Navtelecom Communication Protocol

Version 6.2

Moscow
2021

Table of contents

History of changes

version 5.0:
- removed description of dynamic change of data transfer mask by the server for FLEX;
- removed description of dynamic change of data transfer mask by the server for FLEX;
- added description of FLEX 2.0
- added description of work with autoinformer;
- added description of work with driver display;
- added description of work with tachograph;
- added description of work with digital camera;
- added description of SMS device configuration.

version 5.1:
- fixed a bug in the description of the field " Status of GPS/GLONASS navigation sensor".

version 5.2:
- fixed bugs in the description of commands, requests and responses when working with tachograph;
- fixed bugs in the description of commands, requests and responses when working with camera;
- updated structure of FLEX telemetry records (Annex A.1).

version 5.3:
- fixed bugs in the description of the structure of FLEX telemetry records (Annex A.1), fields No.: 4, 5, 6, 53, 66, 68, 72, 122;
- types of variables in the description are corrected to standard names;
  - corrected the field order in the "FLEX 2.0 additional telemetry packet example" table.
- added description of the NTCB and NTCT commands "Command to resend telemetry from non-volatile memory";
- added a description of the NTCB and NTCT commands "Command to resend telemetry from SD card."

version 5.4:
- updated description of the structure of FLEX telemetry records (Appendix A.1), fields No.: 4, 6;
- made clarifications to the commands: "Command to resend telemetry from non-volatile memory", "Command to resend telemetry from SD card";
- added SMS command to confirm synchronization of the internal memory with the server;
- added "Format of standard SMS message M: 101";
- added arbitrary USSD NTCT and NTCB requests;
- added sections 2.14, 4.2.12 "Working with built-in accelerometer";
- added section 2.15 "Accident detection";
- changed format of the SMS command "SETTLE";
- added description of the SMS query "POS";
- changed formats of SMS commands and requests "SET" and "GET";
- added query "*?ERFT";
- added section "5. Tone control";
- adeed " Format of file that stores accident information" to the section "2.15 Accident detection";
- added "Annex E. Table of digital FLS error codes";
- added "Command of arbitrary sound indication by buzzer" to the section "2.1 System commands, requests, messages";
- added "Standard SMS format M:112" to the section "4.1. Formats of SMS messages coming from device";
- added "Query location-based service (LBS) information" to the section "4.2.4. Telemetry information";
- added "Format of standard SMS message M:113" to the section "4.1. Formats of SMS messages coming from device";

- added "Query information about firmware version of the GPS receiver" to the section "4.2.1. System queries and commands";
- updated "Annex D. Summary of NTCB messages".

version 5.5:
- updated field "Status of functional modules 2" in the section "Annex A.1. Structure of FLEX telemetry records";
- added new type of the field to the section "Structure of the dynamic part of the FLEX 2.0 additional packet"
- added "Command to change SIM card" to the section "2.1. System commands, queries, messages";
- added "Command to change SIM card" to the section "4.2.1. System queries and commands";
- added "Custom command RS485/RS232" to the section "2.12. Communication between external interfaces and server";
- added section "4.2.13. Communication between external interfaces";
- added "Query device status with SMS sending" to the section "2.1. System commands, queries, messages";
- added "Standard SMS format M:114" to the section "4.1. Formats of SMS messages coming from device";
- added "Query device status" to the section "4.2.1. System queries and commands";
- added "Query unique SIM serial number" to the section "2.1. System commands, queries, messages";
- added "Query unique SIM serial number" to the section "4.2.1. System queries and commands";
- added new error code to the section "Appendix E. Table of digital FLS error codes".

version 5.6:
- added "Query information about firmware version of the GPS receiver" to the section "2.1. System commands, queries, messages";
- updated "Appendix D. Summary of NTCB messages";

version 5.7:
- "Custom command RS485/RS232" renamed to "Custom command";
- added GPS interface for "*! UC." NTCB and NTCT commands.

version 5.8:
- removed special value 0xFFFFFFFF from the description of the request for a DDD file block in the section "2.8.2. Generating and transferring DDD file to the server";
- removed special value 0xFFFF from the description of the command "Query photo data" and specified the maximum size of the transmitted block;
- added "Command to reset current calibration" to the section "4.2.12. Working with built-in accelerometer";
- fixed errors in the description of parameter No. 77 of the FLEX structure;
- changed description of parameter No. 108 of the FLEX structure.

version 6.0:
- updated table with description of the negotiation protocol version packet description in the "1.2.1. Basic procedure for establishing connection with a server via FLEX protocol";
- updated description of the current state packet in the section "1.2.1. Basic procedure for establishing connection with a server via the FLEX protocol";
- updated tables "List of supported FLEX protocol versions" and "List of supported FLEX data structure" in the section "1.2.1. Basic procedure for establishing connection with a server via the FLEX protocol";
- updated tables "List of supported FLEX protocol versions" and "List of supported FLEX data structure" in the section "1.2.2. List of FLEX versions and basic messages";
- changed command formats:
  "Arbitrary USSD request",

"Command of arbitrary sound indication by buzzer",
"Command to change SIM card",
"Query device status with SMS sending",
"Query unique SIM serial number",
" Query information about firmware version of the GPS receiver" in the section "2.1. System commands, queries, messages";

- added description of command format
"Command for microphone listening" to the section "2.1. System commands, queries, messages";
- added description of command format
"Command to change output state" to the section "2.2. Outputs control";
- changed command formats:
"Command to lock input",
"Command to unlock input" in the section "2.3. Inputs control";
- changed command formats:
"Command to confirm synchronization of internal memory with the server",
"Command to resend telemetry from non-volatile memory",
"Command to resend telemetry from SD card" in the section "2.4. Telemetry commands, queries, messages";
- added command formats:
"Command to change operating mode of the device",
"Query current operating mode" to the section "2.5 Device operating modes control";
- updated command formats:
"Command to connect to the configurator via RCS",
"Command to update device firmware via RFU" in the section "2.6. Connection to RCS, RFU, DRC services";
- changed command format "Custom command" in the section "2.12. Communication between external interfaces and server";
- added description of the command format "Query CAN-LOG device program number" to the section "2.13. Working with CAN-LOG module";
- updated command "Accelerometer calibration command" in the section "2.14 Working with built-in accelerometer";
- changed description of the accident file format in the table "Format of file that stores accident information" in the section "2.15. Accident detection";
- updated the tables:
"Queries"
"Commands" in the section "4.2.1 System queries and commands";
- updated tables:
"Queries"
"Command to resend telemetry from non-volatile memory",
"Command to resend telemetry from SD card" in the section "4.2.2. Telemetry information";
- updated table "Commands" in the section "4.2.3. Outputs";
- added table description of the new command format in the section "4.2.5. Services RCS, RFU";
- removed section "4.2.6. Subscribers", changed section numeration of clause 4.2, starting from clause 4.2.6 (inclusive);
- updated tables:
"Commands"
"Requests" in the section "4.2.6. Device operating modes";
- updated table "Commands" in the section "4.2.11. Working with built-in accelerometer";
- updated table "Commands" in the section "4.2.12. Communication between external interfaces";
- added description of the new command format for configuration in the section "4.3 SMS configuration";
- updated fields 4, 6, 38-44, 78-93, 100, 109 for FLEX1 and FLEX2 protocols in the Annex "A.1. Structure of FLEX telemetry records";
- added description of FLEX3 fields format in the Annex "A.1. Structure of FLEX telemetry records";

