x21,0x4E,0x0c,ID,DL,DATA,CSA,CSB}.					
ID	DL	DATA	Type	Description	Alias
0x00	0	-	Read	Enables/disables signal processing at primary frequency (L1, E1, B1…)	PRIM_FRQ_EN
0x80	6	BIN_PRIM_FRQ	Write		
0x01	0	-	Read	Selects/disables alternative frequency to be processed (L2/L5/OFF, E5b/E5a/OFF, …)	SCND_FRQ_SEL
0x81	6	BIN_SCND_FRQ	Write		
0x02	0	-	Read	Defines set of satellites to be used for SBAS tracking	SBAS_CONST
0x82	6	BIN_SBAS_CONST	Write		
0x03	0	-	Read	Defines set of PRN codes to be used for NAVIC L5 signals processing	IRNL5_PRN
0x83	28	BIN_IRNL5_PRN	Write		
0x04	0	-	Read	Defines set of PRN codes to be used for NAVIC S band signals processing	IRNS_PRN
0x84	28	BIN_IRNS_PRN	Write		
0x05	0	-	Read	Enables/disables swapping of primary and secondary frequencies	FRQ_SWAP
0x85	6	BIN_FRQ_SWAP	Write		

The BMT_CONST_CTRL commands of the "write" type have got a non-zero length data field. The data fields content is defined by the type of specified command (its ID) and described in this section below. If the command is successfully executed, the module responds with the BMT_STATUS.RequestCompleted message. The commands of the "read" type do not perform data transmission, have a zero-length data field and the most significant bit of the ID field equal to '1'. The module responds to the given command with a message BMT_STATUS.ReturnedData type. Data field of this message has got BIN_XXX_XXX format corresponding to the specified request.

Table 12.2 – Type BIN_PRIM_FRQ

Structure of DATA field of PRIM_FRQ_EN command		
Each byte of Data field enables/disables processing of primary frequency band signals respectively to each GNSS system.		
Byte	**Type**	**Value**
0	uint8	0x00 – disable GPS L1 0x01 – enable GPS L1
1	uint8	0x00 – disable GLONASS L1 0x01 – enable GLONASS L1
2	uint8	0x00 – disable BeiDou B1 0x01 – enable BeiDou B1
3	uint8	0x00 – disable Galileo E1 0x01 – enable Galileo E1
4	uint8	0x00 – disable NavIC L5 0x01 – enable NavIC L5
5	uint8	0x00 – disable SBAS L1 0x01 – enable SBAS L1

Table 12.3 – Type BIN_SCND_FRQ

Structure of DATA field of SCND_FRQ_SEL command		
Each byte of Data field selects alternative (secondary) frequency band to be processed. Each byte is responsible for definite GNSS system.		
Byte	**Type**	**Value**
0	uint8	0x00 – disable secondary GPS frequency 0x01 – enable GPS L2 0x02 – enable GPS L5
1	uint8	0x00 – disable GLONASS L2 0x01 – enable GLONASS L2
2	uint8	0x00 – disable BeiDou B2 0x01 – enable BeiDou B2
3	uint8	0x00 – disable secondary Galileo frequency 0x01 – enable Galileo E5b 0x02 – enable Galileo E5a
4	uint8	0x00 – disable NavIC S band 0x01 – enable NavIC S band
5	uint8	0x00 – disable SBAS L5 0x01 – enable SBAS L5

Table 12.4 – Type BIN_SBAS_CONST

Structure of DATA field of BIN_SBAS_CONST command				
#Offset, Byte	Type	Field	Description	Default value
0x00	uint8	SAT_NUM_1	Up to 6 SBAS satellites may be tracked simultaneously. {SAT_NUM_1 .. SAT_NUM_6} are PRN numbers from the range 120..156 (reserved for SBAS in accordance with GPS ICD) to be used by the receiver. Unused slots should be set to zero.	123 (EGN)
0x01	uint8	SAT_NUM_2		136 (EGN)
0x02	uint8	SAT_NUM_3		121(EGN)
0x03	uint8	SAT_NUM_4		127(GAG)
0x04	uint8	SAT_NUM_5		128(GAG)
0x05	uint8	SAT_NUM_6		0

Table 12.5 – Type BIN_IRNL5_PRN

Structure of DATA field of IRNL5_PRN command				
#Offset, Byte	Type	Field	Description	Default value
0x00	uint16	G2_INIT_01	Two polynomials G1 and G2 are defined for NAVIC PRN L5 band codes generation. {G2_INIT_01 .. G2_INIT_14} are initial values of G2 polynomial, which defines PRN sequence to be generated. By default, PRN initial values are defined in accordance with IRNSS ICD version 1.1.	0x03a7
0x02	uint16	G2_INIT_02		0x0026
...	...	...		...
0x1a	uint16	G2_INT_14		0x01ea

Table 12.6 – Type BIN_IRNS_PRN

Structure of DATA field of IRNS_PRN command				
#Offset, Byte	Type	Field	Description	Default value
0x00	uint16	G2_INIT_01	Two polynomials G1 and G2 are defined for NAVIC PRN S band codes generation. {G2_INIT_01 .. G2_INIT_14} are initial values of G2 polynomial, which defines PRN sequence to be generated. By default PRN initial values are defined in accordance with IRNSS ICD version 1.1.	0x0ef
0x02	uint16	G2_INIT_02		0x17d
...	...	...		...
0x1a	uint16	G2_INT_14		0x10d

Table 12.7 – Type BIN_FRQ_SWAP

Structure of DATA field of FRQ_SWAP command		
Each byte allows to swap (primary <–>secondary) frequency range being used during search procedure.		
Byte	**Type**	**Value**
0	uint8	0x01 – swap GPS frequencies
1	uint8	0x01 – swap GLONASS frequencies
2	uint8	0x01 – swap BeiDou frequencies
3	uint8	0x01 – swap Galileo frequencies
4	uint8	0x01 – swap NavIC frequencies
5	uint8	Not used

13 BMT_RAW_DATA MESSAGE TYPE

This section describes NTL GNSS raw measurement messages. Currently, there are two types of messages defined for range data transmission: 1) RTK data, 2) RINEX data. The RTK data message serves for transmission RTK data from the reference station to the rover receiver. The RINEX data message includes all necessary information to be compatible with RTCM 10403.3 MSM5 message. The contents of RINEX data message can be used for conversion to RINEX GNSS Observation Data File along with the use of corresponding NTL ephemeris messages. Ephemeris messages are defined to provide all necessary information for further conversion into RINEX format as well as for post processing purpose. There are two types of ephemeris messages: 1) Keplerian parameters, 2) GLONASS orbital parameters. Keplerian parameters are universal form of orbital parameters for GPS, GALILEO, BeiDou and NavIC systems.

As well there is one more specific message used as a shell to transmit raw measurements packed into a NOVATEL OEMv7 or RTCM format. Formed at the receiver side, they can be transmitted via NTL Binary flow being placed as DATA field of RAW_SHELL command. Binary shell prevents normal NTL Binary data flow from being corrupted with alien data format.

This section describes only receiver-to-host messages intended to transmit data. All kinds of commands for raw data generation control are described in Section 10.