- added Annex “A.4. Compressor configurations and conditions for ThermoKing units”;
- updated Annex "G. Summary of NTCB messages”.

version 6.1:
- all tables with descriptions of messages or commands were added with information on which communication channels a message or command can be transmitted;
- added examples to some of the tables with descriptions of messages or commands;
- made minor corrections on the entire text of the document;
- fixed a typo "Command for microphone listening" in the section “2.1. System commands, queries, messages";
- added description “Command for time adjustment on device” to the section “2.1. System commands, queries, messages";
- added description “Command for diagnostic information query” to the section “2.1. System commands, queries, messages";
- added description "Response to unsupported messages" to the section “2.1. System commands, queries, messages";
- added description “Query current state of the device in the additional packet” to the section “2.4 Telemetry commands, queries, messages”;
- fixed a typo “Command to change device operating mode” in the section “2.5 Device operating modes control”;
- added description “Command to connect to DRC” to the section “2.6. Connection to RCS, RFU, DRC services”;
- renamed section “2.7. Working with identifiers”, made minor corrections in the text;
- added description “Command to edit the list of registered Touch Memory keys” in the section “2.7. Working with identifiers”;
- fixed description of the FLEX command format for editing registered TM keys in the section “2.7. Working with Identifiers”;
- fixed description “Command to start DDD file generation” and updated description of the logic for downloading a file from tachograph and the logic of working with file cache in the section “Generating and transferring DDD file to the server";
- added section “2.10.1. Universal commands for autoinformer";
- updated description in the section “2.11.2. Transferring photos to the server”;
- updated description in the section “2.12. Communication between external interfaces and server”;
- updated description of the unloading process and commands in the section “2.15. Accident detection”;
- added section “2.16. Working with 1-Wire temperature sensors”;
- added section “2.17. Working with passenger flow counters”;
- added section “2.18. Working with refrigeration units;
- added section “2.19. Configuration commands»;
- updated description in the section “4.2 SMS requests and commands”, updated style of description of all SMS commands and requests;
- added description “Query unique device ID” to the section “4.2.1. System queries and commands”;
- added description “Command of arbitrary sound indication by buzzer” to the section “4.2.1 System queries and commands”;
- added description “Command for time adjustment on device” to the section “4.2.1. System queries and commands”;
- added description " Command for diagnostic information query " to the section “4.2.1. System queries and commands”;
- fixed a typo “Command to switch security modes” in the section “4.2.5 Device operating modes”;
- added description "Command to connect to DRC" to the section “4.2.6. RCS, RFU, DRC services";
- added section “4.2.7. Working with identifiers";
- added description of “Command to change the route” to the section “4.2.10 Autoinformer”;
- added description “Command to playback audio file” to the section “4.2.10 Autoinformer”;

- added description "Command to change autoinformer operating mode" to the section "4.2.10 Autoinformer";
- added description "Command to change the current stop" to the section "4.2.10 Autoinformer";
- added section "4.3.1. EDITS and READ command formats";
- updated description in the section "4.3.2 SET and GET command formats";
- added section "6. Working with device configuration";
- added "Navigation Antenna Status" to the field #123 in the section "Annex A.1. Structure of FLEX telemetry records";
- added explanation regarding the direction of roll and pitch to the field #141 in the section Annex A.1. Structure of FLEX telemetry records";
- changed parameter types: S8 to I8, S16 to I16, S32 to I32, S64 to I64 in the fields #207…255 in the section Annex A.1. Structure of FLEX telemetry records";
- added information about new messages to the section "Annex D. Summary of NTCB messages".

version 6.2:

- added description of the standard SMS M:115 (format of response to a diagnostic request with a list of GSM error codes) to the section "4.1. Formats of SMS messages coming from device"

Notation conventions and accepted abbreviations

Designation	Interpretation
AES128 (Advanced Encryption Standard)	symmetric block encryption algorithm, block size 128 bits.
ASCII (American standard code for information interchange)	American character encoding table.
CAN (Controller Area Network)	industrial network standard, focused primarily on combining various executive devices into a single network.
COM (Communications Port)	RS-232 interface.
GPRS (General Packet Radio Servic)	public packet radionetwork.
Handshake	packet consisting of a message with an individual device ID string.
GPS (Global Positioning System)	satellite navigation system, which provides measurement of distance, time and determines location in the WGS 84 world coordinate system.
GSM (Global System for Mobile communications)	a global digital standard for mobile cellular communications.
ID (Identifier)	identifier.
IP (Internet Protocol)	routable protocol of the TCP/IP stack network layer.
IMEI (International Mobile Equipment Identity)	international mobile equipment identifier.
NTCB (Navtelecom Binary)	binary communication protocol of Navtelecom LLC.
NTCT (Navtelecom Text)	text communication protocol of Navtelecom LLC.
RCS (Remote Configuration Service)	remote configuration service.
RFU (Remote Firmware Update)	remote firmware update service.
RS-232 (Recommended Standard 232)	standard describing a serial communication interface supporting asynchronous communication.
RS-485 (Recommended Standard 485)	standard for transmitting data over a two-wire half-duplex multipoint serial symmetrical communication channel.
SMS (Short Messaging Service)	technology that allows receiving and transmitting short text messages over mobile radiotelephone networks.
TCP (Transmission Control Protocol)	network data transfer protocol.
TCP/IP (Transmission Control Protocol / Internet Protocol)	set of network data protocols.
UDP (User Datagram Protocol)	user datagram protocol.
USB (Universal Serial Bus)	universal serial wire bus.
UTC (Coordinated Universal Time)	international time format.
WGS-84 (World Geodetic System 1984)	world geodetic coordinate system 1984
XOR	boolean exclusive-OR function.
GLONASS	global navigation satellite system of the Russian Federation.
PZ-90.11	state geocentric coordinate system "Land parameters 1990."
PC	personal computer.
soft	software.

Notation conventions

Following designations are used in this document:

- I8 - signed integer with a length of 1 byte (character);
- I16 - signed integer with a length of 2 bytes;
- I32 - signed integer with a length of 4 bytes;
- I64 - signed integer with a length of 8 bytes;
- U8 - unsigned integer with a length of 1 byte;
- U16 - unsigned integer with a length of 2 bytes;
- U32 - unsigned integer with a length of 4 bytes;
- U64 - unsigned integer with a length of 8 bytes;
- Char - character type with a length of one byte.

For all specified integer types, there is little-endian bytes order.

The following types of records can be used to designate an array:

1) 16*U8 - array of 16 unsigned integer bytes;
2) char[7] - array of 7 characters.

To designate a string of arbitrary size without null-terminated, used following record type: char[].

Introduction

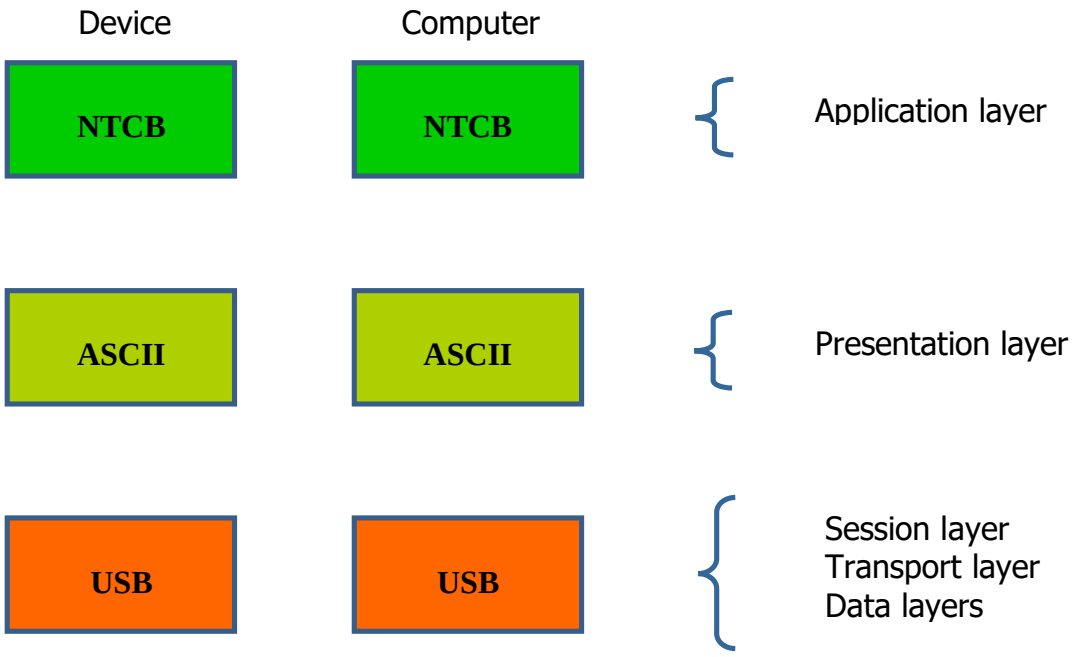
In telematic devices (Signal and Smart series) manufactured by Navtelecom LLC, there are two protocols used for all options for transmitting telematic information via communication channels:
- NTCB binary (Navtelecom Binary) protocol with FLEX extension;
- NTCT text (Navtelecom Text) protocol.

NTCT text (symbolic) protocol is used to transmit telemetry through the SMS service of cellular operators. Packets of this protocol are limited by the length of one SMS message (140 characters) and include the main telematic information about the monitored object.

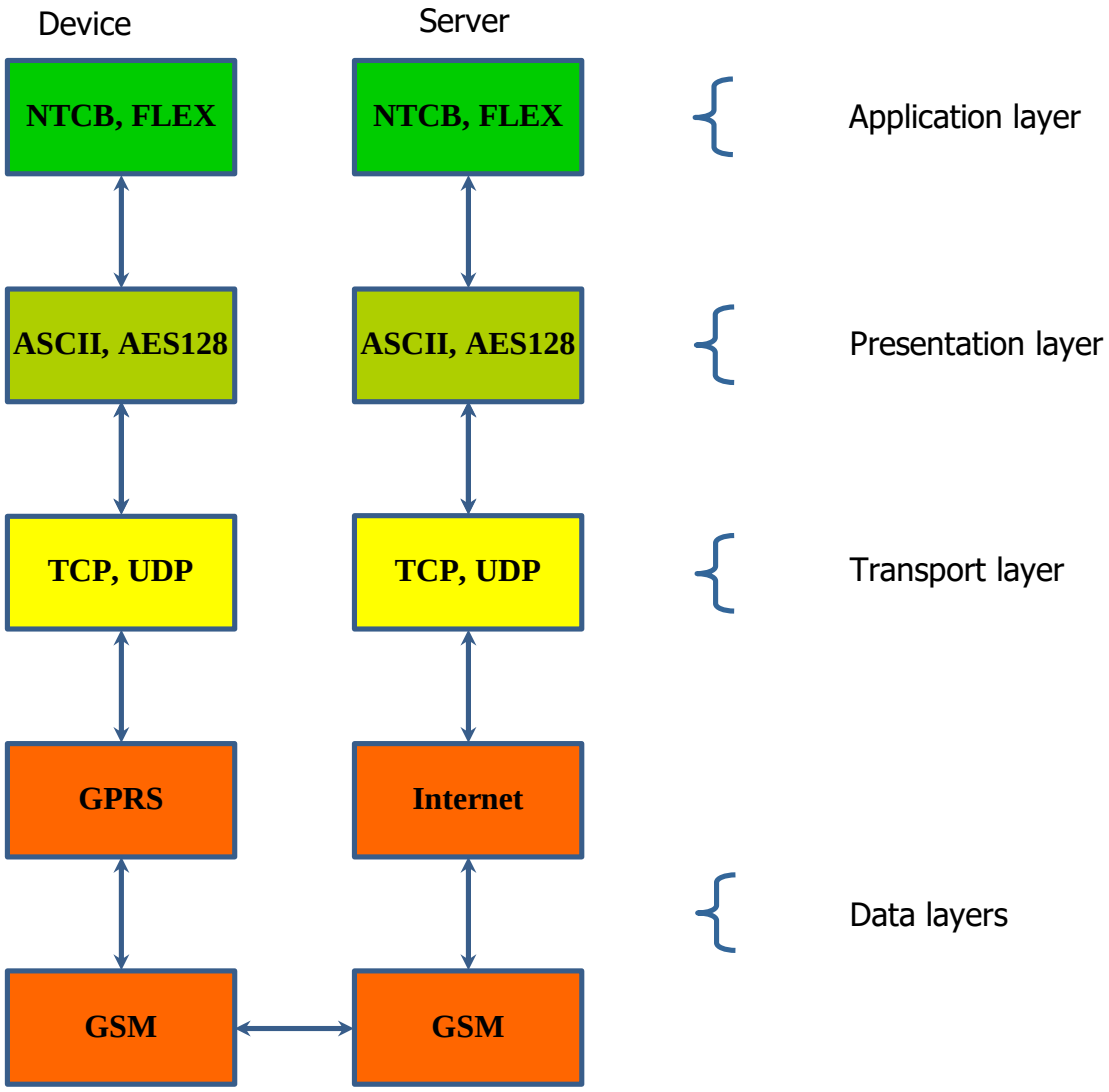
To transmit complete information about the device, change settings and internal firmware, binary NTCB protocol for data exchange via USB, GPRS is used. NTCB protocol is divided into application and transport layers.

When exchanging over GPRS, device supports various options for presentation and transport layers.
Presentation layers: 1) ASCII - data without encryption using the corresponding encoding table; 2) AES128 - data encrypted by a symmetric block encryption algorithm.
Transport layers: 1) TCP - communication protocol, provides data transfer in networks and subnets TCP/IP; 2) UDP - communication protocol for data transmission in IP networks without connection. Operation via UDP is no different from TCP in terms of device interaction with the server.