Table 13.1 – Type BMT_RAW_DATA

BMT_RAW_DATA, MSGST = 2				
Transmission raw measurements				
Direction: from the host to the module.				
Message = {0x21,0x4E,0x02,ID,DL,DATA,CSA,CSB}.				
ID	DL	DATA	Type	Alias
0	--	Novatel OEM7 or RTCM 3.1 message	NTL Binary shell for transmission of raw measurements formed in accordance with Novatel or RTCM 3.1 formats.	RAW_SHELL
1	1524 max	RAW_RTK_DATA	A set of raw measurements for further RTK processing	RAW_RTK
2	1996 max	RAW_RINEX_DATA	A set of raw measurements for further conversion into RINEX.	RAW_RINEX
3	2464 max	RAW_EXT1_DATA	Additional fields for standard deviation estimates	RAW_EXT1
4	3336 max	RAW_EXT2_DATA	Additional fields for both smoothed and unsmoothed range measurements	RAW_EXT2
...			Reserved	
16	812 max	GLO_EPH_DATA	GLONASS orbital parameters for N satellites	NTL_GLO_EPH

17	1288 max	KEPL_EPH_DATA	Keplerian orbital parameters for N satellites of definite system.	NTL_GPS_EPH
18		KEPL_EPH_DATA		NTL_GAL_EPH
19		KEPL_EPH_DATA		NTL_BDO_EPH
20		KEPL_EPH_DATA		NTL_IRN_EPH
21		GPS_IONUTC_DATA	GPS Klobuchar model and UTC shift.	NTL_GPS_IOUT
..			Reserved	
255	--	RTCM 3.1 message piece	NTL Binary shell for transmission of piece of raw measurements formed in RTCM 3.1 format.	RTCM_RAW_PIECE

Identification of DATA field content of RAW_NOV message has to be carried out via the header of the original message. Common structure of other messages will be defined below. Details about raw data messages implementation can be found in Appendix A.

Table 13.2 – Predefined data types

RNG_HDR	There is a list of data types defined for description of raw data messages. Details about their implementation can be found in Appendix A.
RTK_BLOCK	
RTK_HDR	
RTK_SLOT	
RINEX_BLOCK	
RINEX_HDR	
RINEX_SLOT	
EXT1_HDR	
EXT1_BLOCK	
EXT1_SLOT	
EXT2_HDR	
EXT2_BLOCK	
EXT2_SLOT	
KEPLEPH_BLOCK	
GLOEPH_BLOCK	
RAW_CRC	

Table 13.3 – Type RAW_RTK_DATA

Structure of DATA field of RAW_RTK message		
Field	**Type**	**Description**
MsgHeader	RNG_HDR	Mandatory field
SatData_1	RTK_BLOCK	Optional field
SatData_2	RTK_BLOCK	Optional field
…	…	…
SatData_N	RTK_BLOCK	Optional field
S_RTKPaddingBits	uintN	Optional field
CRC	RAW_CRC	CRC32 checksum. Covers fields {MsgHeader, SatData_1 ..N, PaddingBits}.
Notes: N (number of SatData blocks) is not fixed, defined in MsgHeader. Max. value is 63. Size of RTK_BLOCK is not fixed and may be different for different SatData fields. Bounds of SatData blocks are not aligned to bytes, there are no padding bits between SatData blocks. Field PaddingBits is optionally added to align total length of Header and SatData fields to 32 bits. Maximum structure size for N = 63 is [(N*189bit + 64bit)/8] = 1520 byte.		

Table 13.4 – Type RTK_BLOCK

Structure of SatData_n field of RAW_RTK_DATA message		
Field	**Type**	**Description**
BlockHeader	RTK_HDR	Mandatory field
SlotData_1	RTK_SLOT	Mandatory field
…		…
SlotData_M	RTK_SLOT	Optional field
Notes: M (number of frequency slots) is variable and defined in BlockHeader. Max. value is 3. Fields are not aligned to byte boundaries, there are no padding bits between them. Maximum structure size for M = 3 is M*52+33=189 bits.		

Table 13.5 – Type RAW_RINEX_DATA

Structure of DATA field of RAW_RINEX message		
Field	**Type**	**Description**
MsgHeader	RNG_HDR	Mandatory field
SatData_1	RINEX_BLOCK	Optional field
SatData_2	RINEX_BLOCK	Optional field
...	...	...
SatData_N	RINEX_BLOCK	Optional field
PaddingBits	uintN	Optional field
CRC	RAW_CRC	CRC32 checksum. Covers fields {MsgHeader, SatData_1 ..N, PaddingBits}.
Notes: N (number of SatData blocks) is not fixed, defined in MsgHeader. Max. value is 63. Size of RINEX_BLOCK is not fixed and may be different for different SatData fields. Bounds of SatData blocks are not aligned to bytes, there are no padding bits between SatData blocks. Field PaddingBits is optionally added to align total length of Header and SatData fields to 32 bits. Maximum structure size for N = 63 is [(N*248bit + 64bit)/8] = 1992 byte.		

Table 13.6 – Type RINEX_BLOCK

Structure of SatData_n field of RAW_RINEX_DATA message		
Field	**Type**	**Description**
BlockHeader	RINEX_HDR	Mandatory field
SlotData_1	RINEX_SLOT	Mandatory field
...		...
SlotData_M	RINEX_SLOT	Optional field
Notes: M (number of frequency slots) is variable and defined in BlockHeader. Max. value is 3. Fields are not aligned to byte boundaries, there are no padding bits between them. Maximum structure size for M = 3 is (M*67bit + 47bit) = 248 bits.		

Table 13.7 – Type GLO_EPH_DATA

Structure of DATA field of NTL_GLO_EPH message		
Field	**Type**	**Description**
SatEph_1	GLOEPH_BLOCK	Size of GLOEPH_BLOCK is fixed and equal to 58. Total DATA field length is 58*N. Maximum available N is 14. Maximum total DATA field length is 812 bytes.
...		
SatEph_N	GLOEPH_BLOCK	
CRC	RAW_CRC	CRC32 checksum. Covers all SatEph fields

Table 13.8 – Type KEPL_EPH_DATA

Structure of DATA field of NTL_GPS_EPH, NTL_GAL_EPH, NTL_BDO_EPH, NTL_IR_EPH message		
Field	**Type**	**Description**
SatEph_1	KEPLEPH_BLOCK	Size of KEPLEPH_BLOCK is fixed and equal to 92 bytes. Total DATA field length is 92*N. Maximum available N is 14. Maximum total DATA field length is 1288 bytes.
…		
SatEph_N	KEPLEPH_BLOCK	
CRC	RAW_CRC	CRC32 checksum. Covers all SatEph fields

Table 13.9 – Type GPS_IONUTC_DATA

Structure of DATA field of NTL_GPS_IOUT message				
This message contains parameters of Klobuchar ionospheric delay as well as GPS-to-UTC timescale bias. Parameters are extracted from GPS L1 data flow and transmitted "as it is". Should be properly scaled and processed in accordance with GPS ICD.				
Offset	Field	Type	Scale factor	Description
0x00	utcA1	int32	2**-30	All definitions are in accordance with GPS interface control document (IS-GPD-200J.pdf). Sign bit extended in signed values.
0x04	utcA0	int32	2**-50	
0x08	utcShift	int8	1	
0x09	wn	uint8	1	
0x0A	t	uint8	2**12	
0x0B	utcShiftLsf	int8	1	
0x0C	wnLsf	uint8	1	
0x0D	dayLsf	uint8	1	
0x0E	ionA0	int8	2**-30	
0x0F	ionA1	int8	2**-27	
0x10	ionA2	int8	2**-24	
0x11	ionA3	int8	2**-24	
0x12	ionB0	int8	2**11	
0x13	ionB1	int8	2**14	
0x14	ionB2	int8	2**16	
0x15	ionB3	int8	2**16	
0x16	CRC	uint32	-	CRC32 checksum. Covers bytes [0..0x16].