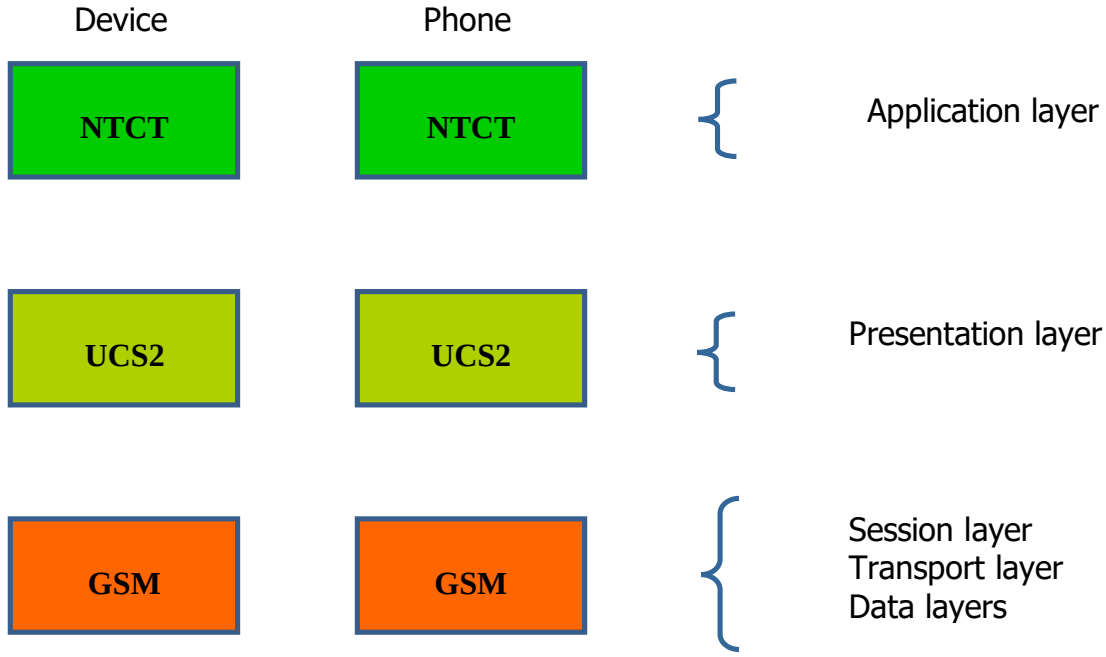
Device-Computer interaction diagram



Device-Server interaction diagram



Device-GSM phone (SMS) interaction diagram



1. NTCB binary protocol and its FLEX extension

NTCB is a basic protocol with static telemetry packets. Static refers to the structures of telemetry records: F1, F2, F3, F4, F5, F5.1, F5.2, F6. At the moment, the part of the protocol related to static telemetry messages is not used, FLEX protocol extension is used. FLEX protocol is a set of telemetry messages that allows flexible customization of the transmitted information. FLEX also includes an extension of the set of commands and messages for working with the device and peripherals.

NTCB protocol allows packets with a maximum length of 65551 bytes. Each message within this protocol consists of two parts: ransport header and application layer data (telemetry information).

NTCB message structure

Protocol Fields	**<HEAD>** *(transport layer header)*	**<BODY>** *(application layer data)*
Fields size in bytes	16	from 0 to 65535

When device is connected to the server via GPRS, packets can be transmitted both at the initiative of the server and at the initiative of the device. However, the NTCB protocol is not full duplex, i.e. full transmission of packets in both directions is impossible at the same time. When working with telematic server, device initiates the transmission of telemetry packets. If there is a situation of simultaneous data transfer from the device and a command from the server, above exchange procedure may be violated. Initially, device executes a command or request and sends an acknowledgment to the server. After that, device proceeds to wait for confirmation of reception of telemetry information by the server. In some cases, device may miss incoming packet if at the same time it sends another packet (packet entering the device may be very delayed or not received at all). In this case, it is necessary, in the absence of reaction to the sent command, to provide for a repetition of the command. During active communication between server and device, it is necessary to send a command to the device instead of an acknowledgement to the telemetry message, then, after waiting for the command to be executed, it is necessary to send an acknowledgement or wait for the retransmission of the telemetry packet.

If there is no confirmation of receiving information from the server, device will repeat sending the telematics packet after pauses of 20 to 90 seconds, depending on the quality of the cellular network signal. It should also be noted that the indicated time intervals also depend not only on the signal quality, but also on the degree of availability of the GPRS channel.

During information exchange via USB interface, packet exchange begins exclusively at the initiative of the "host". For each packet from "host," the device sends an acknowledgement or response to the request. Before sending a new packet, "host" must wait for confirmation or take a pause. If the command or request does not meet the protocol requirements (incorrect type, broken structure, checksum does not match), confirmation of the reception of this command is not sent. Similarly, when sending incorrect telemetry data from a device when connecting over GPRS, server should not send back an acknowledgement of the reception of this data.

When device is connected to a personal computer via USB, device is defined as a virtual COM port. Waiting time for a response must not be less than a second. It is advisable to select the value of this pause equal to 1... 5 seconds. If there is no response, we recommend to try sending the packet to device again.

1.1. Transport layer of NTCB protocol

In the NTCB protocol, in the information packet, application layer data (commands, requests, acknowledgements and responses) is preceded by a 16-byte transport layer header.

Transport layer header structure

Bytes order	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
Header field	Preamble				Recipient ID IDr				Sender ID IDs				Number of data bytes n		CSd Data Checksum	CSp Header Checksum
Extension and format of the field	char[4]				U32				U32				U16		U8	U8
Default packet field values when it is sent from the device to the server without application data (n=0)	@NTC				1				0				0		0x00	0x18
	0x40 0x4e 0x54, 0x43				**0x01** 0x00 0x00 0x00				0x00 0x00 0x00 0x00				0x00 0x00		0x00	0x18
Default packet field values when it is sent from the server to the device without application data (n=0)	@NTC				0				1				0		0x00	0x18
	0x40 0x4e 0x54, 0x43				0x00 0x00 0x00 0x00				**0x01** 0x00 0x00 0x00				0x00 0x00		0x00	0x18

The first three fields (preamble, recipient ID, sender ID) serve to uniquely identify device and server when trying to establish a connection. Values of these fields are set at the stage of device configuration. If you do not configure these settings for the device, device will use default values for these fields.

The preamble consists of any four characters. By default, the first four characters of this packet form the string "@NTC." When the computer communicates with the device via USB interface, preamble is always "@NTC" regardless of the device settings.

"Host" (server) and device IDs are specified in the device settings. In case of transmitting a packet from host to device, packet recipient ID corresponds to the device ID and sender ID corresponds to the host ID. When sending a reverse packet (confirmation), identifiers **are replaced:** identifier of the packet recipient corresponds to the "host identifier and sender identifier corresponds to the device identifier.
To connect via USB and for devices in which these parameters are not set, default parameters are: host ID (PC) - 1, device ID - 0.

Number of data bytes indicates the number of bytes of this packet following this 16-byte header. Number of bytes cannot exceed 65535.

Checksums used in the header are calculated over the data entire length specified in the previous field using the XOR algorithmthe following function:

```
unsigned char xor_sum
(
      unsigned char     *buffer,     / * (in) pointer to the data buffer      */
      unsigned int      length       / * (in) number of bytes to count       */
)
{
      unsigned char temp_sum = 0;

      while ( length-- > 0 )
      {
            temp_sum ^= *buffer++;
      }

      return ( temp_sum );
}
```

First, data checksum Csd is calculated along the length of the data n.

Then checksum of the Csp header is calculated for the first 15 bytes: from the 1st to the 15th. Use this procedure to monitor the integrity of the transport layer header data.

If identifiers, preamble or checksums calculated on both sides do not match, packet is considered corrupted and no response messages are sent from the receiving side.

It is allowed to transmit an "empty" transport layer packet to maintain a communication channel consisting of 16 overhead bytes without application layer data. The receiving party should not send a response to this packet.