Table 13.10 – Type RAW_EXT1_DATA

Structure of DATA field of RAW_EXT1 message		
Field	Type	Description
MsgHeader	EXT1_HDR	Mandatory field
SatData_1	EXT1_BLOCK	Optional field
SatData_2	EXT1_BLOCK	Optional field
…	…	…
SatData_N	EXT1_BLOCK	Optional field
PaddingBits	uintN	Optional field
CRC	RAW_CRC	CRC32 checksum. Covers fields {MsgHeader, SatData_1 ..N, PaddingBits}.
Notes: N (number of SatData blocks) is not fixed, defined in MsgHeader. Max. value is 63. Size of EXT1_BLOCK is not fixed and may be different for different SatData fields. Bounds of SatData blocks are not aligned to bytes, there are no padding bits between SatData blocks. Field PaddingBits is optionally added to align total length of Header and SatData fields to 32 bits. Maximum structure size for N = 63 is [(N*311bit + 64bit)/8] = 2458 byte.		

Table 13.11 – Type EXT1_BLOCK

Structure of SatData_n field of RAW_EXT1_DATA message		
Field	Type	Description
BlockHeader	EXT1_HDR	Mandatory field
SlotData_1	EXT1_SLOT	Mandatory field
…		…
SlotData_M	EXT1_SLOT	Optional field
Notes: M (number of frequency slots) is variable and defined in BlockHeader. Max. value is 3. Fields are not aligned to byte boundaries, there are no padding bits between them. Maximum structure size for M = 3 is (M*88bit + 47bit) = 311 bits.		

Table 13.12 – Type RAW_EXT2_DATA

Structure of DATA field of RAW_EXT2 message		
Field	Type	Description
MsgHeader	EXT2_HDR	Mandatory field
SatData_1	EXT2_BLOCK	Optional field
SatData_2	EXT2_BLOCK	Optional field
...	...	...
SatData_N	EXT2_BLOCK	Optional field
PaddingBits	uintN	Optional field
CRC	RAW_CRC	CRC32 checksum. Covers fields {MsgHeader, SatData_1 ..N, PaddingBits}.
Notes: N (number of SatData blocks) is not fixed, defined in MsgHeader. Max. value is 63. Size of EXT2_BLOCK is not fixed and may be different for different SatData fields. Bounds of SatData blocks are not aligned to bytes, there are no padding bits between SatData blocks. Field PaddingBits is optionally added to align total length of Header and SatData fields to 32 bits. Maximum structure size for N = 63 is [(N*422bit + 64bit)/8] = 3332 byte.		

Table 13.13 – Type EXT2_BLOCK

Structure of SatData_n field of RAW_EXT2_DATA message		
Field	Type	Description
BlockHeader	EXT2_HDR	Mandatory field
SlotData_1	EXT2_SLOT	Mandatory field
...		...
SlotData_M	EXT2_SLOT	Optional field
Notes: M (number of frequency slots) is variable and defined in BlockHeader. Max. value is 3. Fields are not aligned to byte boundaries, there are no padding bits between them. Maximum structure size for M = 3 is (M*125bit + 47bit) = 422 bits.		

14 BMT_PPP_CTRL MESSAGE TYPE

Commands of this type are intended for setting up the parameters related to Precise Positioning mode.

Table 14.1 – Type BMT_PPP_CTRL

BMT_RAW_CTRL, MSGST = 0x0D					
Parameters of the raw measurements generator					
Direction: from the host to the module.					
Message = {0x21,0x4E,0x0D,ID,DL,DATA,CSA,CSB}.					
Common Controls					
ID	DL	DATA	Type	Description	Legend
0x00	0	–	Read	Switch PPP mode ON/OFF	PPP_TUMBLER
0x80	1	BIN_PPP_TUMBLER	Write		
0x01	0	–	Read	Set/get second frequency processing mode	PPP_FRQ_MODE
0x81	1	BIN_PPP_FMODE	Write		
0x02	0	–	Read	Set/get mode of motion of the object	PPP_OBJ_DYN
0x82	1	BIN_PPP_OBJ_DYN	Write		
0x03	0	–	Read	Set/get satellite elevation cutoff mask	PPP_EL_MASK
0x83	1	BIN_PPP_EL_MASK	Write		
0x04	0	–	Read	Set/get satellite C/No value cutoff mask	PPP_C2N_MASK
0x84	1	BIN_PPP_C2N_MASK	Write		
0x05	0	–	Read	Set/get initial position accuracy	PPP_POS_ACR
0x85	4	BIN_PPP_POS_ACR	Write		
0x06	0	–	Read	Set/get maximum object velocity	PPP_MAX_VEL
0x86	4	BIN_PPP_MAX_VEL	Write		
0x07	0	–	Read	Set/get Tau value	PPP_TAU
0x87	4	BIN_PPP_TAU	Write		
0x08	0	–	Read	Set/get noise of Tau value	PPP_TAU_NOISE
0x88	4	BIN_PPP_TAU_NOISE	Write		
0x09	0	–	Read	Set/get tropospheric ambiguity	PPP_TRAMB
0x89	4	BIN_PPP_TRAMB	Write		
0x0a	0	–	Read	Set/get noise of tropospheric ambiguity	PPP_TRAMB_NS
0x8a	4	BIN_PPP_TRAMB_NS	Write		
0x0b	0	–	Read	Set/get ionospheric ambiguity	PPP_IAMB
0x8b	4	BIN_PPP_IAMB	Write		
0x0c	0	–	Read	Set/get noise of ionospheric ambiguity	PPP_IAMB_NS
0x8c	4	BIN_PPP_IAMB_NS	Write		
0x0d	0	–	Read	Set/get accuracy of range measurements	PPP_RNG_ACR
0x8d	4	BIN_PPP_RNG_ACR	Write		
0x8e	0	–	Read	Set/get accuracy of phase range measurements	PPP_PH_ACR
0x8e	4	BIN_PPP_PH_ACR	Write		

Table 14.2 – Type BIN_PPP_TUMBLER

Structure of DATA field of PPP_TUMBLER command		
The only byte of Data field enables/disables PP processing mode.		
Byte	Type	Value
0	uint8	0x00 – disable PPP 0x01 – enable PPP

Table 14.3 – Type BIN_PPP_FMODE

Structure of DATA field of PPP_FRQ_MODE command		
The only byte of Data field selects mode of second frequency processing.		
Byte	Type	Value
0	uint8	0x00 – dual frequency 0x01 – iono-free

Table 14.4 – Type BIN_PPP_OBJ_DYN

Structure of DATA field of PPP_OBJ_DYN command		
The only byte of Data field specifies motion mode of the object		
Byte	Type	Value
0	uint8	0x00 – static 0x01 – kinematic 0x02 – auto

Table 14.5 – Type BIN_PPP_EL_MASK

Structure of DATA field of PPP_EL_MASK command		
The only byte of Data field specifies satellite elevation cut-off value		
Byte	Type	Value
0	uint8	Valid values: 0..70 degree.