1.2. FLEX protocol. Working with telematics servers via GPRS

Working in FLEX protocol, preamble of the NTCB packet remains fixed, containing default value "@ NTC".

FLEX message headers contain a preamble starting with the character '~'. There are no sixteen-byte headers in these messages. To inform the server that the data transfer will be carried out in the FLEX protocol, additional protocol version negotiation message is sent.

To maintain communication (ping), a packet consisting of one byte with a fixed 0x7F value is used. Server confirmation for this packet is **not required**.

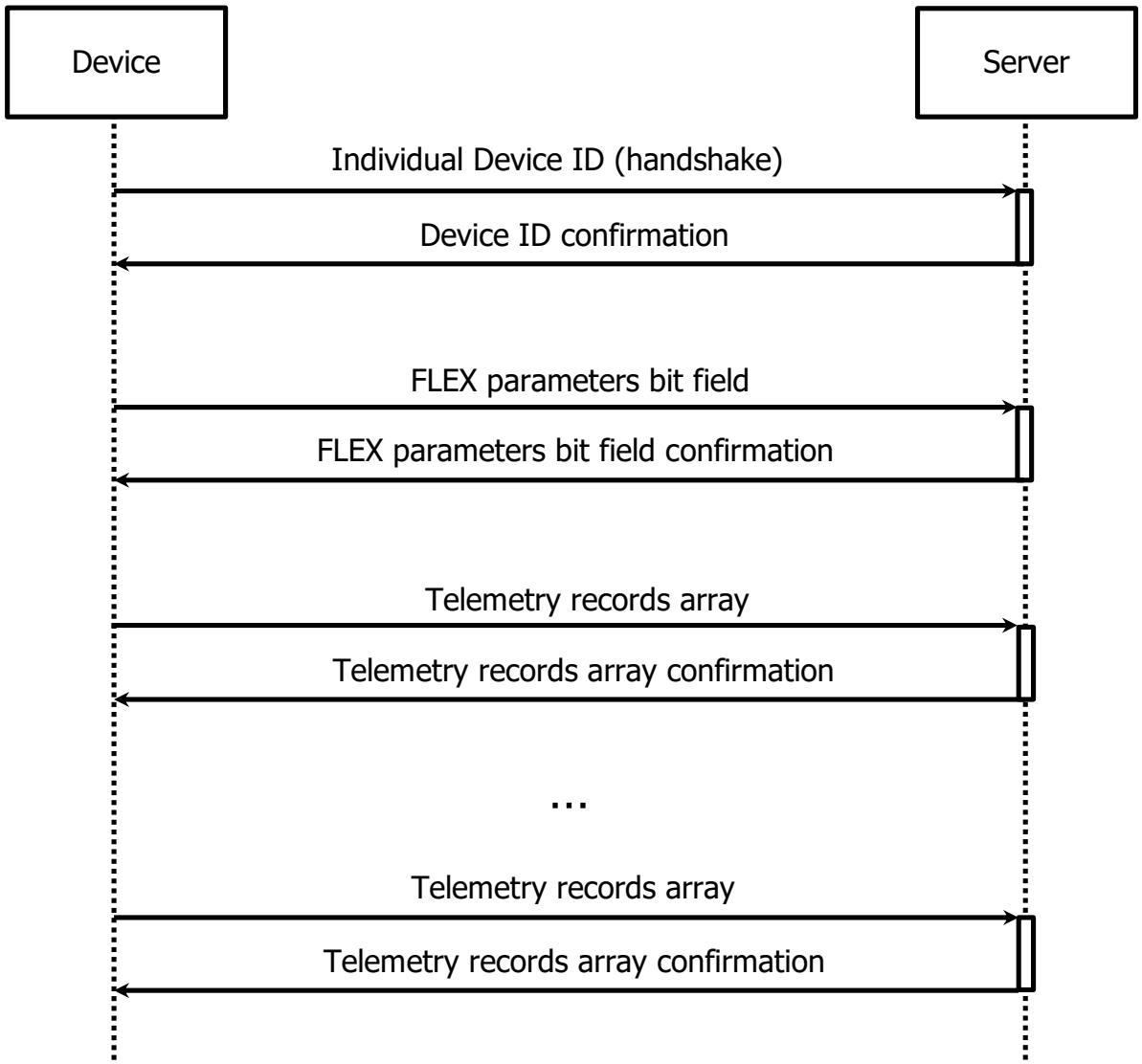
Thus, FLEX messages can be distinguished from other NTCB messages by the first character:
@ - NTCB message containing a 16-byte header;
~ - FLEX message;
DEL (0x7F) - FLEX ping message.

This protocol uses a universal telemetry configurable data table to transmit telemetry information. Each field of this table is indicated by a flag, they are transmitted to the server during the authorization process as a set of bits. If the value of the flag is "1," then the corresponding field will be transmitted, and if the flag is "0," then the field will not be transmitted. Fields are transmitted without gaps in case of their absence, there is offset to the end of the last recorded field.

1.2.1. Basic procedure for establishing connection with a server via FLEX protocol

When working via GPRS channel, device is always the initiator of establishing a connection with the server.

General case of communication with the server:



Communication with telematic servers is carried out according to the settings of the device itself. After opening a connection (socket), device independently sends a packet to the server, consisting of a message with a string of an individual device identifier - handshake. From this packet, it can be extracted the protocol settings: preamble, object ID and server ID, as well as the unique device ID (IMEI number of the GSM modem).

Parameters obtained during connection are compared with the settings available for this device and, based on the result, decision is made: to allow further work or to break connection. If the parameters match, response to this handshake packet is sent to the device, which informs the device that it is possible to start transmitting telemetry data.

After opening the connection (socket), device sends a handshake packet:

Message	<HEAD>*>S:<s>	
Response	<HEAD>*<S	
Exchange channels	GPRS	
Designations	**Interpretation**	**Data Format**
<HEAD>	16-byte NTCB packet header with @NTC preamble	U8[16]
*>S	0x2A 0x3E 0x53	char[3]
*<S	0x2A 0x3C 0x53	char[3]
<s>	ID String *	char[15]

** This line includes IMEI of the modem, so modem must be enabled at least once to receive it. When replacing a modem, unique identifier changes.*

Next, message is sent to negotiate protocol versions. This message defines the composition and quantity of data to be transmitted, FLEX protocol version and version of the data structure.

Message	<HEAD>*>FLEX<protocol><protocol_version><struct_version><data_size><bit field[data_size/8]>	
Response	<HEAD>*<FLEX<protocol><protocol_version><struct_version>	
Exchange channels	GPRS	
Designations	**Interpretation**	**Data Format**
<HEAD>	16-byte NTCB packet header with @NTC preamble	U8[16]
*>FLEX	0x2A 0x3E 0x46 0x4C 0x45 0x58	char[6]
*<FLEX	0x2A 0x3C 0x46 0x4C 0x45 0x58	char[6]
<protocol>	Symbol of the protocol in which device is going to work: 0xB0 – FLEX	U8
<protocol_version>	Protocol version is needed to identify compatibility of the command set and packet format on the server and in the device. For version 1.0 - 10 (0x0A) For version 2.0 - 20 (0x14) For version 3.0 - 30 (0x1E)	U8
<struct_version>	Data structure version is needed to identify compatibility of the transmitted data format on the server and in the device. For version 1.0 - 10 (0x0A) For version 2.0 - 20 (0x14) For version 3.0 - 30 (0x1E)	U8
<data_size>	Subsequent configuration field size < bit field [data _ size/8 + (1)] > For structure version 1.0- 69 For structure version 2.0 - 122 For structure version 3.0 - 255	U8
<bit field[data_size/8+(1)]>	Bit array with information about transmitted fields of the data structure. Set bit corresponds to the transmitted field, if the bit is zero, corresponding field is not transmitted. Array value is determined by the device configuration. The length in bytes is calculated as an integer number of bytes that can fit number specified in the <data_size> field. For structure version 1.0 - 9 bytes For structure version 2.0- 16 bytes For structure version 3.0 - 32 bytes	[U8] (byte array)

Command bit field structure

When the number of fields "n" is a multiple of eight, the bytes are filled completely.