Table 14.6 – Type BIN_PPP_C2N_MASK

Structure of DATA field of PPP_C2N_MASK command		
The only byte of Data field specifies satellite C/No cut-off value		
Byte	Type	Value
0	uint8	Valid values: 10..50 dB/Hz.

Table 14.7 – Type BIN_PPP_POS_ACR

Structure of DATA field of PPP_POS_ACR command		
Specifies accuracy estimate of initial position to start PP positioning mode.		
Byte	Type	Value
3..0	SGL	Float value in meters, range 0.0 to 1000.0 m.

Table 14.8 – Type BIN_PPP_MAX_VEL

Structure of DATA field of PPP_MAX_VEL command		
Specifies expected maximum value of receiver velocity.		
Byte	Type	Value
3..0	SGL	Float value in m/s, range 0.0 to 1000.0 m/s.

Table 14.9 – Type BIN_PPP_TAU

Structure of DATA field of PPP_TAU command		
Specifies estimate of		
Byte	Type	Value
3..0	SGL	Float value in meters, range 0.0 to 100.0 m.

Table 14.10 – Type BIN_PPP_TAU_NOISE

Structure of DATA field of PPP_TAU_NOISE command		
Specifies estimate of		
Byte	Type	Value
3..0	SGL	Float value in m/s^2, range 0.0 to 1e-3 $m/(s^{1/2})$.

Table 14.11 – Type BIN_PPP_TRAMB

Structure of DATA field of BIN_PPP_TRAMB command		
Specifies estimate of tropospheric ambiguity.		
Byte	Type	Value
3..0	SGL	Float value in meters, expected range 0.0 to 2.0 m.

Table 14.12 – Type BIN_PPP_TRAMB_NS

Structure of DATA field of PPP_TRAMB_NS command		
Specifies estimate of tropospheric ambiguity noise.		
Byte	Type	Value
3..0	SGL	Float value in m/s^2, expected range 0.0 to 0.01 m/s^2.

Table 14.13 – Type BIN_PPP_IAMB

Structure of DATA field of PPP_IAMB command		
Specifies estimate of ionospheric ambiguity.		
Byte	Type	Value
3..0	SGL	Float value in meters, expected range 0.0 to 100 m.

Table 14.14 – Type BIN_PPP_IAMB_NS

Structure of DATA field of PPP_IAMB_NS command		
Specifies estimate of ionospheric ambiguity noise.		
Byte	Type	Value
3..0	SGL	Float value in m/s^2, expected range 0.0 to 1.0 m/s^2.

Table 14.15 – Type BIN_PPP_RNG_ACR

Structure of DATA field of PPP_RNG_ACR command		
Specifies estimate of satellite range measurements accuracy.		
Byte	Type	Value
3..0	SGL	Float value in meters, expected range 0.0 to 10.0 m.

Table 14.16 – Type BIN_PPP_PH_ACR

Structure of DATA field of PPP_PH_ACR command		
Specifies estimate of satellite phase measurements accuracy.		
Byte	Type	Value
3..0	SGL	Float value in meters, expected range 1e-3 to 50.e-3 m.

APPENDIX A1. Format of NTL GNSS raw measurement messages

General format of NTL Binary raw measurement messages was described in Section 13. Intended data types are described in this section below.

No byte alignment is applied for satellite and signal parts of the messages to make them more compact. Therefore, the length of the message in bits may not line up with byte (8 bits each) boundaries. If it happens, "dummy bits" are added to the end of the message to make the length of the message equal to integer number of bytes. Total number of dummy bits is in the range of 0-7 (zero means that the length of the message is already aligned to byte boundaries and no dummy bits are required). If dummy bits are added, they should be equal to "0", "10", "010", "1010", "01010", "101010" and "0101010" to fulfill 1, 2, 3, 4, 5, 6 and 7 dummy bits respectively.

Table A1.1 RNG_HDR structure

Data field	Format	Description	Notes
D001	uint30	GNSS system time	Range: 0-604,799,999 ms. Value 0x3FFFFFFF encodes "ERROR" or "UNDEFINED" state of system time. Note: For GLONASS, this value consists of two parts: Day number within GLONASS week in the range 1-7 (0, if unknown): 3 MSB GLONASS system time in the range 0-86,399,999: 27 LSB
D002	uint14	GNSS cycle	Range: 0 – 8191 0x2000 and higher numbers indicate invalid value.
D003	uint3	System time status	Status of D001 field: 0 – tmaUNDEF; 1 – tmaROUGH; 2 – tmaNORMAL; 3 – tmaFINE;
D004	uint2	Clock steering	Indicator of clock steering mode: 0- clock steering mode is disabled; 1 – clock steering mode is enabled; 2 – unknown status of clock steering mode; 3 – reserved;
D005	uint12	Reference station ID	Range: 0-4095 Reference station ID
D006	uint2	External clock indicator	Indicator of use of external clock: 0 – internal clock is used; 1 – external clock is used; 2-3 – reserved;
D007	1 bit	Indicator of positive leap second	If this bit is set to 1, it indicates about epochs that relate to time interval of positive leap second addition. This bit shall be set, if raw measurements include GLONASS (independently on chosen GNSS system time ID)
D008	1 bit	End of data for current epoch	If this bit is set to 1, it means the next message(s) become available for the current epoch. Zero indicates

			the last message (end of data) has been received for the current epoch.
D009	uint3	GNSS system time ID	Range: 0-7 0: GPS, 1: GLONASS, 2: Galileo, 3: BeiDou, 4: IRNSS, 5-7 reserved Note: QZSS system time corresponds to GPS system time: there is no necessity to have a separate ID for QZSS system time. The same holds true with respect to SBAS systems
D010	uint6	Total number of satellites	Range: 0-63 Defines total number of satellites, which this message contains.
D011	uint2	Smoothing indicator	Range: 0-3 0 – pseudoranges are not smoothed with carrier phases measurements; 1 – pseudoranges are smoothed with carrier phases, using divergence-free smoothing; 2 – pseudoranges are smoothed with carrier phases (not divergence-free smoothing); 3- status of smoothing is unknown;
D012	uint3	Smoothing interval	Smoothing interval provides additional information, if Smoothing indicator is set to 2: 0 – reserved 1 – < 30 s 2 – 30-60 s 3 – 1-2 min 4 – 2-4 min 5 – 4-8 min 6 – >8 min 7 – reserved
D013	uint2	Extension of the header	If this data field is not equal to zero, it indicates about the total number of additional bytes (8 bits each) that follows D012. The contents of additional bytes is reserved for future needs
Total bits: 64 + 8*[D013]			

Table A1.2 RTK_HDR and RINEX_HDR structure

N	Format	Description	Notes
D012	uint8	Satellite ID	Range: 0-255 0 – satellite ID is not available 205-255 reserved; See Table 4.
D013	uint4	Total number of signal data	Range: 0-15

			Total number of signal data corresponds to this number plus one (thus, measurements for up to 16 signals can be available)
D014	uint21	Rough reference pseudorange	Range: (2^21-1): 0 – 67108832 meters. 0x1FFFFF indicates invalid value. Rough reference pseudorange is computed as the abridged value (without rounding off to the nearest value, i.e., Rough reference pseudorange is always less than or equal to the real value of pseudorange) of the pseudorange measurement associated with the first slot in the raw of slot measurements. With respect to this rough reference pseudorange, all other pseudorange and carrier phase measurements are defined. Resolution: 25 meters (32 meters).
This part of satellite data depends on Message Type:			
Message Type RTK_HDR			
-	-	-	
Total RTK_HDR size: 33 bits.			
Message Type RINEX_HDR			
D015	int14	Rough reference Doppler	Range: ±(2^13-1): ±8192 meter/second 0x2000 indicates invalid value. Rough reference Doppler serves as a reference for Doppler measurements of signal slots Resolution: 1 meter/second
Total RINEX_HDR size: 47 bits. RINEX_HDR is equal to EXT1_HDR as well as to EXT2_HDR.			

Table A1.3 RTK_SLOT and RINEX_SLOT structure

N	Format	Description	Notes
D017	uint4	Signal ID	Range: 0-15 See Table 5.
These data fields are available for the first set of signal data only:			
D018	uint4	Additional GNSS information.	Range: 0-15 For GLONASS: Frequency Channel Numbers (FCN) in the range -7 - +6 are mapped into the range 0-13 respectively, 14, 15 – reserved. For other GNSS systems: Reserved
D019	uint11	GNSS signal fine pseudorange (first set of signal data)	Range: 0 – 2^11-1: 0 – about <32 meters Pseudorange for the given signal is computed as [D014] + [D019] Resolution: 2^-6 meters (about 0.016 meters)
This data field is available for all sets of signal data, starting at the second set:			

D020	int15	GNSS signal fine pseudorange (second and next sets of signal data)	Range: ±(2^14-1): about ±256 meters 0x8000 indicates invalid value. Pseudorange for the given signal is computed as [D014] + [D020] Resolution: 2-6 meters (about 0.016 meters)
Next data fields are available for all sets of signal data			
D021	int23	GNSS signal fine phase range	Range: ±(2^22-1): about ±2048 meters 0x400000 indicates invalid value. Phase range for the given signal is computed as [D014] + [D021] Resolution: 2^-11 meters (about 0.0005 meters)
D022	uint6	SNR	Range: 1-63, 0 indicates that estimation of SNR is not available. Resolution: 1 dB-Hz.
D023	uint4	Lock time indicator	Range: 0-15 The lock time indicator is a measure of amount of time during which a continuous tracking of phase range observable was provided without cycle slips. If a cycle slip occurs, the lock indicator will be reset to zero.
This part of Signal data depends on Message Type:			
Message Type RTK_SLOT			
-	-	-	
Total: M*52 bits (M – total number of sets of signal data)			
Message Type RINEX_SLOT			
D024	int15	Doppler "delta" with respect to rough reference Doppler	Range: ±(2^14-1): ±2 m/s 0x4000 indicates invalid value. Doppler for the given slot is computed as [Rough Doppler] + [Doppler "delta"] Resolution: 2^-13 m/s (about 0.0001 m/s)
Total: M*67 bits (M – total number of sets of signal data)			
Message Type EXT1_SLOT			
D025	uint7	Pseudorange STD	Standard deviation of pseudorange measurements. Value 0 indicates that estimate is not available. Refer to Table A1.7 for decoding scheme.
D026	uint7	Phase range STD	Standard deviation of carrier phase range measurements. Value 0 indicates that estimate is not available. Refer to Table A1.8 for decoding scheme.
D027	uint7	Doppler STD	Standard deviation of Doppler measurements. Value 0 indicates that estimate is not available. Refer to Table A1.9 for decoding scheme.
-	-	-	-
Total for EXT1_SLOT: M*88 bits (M – total number of sets of signal data)			
Message Type EXT2_SLOT			
D028	uint15	Full Lock Time	Range: 0 .. 2^15-1. Amount of time during which a continuous tracking of phase range observable was provided without cycle slips [sec]. If a cycle slip occurs, the lock indicator will be

			reset to zero. Values greater than 0x7fff would be saturated to 0x7FFF.
D029	int15	GNSS signal unsmoothed fine pseudorange	Range: ±(2^14-1): about ±256 meters 0x4000 indicates invalid value. Pseudorange for the given not smoothed measurements is computed as [D014] + [D028] Resolution: 2^-6 meters (about 0.016 meters)
D030	uint7	Unsmoothed pseudorange STD	Standard deviation of unsmoothed pseudorange measurements. Value 0 indicates that estimate is not available. Refer to Table A1.7 for decoding scheme.
-	-	-	
Total for EXT2_SLOT: M*125 bits (M – total number of sets of signal data)			

Table A1.4 Satellite ID (D012) to SV ID mapping

Satellite ID (ID) (From D012)	**SV ID (SVID)**	**Notes**
GPS		
1-37	SVID= ID+0	
GLONASS		
38-69	SVID= ID-37	
Galileo		
70-105	SVID= ID-69	
BeiDou		
106-142	SVID= ID-105	
IRNSS (NavIC)		
143-156	SVID= ID-142	
QZSS		
157-166	SVID= ID-156	
SBAS		
167-204	SVID= ID-166	

Table A1.5 GNSS signals ID

Signal ID	**Frequency band**	**Signal**	**Notes**
GPS			
0	L1	C/A	
1	L1	P	
2	L1	P(Y)	
3	L2	P	
4	L2	P(Y)	
5	L2	L2C(M)	
6	L2	L2C(L)	
7	L2	L2C(M+L)	
8	L5	L5(I)	
9	L5	L5(Q)	
10	L5	L5(I+Q)	
11	L1	L1C(D)	
12	L1	L1C(P)	
13	L1	L1C-(D+P)	
14,15			Reserved
GLONASS			
0	G1	SP	
1	G1	HP	
2	G2	SP	
3	G2	HP	
4	G1	G1A	
5	G1	G1B	
6	G1	G1X	
7	G2	G2A	
8	G2	G2B	
9	G2	G2X	
10	G3	G3I	
11	G3	G3Q	
12	G3	G3X	
13-15			Reserved
Galileo			
0	E1	B	
1	E5A	E5A(I)	
2	E5A	E5A(Q)	
3	E5A	E5A(I+Q)	
4	E5B	E5B(I)	
5	E5B	E5B(Q)	
6	E5B	E5B(I+Q)	
7	E5(A+B)	I	
8	E5(A+B)	Q	
9	E5(A+B)	I+Q	
10	E1	C	
11	E1	B+C	
12-15			Reserved
BeiDou			

0	B1-2	B1(I)	1561.098
1	B1	B1CD	1575.42
2	B1	B1CP	
3	B1	B1CX	
4	B2	B2I	1207.14
5	B2	B2bD	
6	B2	B2bP	
7	B2	B2bX	
8	B2a	B2aD	1176.45
9	B2a	B2aP	
10	B2a	B2aX	
11	B3	B3(I)	1268.52
12	B3	B3(Q)	
13	B3	B3(I+Q)	
14-15			Reserved
IRNSS (NavIC)			
0	L5	SPS L5(I)	L5A
1	L5	SPS L5(Q)	L5B
2	L5	SPS L5(I+Q)	L5X
3	S-band	SPS S-band(I)	SA
4	S-band	SPS S-band(Q)	SB
5	S-band	SPS S-band(I+Q)	SX
6	L1	L1P	
7	L1	L1D	
8	L1	L1X	
5-15			Reserved
QZSS			
0	L1	C/A	
1	L2	L2C(M)	
2	L2	L2C(L)	
3	L2	L2C(M+L)	
4	L5	L5(I)	
5	L5	L5(Q)	
6	L5	L5(I+Q)	
7	L1	L1C(D)	
8	L1	L1C(P)	
9	L1	L1C-(D+P)	
10	LEX	LEX(S)	
11	LEX	LEX(L)	
12	LEX	LEX(S+L)	
13-15			Reserved
SBAS			
0	L1	C/A	
1	L5	L5(I)	
2	L5	L5(Q)	
3	L5	L5(I+Q)	
4-15			Reserved