Bytes	0								1								..	n/8
Bits	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	..	7 6 5 4 3 2 1 0
FLEX Fields	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	..	n-7..... n

When the number of fields "n" is not a multiple of eight, the filling of the last "incomplete" byte starts from the 7th high bit, and the low bits remain insignificant.

Bytes	0								1								..	n/8+1
Bits	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	..	7 6 5 4 3 2 1 0
FLEX Fields	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	..	n/8*8+1.....(n/8+1)*8

For example, the number of fields n = 21. The number of bytes is equal to 21/8 + 1 = 3 (division is carried out to integers). The first two bytes are full. In the third byte, only the last 21-16 = 5 bits are filled.

Bytes	0								1								2							
Bits	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
FLEX Fields	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	0	0	0

Fields of the FLEX table for which the corresponding bits are not set are omitted. For example, for a field that takes form 0x00 0xE0 0x00 0x00 0x00 0x00 0x00 0x00 0x00 for a FLEX table version 1.0, the transmitted telemetry information will look as follows:

№	№ FLEX	Record Field	Record Item Size	Data Format	Accepted values
1	9	Time of the last valid coordinates (before the event)	4	U32	Number of seconds since 1970
2	10	Last valid latitude	4	I32	Latitude angle recorded when the last valid coordinates were obtained. In ten thousandths of a minute. For example, 55° 42,2389' will be represented as 33422389
3	11	Last valid longitude	4	I32	Longitude angle recorded when the last valid coordinates were obtained. In ten thousandths of a minute. For example, 37° 41,6063' will be represented as 22616063

In order device can start transmitting data specified in the protocol packet, version of the bit mask transmitted from the device and in the response from the server must match. FLEX data formats are backward compatible, i.e. version 2.0 of the data format includes version 1.0, with the added data fields at the end of the data structure. The protocol version is updated when the format and composition of the packets change. Device does not send new FLEX messages to incompatible server software. Thus, the device and the server always operate with compatible sets of commands and data.

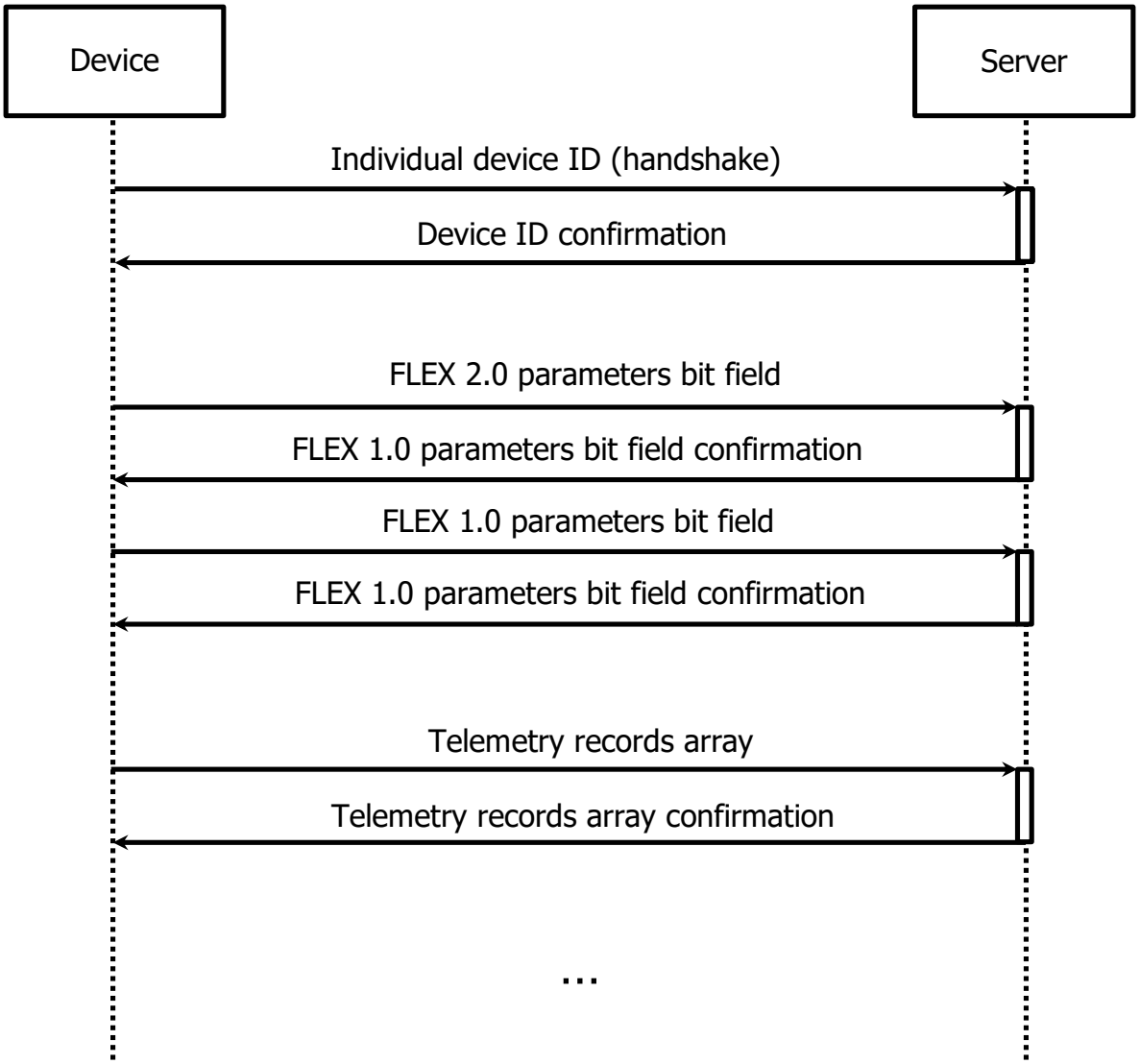
Consider an example of an exchange between a device with an updated version of the protocol and a server that does not support this update. For example, the device supports version 2.0 of the data format, while the server only supports version 1.0 of the data format.

Negotiation process looks as follows:
- device sends a negotiation message with a version 2.0 data bit mask;
- server returns a response containing a 1.0 bit mask;
- device, processing the packet, remembers the server's response and sends a new FLEX message with a version 1.0 bit mask.

If the server again returns a response that specifies the version 1.0 bit mask, then the data transfer starts. If the server returns a response that contains a mismatched bit mask, then the device disconnects from that server. The negotiation of the protocol version looks the same and can be carried out at the same time as the negotiation of the data version, it is enough for the server in the response to immediately indicate the required protocol version and data structure.

Device monitors the number of failed attempts to connect to the server. After three attempts, the server is locked for the interval specified in the configuration.