Table A1.6 Lock time indicator

Indicator	Lock time [ms]	Range of lock time [ms]
0	0	0 ⩽ t < 32
1	32	32 ⩽ t < 64
2	64	64 ⩽ t < 128
3	128	128 ⩽ t < 256
4	256	256 ⩽ t < 512
5	512	512 ⩽ t < 1024
6	1024	1024 ⩽ t < 2048
7	2048	2048 ⩽ t < 4096
8	4096	4096 ⩽ t < 8192
9	8192	8192 ⩽ t < 16384
10	16384	16384 ⩽ t < 32768
11	32768	32768 ⩽ t < 65536
12	65536	65536 ⩽ t < 131072
13	131072	131072 ⩽ t < 262144
14	262144	262144 ⩽ t < 524288
15	524288	524288 ⩽ t

Table A1.7 Pseudorande STD (D025)

range = STD[6..5]	i = STD[4..0]	Equation	Range, [m]
0	1..31	0.0 + i*0.125	[0.125 .. 3.875]
1	0..31	4.0 + i*0.25	[4.0 .. 11.75]
2	0..31	12.0 + i*1.0	[12.0 .. 43]
3	0..31	44.0 + i*4.0	[44.0 .. 168]
Notes: 0 value encodes UNDEFINED STD status; STD estimates greater than 168 meters are saturated to 168.			

Table A1.8 Phase range STD (D026)

range = STD[6..5]	i = STD[4..0]	Equation	Range, [mm]
0	1..31	0.0 + i*0.25	[0.25 .. 7.75]
1	0..31	8.0 + i*0.5	[8.0 .. 23.5]
2	0..31	24.0 + i*1.0	[24.0 .. 55.0]
3	0..31	56.0 + i*2.0	[56.0 .. 118.0]
Notes: 0 value encodes UNDEFINED STD status; STD estimates greater than 118 millimeters are saturated to 118.			

Table A1.9 Doppler STD (D027)

range = STD[6..5]	i = STD[4..0]	Equation	Range, [m/s]
0	1..31	0.0 + i*0.125	[0.125 .. 3.875]
1	0..31	4.0 + i*0.25	[4.0 .. 11.75]
2	0..31	12.0 + i*0.5	[12.0 .. 27.5]
3	0..31	28.0 + i*1.0	[28.0 .. 59.0]
Notes: 0 value encodes UNDEFINED STD status; STD estimates greater than 59.0 meters are saturated to 59.0.			

APPENDIX A2. Raw data message CRC-32 checksum

Type RAW_CRC is a result of computation of CRC-32 checksum in accordance with the followings.

```
// Polynomial used: 0x04C11DB7L
const long lCRCTable[256]= {
 0x00000000L, 0x04C11DB7L, 0x09823B6EL, 0x0D4326D9L,
 0x130476DCL, 0x17C56B6BL, 0x1A864DB2L, 0x1E475005L,
 0x2608EDB8L, 0x22C9F00FL, 0x2F8AD6D6L, 0x2B4BCB61L,
 0x350C9B64L, 0x31CD86D3L, 0x3C8EA00AL, 0x384FBDBDL,
 0x4C11DB70L, 0x48D0C6C7L, 0x4593E01EL, 0x4152FDA9L,
 0x5F15ADACL, 0x5BD4B01BL, 0x569796C2L, 0x52568B75L,
 0x6A1936C8L, 0x6ED82B7FL, 0x639B0DA6L, 0x675A1011L,
 0x791D4014L, 0x7DDC5DA3L, 0x709F7B7AL, 0x745E66CDL,
 0x9823B6E0L, 0x9CE2AB57L, 0x91A18D8EL, 0x95609039L,
 0x8B27C03CL, 0x8FE6DD8BL, 0x82A5FB52L, 0x8664E6E5L,
 0xBE2B5B58L, 0xBAEA46EFL, 0xB7A96036L, 0xB3687D81L,
 0xAD2F2D84L, 0xA9EE3033L, 0xA4AD16EAL, 0xA06C0B5DL,
 0xD4326D90L, 0xD0F37027L, 0xDDB056FEL, 0xD9714B49L,
 0xC7361B4CL, 0xC3F706FBL, 0xCEB42022L, 0xCA753D95L,
 0xF23A8028L, 0xF6FB9D9FL, 0xFBB8BB46L, 0xFF79A6F1L,
 0xE13EF6F4L, 0xE5FFEB43L, 0xE8BCCD9AL, 0xEC7DD02DL,
 0x34867077L, 0x30476DC0L, 0x3D044B19L, 0x39C556AEL,
 0x278206ABL, 0x23431B1CL, 0x2E003DC5L, 0x2AC12072L,
 0x128E9DCFL, 0x164F8078L, 0x1B0CA6A1L, 0x1FCDBB16L,
 0x018AEB13L, 0x054BF6A4L, 0x0808D07DL, 0x0CC9CDCAL,
 0x7897AB07L, 0x7C56B6B0L, 0x71159069L, 0x75D48DDEL,
 0x6B93DDDBL, 0x6F52C06CL, 0x6211E6B5L, 0x66D0FB02L,
 0x5E9F46BFL, 0x5A5E5B08L, 0x571D7DD1L, 0x53DC6066L,
 0x4D9B3063L, 0x495A2DD4L, 0x44190B0DL, 0x40D816BAL,
 0xACA5C697L, 0xA864DB20L, 0xA527FDF9L, 0xA1E6E04EL,
 0xBFA1B04BL, 0xBB60ADFCL, 0xB6238B25L, 0xB2E29692L,
 0x8AAD2B2FL, 0x8E6C3698L, 0x832F1041L, 0x87EE0DF6L,
 0x99A95DF3L, 0x9D684044L, 0x902B669DL, 0x94EA7B2AL,
 0xE0B41DE7L, 0xE4750050L, 0xE9362689L, 0xEDF73B3EL,
 0xF3B06B3BL, 0xF771768CL, 0xFA325055L, 0xFEF34DE2L,
 0xC6BCF05FL, 0xC27DEDE8L, 0xCF3ECB31L, 0xCBFFD686L,
 0xD5B88683L, 0xD1799B34L, 0xDC3ABDEDL, 0xD8FBA05AL,
 0x690CE0EEL, 0x6DCDFD59L, 0x608EDB80L, 0x644FC637L,
 0x7A089632L, 0x7EC98B85L, 0x738AAD5CL, 0x774BB0EBL,
 0x4F040D56L, 0x4BC510E1L, 0x46863638L, 0x42472B8FL,
 0x5C007B8AL, 0x58C1663DL, 0x558240E4L, 0x51435D53L,
 0x251D3B9EL, 0x21DC2629L, 0x2C9F00F0L, 0x285E1D47L,
 0x36194D42L, 0x32D850F5L, 0x3F9B762CL, 0x3B5A6B9BL,
 0x0315D626L, 0x07D4CB91L, 0x0A97ED48L, 0x0E56F0FFL,
 0x1011A0FAL, 0x14D0BD4DL, 0x19939B94L, 0x1D528623L,
 0xF12F560EL, 0xF5EE4BB9L, 0xF8AD6D60L, 0xFC6C70D7L,
 0xE22B20D2L, 0xE6EA3D65L, 0xEBA91BBCL, 0xEF68060BL,
```

```
  0xD727BBB6L, 0xD3E6A601L, 0xDEA580D8L, 0xDA649D6FL,
  0xC423CD6AL, 0xC0E2D0DDL, 0xCDA1F604L, 0xC960EBB3L,
  0xBD3E8D7EL, 0xB9FF90C9L, 0xB4BCB610L, 0xB07DABA7L,
  0xAE3AFBA2L, 0xAAFBE615L, 0xA7B8C0CCL, 0xA379DD7BL,
  0x9B3660C6L, 0x9FF77D71L, 0x92B45BA8L, 0x9675461FL,
  0x8832161AL, 0x8CF30BADL, 0x81B02D74L, 0x857130C3L,
  0x5D8A9099L, 0x594B8D2EL, 0x5408ABF7L, 0x50C9B640L,
  0x4E8EE645L, 0x4A4FFBF2L, 0x470CDD2BL, 0x43CDC09CL,
  0x7B827D21L, 0x7F436096L, 0x7200464FL, 0x76C15BF8L,
  0x68860BFDL, 0x6C47164AL, 0x61043093L, 0x65C52D24L,
  0x119B4BE9L, 0x155A565EL, 0x18197087L, 0x1CD86D30L,
  0x029F3D35L, 0x065E2082L, 0x0B1D065BL, 0x0FDC1BECL,
  0x3793A651L, 0x3352BBE6L, 0x3E119D3FL, 0x3AD08088L,
  0x2497D08DL, 0x2056CD3AL, 0x2D15EBE3L, 0x29D4F654L,
  0xC5A92679L, 0xC1683BCEL, 0xCC2B1D17L, 0xC8EA00A0L,
  0xD6AD50A5L, 0xD26C4D12L, 0xDF2F6BCBL, 0xDBEE767CL,
  0xE3A1CBC1L, 0xE760D676L, 0xEA23F0AFL, 0xEEE2ED18L,
  0xF0A5BD1DL, 0xF464A0AAL, 0xF9278673L, 0xFDE69BC4L,
  0x89B8FD09L, 0x8D79E0BEL, 0x803AC667L, 0x84FBDBD0L,
  0x9ABC8BD5L, 0x9E7D9662L, 0x933EB0BBL, 0x97FFAD0CL,
  0xAFB010B1L, 0xAB710D06L, 0xA6322BDFL, 0xA2F33668L,
  0xBCB4666DL, 0xB8757BDAL, 0xB5365D03L, 0xB1F740B4L
};

long computeCRC( const unsigned char *d, int n )
{
  const long *table= lCRCTable;
  long crc=         0;
  long table_mask=   0xFFFFFFFFL;

  while(n--)
    crc = ((crc << 8) ^
          table[int((crc >> 24) ^ *d++)]) & table_mask;
  return crc;
}
```