Example of switching to an old version of the FLEX protocol if the server does not support the new one:



After successful connection to the server, device sends telematic data. In this case, three types of telematics packets are used:

- telemetry record array packet with events that have occurred earlier and are not currently transmitted to the server for some reason; they are contained in non-volatile memory;

- out-of-order messages packet with the current event. These packets are out-of-order and have a higher priority than the archived one. The telemetry transmitted by the device in the out-of-order message may not be duplicated in the archive message, so parsing these packets is mandatory.
- current state packet does not have a corresponding event to be archived. It is sent if the device should transmit telemetry instead of "ping," or is added to the telemetry array packet if a large fragment of old data is transmitted. The current state has a record index of zero and a 0xFF00 event ID.

Telemetry record array packet structure

ATTENTION!
Implementation of responses from the server to the structures listed in this section is required for the correct operation of the device.

Data accumulated in the non-volatile memory is always transmitted in the "Array of telemetry messages". The packet contains a certain number of entries from the non-volatile memory. The packet size does not exceed 1.3 KB (excluding the header). Packets with a single record are allowed.

Message	~A<size><x[0]-x[size-1]><crc8>	
Response	~A<size><crc8>	
Exchange channel	GPRS	
Designations	**Interpretation**	**Data Format**
~A	0x7E 0x41	char[2]
<size>	Number of telemetry records transferred in the array	U8
<x[0]-x[size-1]>	Array of telemetry records with FLEX structure. The number of parameters passed and the size in bytes correspond to the value in the <bit field [data_size/8 + (1)] > field of FLEX packet. The entries follow each other without any separators.	-
<crc8>	8-bit byte CRC8 ~A characters and fields <size><x[0]-x[size-1]> See Annex B. Checksum calculation algorithm CRC8.	U8

To store and transfer some rarely changing data, additional packets and telemetry recording introduced in the protocol extension starting with FLEX 2.0 are used. These packets replace the standard telemetry packets.

Message	~E<count><x[0]-x[count-1]><crc8>	
Response	~E<count><crc8>	
Exchange channel	GPRS	
Designations	**Interpretation**	**Data Format**
<count>	Number of telemetry records transferred in the array	U8
<x[0]-x[size-1]>	Array of additional telemetry records. The entries follow each other without any separators.	-
<crc8>	8-bit byte CRC8 ~E characters <count><x[0]-x[count-1]> See Annex B. Checksum calculation algorithm CRC8.	U8

Out-of-order message packet structure

ATTENTION!

Implementation of responses from the server to the structures listed in this section is required for the correct operation of the device.

Device may transmit important events out of sequence, i.e. out of order of increasing message index in the non-volatile memory. If the event is the last one and has been sent to the server, it will not be sent in an archived message. Sending an out-of-order message has a higher priority than all other packets. Until a response to this message is received, the device will suspend transmission of other packets.

Message	~T<eventindex><x><crc8>	
Response	~T<eventindex><crc8>	
Exchange channel	GPRS	
Designations	**Interpretation**	**Data Format**
~T	0x7E 0x54	char[2]
<x>	Telemetry record, with FLEX structure. The number of parameters passed and the size in bytes correspond to the value in the <bit field[data_size/8+(1)]> field of FLEX packet	-
<eventindex>	The index of the received telemetry record.	U32
<crc8>	8-bit byte CRC8 ~T characters and fields <size><x[0]-x[size-1]> See Annex B. Checksum calculation algorithm CRC8.	U8

Similar to the array of telemetry messages, there are additional out-of-order packets for out-of-order messages introduced in the protocol extension from FLEX 2.0.

Message	~X<eventindex><x><crc8>	
Response	~X<eventindex><crc8>	
Exchange channel	GPRS	
Designations	**Interpretation**	**Data Format**
<x>	Additional telemetry record.	-
<eventindex>	Index of the received telemetry record.	U32
<crc8>	8-bit byte CRC8 ~X characters and fiels <count> and <x[0]-x[count-1]> See Annex B. Checksum calculation algorithm CRC8	U8

Current state packet structure

ATTENTION!
Implementation of responses from the server to the structures listed in this section is required for the correct operation of the device.

Device can also send current state data.
Packet can be added to the transmitted arrays (~A) as the last record, or contained in a separate packet (~C). The current state has a zero record index and eveny code 0xFF00.

Message	~C<x><crc8>	
Response	~C<crc8>	
Exchange channel	GPRS	
Designations	**Interpretation**	**Data Format**
~	0x7E 0x43	char[2]
<x>	Telemetry record, with FLEX structure. The number of parameters passed and the size in bytes correspond to the value in the field <bit field [data_size/8+(1)]> of FLEX packet.	-
<crc8>	8-bit byte CRC8 ~C characters and fields <size><x[0]-x[size-1]> See Annex B. Checksum calculation algorithm CRC8	U8

Examples of basic FLEX packets are provided in Appendix A.3. Examples of basic FLEX packets.

1.2.2. List of FLEX versions and basic messages

List of supported FLEX protocol versions:

Protocol Version	Description
FLEX 1.0	Basic version includes messages described in FLEX Message List for FLEX 1.0.
FLEX 2.0	Includes messages described in FLEX 2.0 FLEX Message List. Added option for work with following devices: camera, tachograph, transparent port. Has its own peculiarities: if camera is configured, a special message will be sent to the server. It informs about the presence of photos.
FLEX 3.0	Protocol version has been increased due to the introduction of a new version of the data structure.

Each next version of the protocol includes the features of the previous one.
If the FLEX version is not supported, you must upgrade to the previous version of the protocol when you connect the device.
It is allowed to use different versions of the protocol and data structure at the same time. For example, FLEX protocol version 1.0, FLEX data structure version 2.0.

List of supported FLEX data structure versions:

Structure Version	Description
FLEX 1.0	Basic data structure is presented in Appendix A.1. Structure of FLEX telemetry records for FLEX 1.0.
FLEX 2.0	Extension of the basic data structure is presented in Appendix A.1. Structure of FLEX telemetry records for FLEX 2.0. Structure of the FLEX 2.0 extension additional telemetry record is presented in Annex A.2. Structure of additional FLEX 2.0 telemetry records
FLEX 3.0	Extension of the basic data structure is presented in Annex A.1. Structure of FLEX telemetry records for FLEX 3.0.

List of FLEX messages

Message Type	Appointment
FLEX 1.0 Telemetry Messages *(implementation of responses from the server is mandatory for correct operation of the device)*	
~A<size><x[0]-x[size-1]><crc8>	Transmission of accumulated telemetry messages from non-volatile memory with a FLEX type structure. Field flags are either taken from the device settings or from the server and specified in a separate command at the beginning of the connection.
~T<eventindex><x><crc8>	Transmission of out-of-order messages with a FLEX type structure. Field flags are either taken from the device settings or from the server and specified in a separate command at the beginning of the connection.
~C<x><crc8>	Transmission of current telemetry messages with a FLEX type structure. Field flags are either taken from the device settings or from the server and specified in a separate command at the beginning of the connection.
DEL (0x7F)	FLEX Protocol Connection Support Message.
FLEX 2.0 Telemetry Messages *(implementation of responses from the server is mandatory for correct operation of the device)*	
~E<count><x[0]-x[count-1]><crc8>	Transmission of accumulated additional telemetry messages from non-volatile memory with a FLEX 2.0 structure.