APPENDIX A3. Structure of ephemerids messages

There are two types of NTL GNSS ephemeris messages. The first message contains unified Keplerian elements for GPS, Galileo, BeiDou and IRNSS. The second message (TBD) includes GLONASS ephemeris data.

Table A3.1 Unified KEPLEPH_BLOCK structure

#Offset, byte	Format	Description	Notice			
			GPS	Galileo	BeiDou	IRNSS
0x00	uint32	Auxiliary data	gpsAuxData	galAuxData	bdoAuxData	irnAuxData
0x04	uint32	TOW	Scaled to seconds			
0x08	uint32	toc				
0x0c	uint32	toe				
0x10	int32	af0				
0x14	int32	af1				
0x18	int16	af2				
0x1a	uint16	(WN)	Week Number			
0x1c	uint16	IODC		IODC=IODE= IODnav	AODC	IODC=IODE
0x1e	uint16	IODE			AODE	
0x20	int32	M0				
0x24	int32	e				
0x28	uint32	sqrt(A)				
0x2c	int32	Ω0				
0x30	int32	i0				
0x34	int32	ω				
0x38	int32	dΩ/dt				
0x3c	int32	Δn				
0x40	int32	Cuc				
0x44	int32	Cus				
0x48	int32	Crc				
0x4c	int32	Crs				
0x50	int32	Cic				
0x54	int32	Cis				
0x58	int16	IDOT				
0x5a	int16	TGD1	TGD1= TGD2	BGD(E1,E5a)	TGD1	TGD1= TGD2
0x5c	int16	TGD2		BGD(E1,E5b)	TGD2	
0x5e	uint8	URA				
0x5f	uint8	SV ID	See D012			

Total size, bytes: 96.
Signed fields have 2th complement representation. Sign bit extension has been executed.
GNSS system dependent scaling values should be applied additionally before usage.

Table A3.2 Structure gpsAuxData

#OFFSET, bits	Format	Name	Description
0	1 bit	Fit interval	
1	1 bit	AS mode	
2	uint2	Code on L2	
4	uint1	L2 P data flag	
5	1 bit	Alert	
6	uint6	SV Health	

Table A3.3 Structure galAuxData

#OFFSET, bits	Format	Name	Description
0	uint2	E1B_HS	
2	uint2	E5B_HS/E5A_HS	E5A for Fnav, E5B for Inav
4	uint1	E1B_DVS	
5	uint1	E5B_DVS/ E5A_DVS	E5A for Fnav, E5B for Inav
6	int10	BGDE5A/E1	
...			
...			
29	uint2	Data_flow	0 – Inav, E1 1 – Inav, E5b 2 – Fnav
31	uint1	Integral validity	

Table A3.4 Structure bdoAuxData

#OFFSET, bits	Format	Name	Description
0		Reserved	
..			
31	Uint1	Health_flag	1-healthy, 0-unhealthy, (reversed ephemeris value)

Table A3.5 Structure irnAuxData

#OFFSET, bits	Format	Name	Description
0	1 bit	Alert	
2	1 bit	L5 flag	
3	1 bit	S flag	
4	1 bit	Autonav	
5	2 bits	2 spare bits after IDOT	
7	2 bits	2 spare bits after I0	
...			
31	uint1	Integral validity	

Table A3.6 GLOEPH_BLOCK structure

#Offset	Format	Description	Notice
0x00	uint32	gloAuxData	
0x04	int32	Xn	
0x08	int32	Yn	
0x0c	int32	Zn	
0x10	int32	dXn/dt	
0x14	int32	dYn/dt	
0x18	int32	dZn/dt	
0x1c	int32	τn	
0x20	int32	τc	
0x24	int32	τGPS	
0x28	uint16	NT	
0x2a	uint16	NA	
0x2c	uint16	tk	
0x2e	uint16	tb	
0x30	int16	γn	
0x32	int8	d2Xn/dt2	
0x33	int8	d2Yn/dt2	
0x34	int8	d2Zn/dt2	
0x35	int8	Δτn	
0x36	uint8	En	
0x37	int8	Frequency channel number	
0x38	uint8	n	
0x39	uint8	SV ID	See D012
Total size, bytes: 58. Signed fields have 2th complement representation. Sign bit extension has been executed. GNSS system dependent scaling values should be applied additionally before usage.			

Table A3.7 Structure gloAuxData

#OFFSET	Format	Name	Description
0	uint2	P1	
2	1 bit	P2	
3	1 bit	P3	
4	1 bit	P4	
5	1 bit	Cn	SV health from almanac data
6	1 bit	Cn availability	If this bit is set to 1, Cn is available
7	uint3	Bn	
10	uint2	P	
12	uint2	M	
14	1 bit	ln (3rd string)	
15	1 bit	ln (5th string)	
16	uint4	FT	
20	uint5	N4	
25	7 bits	Reserved	

APPENDIX B. NOVATEL RAW DATA MESSAGES

The set of the supported Novatel messages is given in Table B.1. The description of the format can be found on the company's website www.novatel.com (see document OM-20000169 v3).

Table B.1 – Novatel messages

Message ID	Message name	Message description
43	RANGEB	Ranging measurements
140	RANGECMPB	Ranging measurements, packed format
42	BESTPOSB	Receiver coordinates
723	GLOEPHEMERISB	Ephemeris of the GLONASS satellites
7	GPSEPHEMB	Ephemeris of the GPS satellites
719	GLOCLOCKB	Calendar data and GLONASS time scale data
1696	BDSEPHEMERISB	Ephemeris of the BeiDou satellites
1309	GALINAVEPHB	Ephemeris of the Galileo satellites
2123	IRNEPHEMERISB	Ephemeris of the NavIC satellites

APPENDIX C. EXTENSION OF NMEA 0183 SENTENCES

Table C.1 – Sentence HDG

$--HDG,hhmmss.ss,yyy.yyyy,pp.pppp,rr.rrrr,n,aaa.a,q*hh<CR><LF>	
hhmmss.ss	UTC of position
yyy.yyyy	Yaw, in the range of 0.0 - 360.0 degrees
ppp.pppp	Pitch, in the range of -90.0 - 90.0 degrees
rrr.rrrr	Roll, in the range of -90.0 - 90.0 degrees
n	Total number of satellites that were used for computing Yaw/Pitch/Roll parameters
a.a	Indicator of ambiguity resolution in RTK mode (ARF)
q	GNSS quality indicator: 0 = Fix not available or invalid 1 = GPS SPS Mode, fix valid 2 = Differential GPS, SPS Mode, fix valid 3 = GPS PPS Mode, fix valid 4 = Real Time Kinematic. System used in RTK mode with fixed integers 5 = Float RTK. Satellite system used in RTK mode, floating integers 6 = Estimated (dead reckoning) Mode 7 = Manual Input Mode 8 = Simulator Mode

Table C.2 – Sentence PELE110. Indian regional time (structure is equal to xxZDA)

$PELE110,hhmmss.ss,dd,mm,yyyy,zh,zm*hh<CR><LF>	
hhmmss.ss	Indian regional time
dd	Day
mm	Month
yyyy	Year
zh	Local zone hours, always 0.
zm	Local zone minutes, always 0.
hh	Checksum.

Table C.3 – Sentence PELE111. Message from GAGAN Messaging Service (MT63).

$PELE111,nnn,ttt,m1m2..m53*hh<CR><LF>	
nnn	Source satellite number
ttt	Tag - sequential number of incoming message, range 1..255.
m_1 .. m_{53}	mi – hex character '0'..'f' encoding 4 sequential bits of the message. 53 characters encode full 212 bit data message. $m_1=\{b_1b_2b_3b_4\}$, $m_{53}=\{b_{209}b_{210}b2_{211}b_{212}\}$, where b_1- most significant (first coming bit), b_{212} – least significant (last coming) bit.
hh	Checksum.

Table C.4 – Sentence PELE112. Message from NavIC Messaging Service (IRNSS-A1 sat.).

$PELE112,nnn,ttttt,f,id,$m_1m_2..m_{55}$*hh<CR><LF>	
nnn	Source satellite number, always 1.
ttttt	TOW (Time of Week) extracted from corresponding subframe.
f	"MESSAGING ENABLE" flag
id	Message ID
m_1 .. m_{55}	mi – hex character '0'..'f' encoding 4 sequential bits of the message. 55 characters encode full 220 bit data message. $m_1=\{b_1b_2b_3b_4\}$, $m_{55}=\{b_{217}b_{218}b2_{219}b_{220}\}$, where b_1- most significant (first coming) bit, b_{220} – least significant (last coming) bit.
hh	Checksum.

Table C.5 – Sentence PELE113. Message from NavIC MT18 subframe.

$PELE113,nnn,id,xx,yy,$c_1c_2..c_N$*hh<CR><LF>	
nnn	Source satellite number.
id	Message Identifier
xx	Total number of text blocks in the message
yy	Number of text block in the message
c_1 .. c_N	Length N (N<25) textual string.
hh	Checksum.

APPENDIX D. STRUCTURE OF THE HEADER OF FIRMWARE BINARY FILE

#Offset	Field	Length	Description
...			Service information
0x0028	Major_FW	1	Data, describing hardware and firmware. Actual for firmware versions earlier than 6.60. For elder versions this fields contain fixed values: Major_FW = 6; Minor_FW = 60; FRIC_ID = 1; BaseBand_ID = 4; FLASH_ID = 7.
0x0029	Minor_FW	1	
0x002a	RFIC_ID	1	
0x002b	BaseBand_ID	1	
0x002c	FLASH_ID	1	
0x002d	Reserved	1	
0x002e	Reserved	1	
0x002f	Reserved	1	
...			Service information
0x0048	DeviceAtrPointer	4	Zero-based address of DeviceAttributesTable
...			Service information
0x0060	0x05111983	4	BootHeader marker
...			Service information
[DeviceAtrPointer] + 0	Primary_ID	4	Device unique identifier
[DeviceAtrPointer] + 4	Secondary_ID	4	Device unique identifier
[DeviceAtrPointer] + 8	MinorBldrVer	1	Boot loader version
[DeviceAtrPointer] + 9	MajorBldrVer	1	
[DeviceAtrPointer] +10	Reserved	1	
[DeviceAtrPointer] +11	Reserved	1	
[DeviceAtrPointer] +12	MajorHW	1	Major hardware type
[DeviceAtrPointer] +13	MinorHW	1	Minor hardware type
[DeviceAtrPointer] +14	MajorFW	1	Major firmware version
[DeviceAtrPointer] +15	MinorFW	1	Minor firmware version
[DeviceAtrPointer] +16	OptionsFW	1	Firmware build options
[DeviceAtrPointer] +17	BuildDate	11	A string, containing date
[DeviceAtrPointer] +28	–	–	Service information
...			

CONTACTS

For complete contact information visit us at **www.ntlab.lt**

Head Office
4th floor, 41 Surganova str., 220013 Minsk, Republic of Belarus
Tel.: +375 17 290 09 99
Fax: +375 17 290 98 98
e-mail: ntlab@ntlab.com, sales@ntlab.com

EU Branch Office
NTLAB, UAB
Švenčionių g. 112, Nemenčinė, LT-15168 Vilniaus r., Lithuania
Tel.: +370 6 169 5418
e-mail: sales@ntlab.lt