~X<eventindex><x><crc8>	Transmission of additional out-of-order messages with a FLEX 2.0 structure.
FLEX 2.0 Service Messages List	
Q (query)	Data query (version, device status, etc.).
I (information)	Response to the request if the requested information is available.
U (unavailability)	Response to the request if the requested information is not available.
O (order)	Command (activation of output lines, protection, etc.).
R (response)	Response to the command if executed.
F (failure)	Response to the command if it is not executed.
N (notification)	Notification.
G (get)	Data block query.
L (lack)	Negative response to block query.
D (data)	Data block transmitted on query.
P (put)	Loading data block.
S (saturation)	Negative response to data block loading.
M (more)	Confirms the load of the data block.

For detailed examples of FLEX service messages, see the sections that describe specific functions.

2. NTCB and FLEX application layer. Description of the structure of request packets, commands and information messages

The main types of the messages are:

1) requests - transmitted from the "host" to the device. Requests are not captured in the device's non-volatile memory, excluding requesting the current state of the device;

2) information messages and requests from the device - transmitted at the initiative of the device itself and used to transmit telemetry from the device to the server and exchange information with RCS (Remote Configuration Service) and RFU (Remote Firmware Update) services;

3) commands - transmitted from the server ("host") to the device. Execution of the "host" command is monitored when a response to commands is received. Commands are mapped to events that occur in the device, and when executed, they are recorded in a non-volatile memory.

2.1. System commands, queries, messages

Query model and firmware version

Query	<HEAD>*?V	
Response	<HEAD>*#V:<n>:<v1>.<v2>.<v3>:<d>.<m>.<y>:<loc> Example: <HEAD>*#V:E-1110:01.00.53:07.02.08:RU	
Exchange channel	GPRS, USB, Bluetooth, SMS	
Designations	**Interpretation**	**Data Format**
<HEAD>	16 byte NTCB packet header with preamble	16*U8
*?V	0x2A 0x3F 0x56	char[3]
*#V	0x2A 0x23 0x56	char[3]
<n>	Device model string (6 characters).	char[6]
<v1>.<v2>.<v3>	Firmware version indexes of 2 characters.	char[2]
<d>.<m>.<y>	Accordingly, day, month and year of this firmware version are 2 characters each.	char[2]
<loc>	Firmware language version 2 characters (RU, DE, EN).	char[2]

Query unique device ID

Query	<HEAD>*?S	
Response	<HEAD>*#S:<s> Example: <HEAD>*#S:123456789012345	
Exchange channel	GPRS, USB, Bluetooth, SMS	
Designations	**Interpretation**	**Data Format**
<HEAD>	16 byte NTCB packet header with preamble	16*U8
*?S	0x2A 0x3F 0x53	char[3]
*#S	0x2A 0x23 0x53	char[3]
<s>	Identifier string.	char[]

Note

This line includes IMEI of the modem, so the modem must be enabled at least once to receive it. When replacing a modem, the unique identifier changes.

Message with a string of unique device ID, sent during handshake procedure

Message	<HEAD>*>S:<s>	
Response	<HEAD>*<S	
Exchange channel	GPRS	
Designations	**Interpretation**	**Data Format**
<HEAD>	16 byte NTCB packet header with preamble	16*U8
*>S	0x2A 0x3E 0x53	char[3]
*<S	0x2A 0x3C 0x53	char[3]
<s>	Identifier string.	char[]

Note
This line includes IMEI of the modem, so the modem must be enabled at least once to receive it. When replacing a modem, the unique identifier changes.

Device restart command

Command	<HEAD>*!DEV_RESET	
Response	<HEAD>*@DEV_RESET	
Exchange channel	GPRS, USB, Bluetooth, SMS	
Designations	**Interpretation**	**Data Format**
<HEAD>	16 byte NTCB packet header with preamble	16*U8
*!DEV_RESET	0x2a 0x21 0x44 0x45 0x56 0x5f 0x52 0x45 0x53 0x45 0x54	char[11]
*@DEV_RESET	0x2a 0x40 0x44 0x45 0x56 0x5f 0x52 0x45 0x53 0x45 0x54	char[11]

Arbitrary USSD request

Command	<HEAD>*?USSD<s><code>	
Response	<HEAD>*#USSD<s><string>	
Exchange channel	GPRS, USB, Bluetooth, SMS	
Designations	**Interpretation**	**Data Format**
<HEAD>	16 byte NTCB packet header with preamble	16*U8
*?USSD	0x2A 0x3F 0x55 0x53 0x53 0x44	char[6]
*#USSD	0x2A 0x23 0x55 0x53 0x53 0x44	char[6]
<s>	Parameter separator - space (0x20).	char
<code>	USSD code string of the request.	char[]
<string>	Response string of the cellular operator.	char[]

Command of arbitrary sound indication by buzzer

Command	<HEAD>*!BEEP<s><count>,<group>,<interval>,<freq>	
Response	<HEAD>*@BEEP<s><result>	
Exchange channel	GPRS, USB, Bluetooth, SMS	
Designations	**Interpretation**	**Data Format**
<HEAD>	16 byte NTCB packet header with preamble	16*U8
*!BEEP	0x2A 0x21 0x42 0x45 0x45 0x50	char[6]
*@BEEP	0x2A 0x40 0x42 0x45 0x45 0x50	char[6]
<s>	Parameter separator - space (0x20).	char
<result>	Command Processing Result (ASCII): "OK" - commands executed; "FAIL" - command execution error;	char[]
<count>	Number of pulses in each pulse group. If >=32, sound indication is continuous.	char[]
<group>	Number of groups in audible indication.	char[]

<interval>	Duration of the group's sound in 1/128 fractions of a second.	char[]
<freq>	Frequency of audible indication of each pulse (Hz).	char[]

Command to change SIM card

Command	<HEAD>*!CHNGSIM	
Response	<HEAD>*@CHNGSIM<s><x>-><y>	
Exchange channel	GPRS, USB, Bluetooth, SMS	
Designations	**Interpretation**	**Data Format**
<HEAD>	16 byte NTCB packet header with preamble	16*U8
*!CHNGSIM	0x2A 0x21 0x43 0x48 0x4E 0x47 0x53 0x49 0x4D	char[9]
*@CHNGSIM	0x2A 0x40 0x43 0x48 0x4E 0x47 0x53 0x49 0x4D	char[9]
<s>	Parameter separator - space (0x20).	char
->	Separating characters - 0x2D 0x3E	char[2]
<x>	Current working SIM card: "1" - SIM 1 (External), "2" - SIM 2 (Internal).	char
<y>	SIM card to switch to: "1" - SIM 1 (External), "2" - SIM 2 (Internal).	char

Query device status with SMS sending

Upon this request, SMS message M:114 is forwarded to the number specified in it.

Command	<HEAD>*?ES<s><phone>	
Response	<HEAD>*#ES<s><phone>	
Exchange channel	GPRS, USB, Bluetooth, SMS	
Designations	**Interpretation**	**Data Format**
<HEAD>	16 byte NTCB packet header with preamble	16*U8
<s>	Parameter separator - space (0x20).	char
*?ES	0x2A 0x3F 0x45 0x53	char[4]
*#ES	0x2A 0x23 0x45 0x53	char[4]
<phone>	Phone number (starting from "+") from which you need to send an SMS.	char[]

Query unique SIM serial number

Command	<HEAD>*?ICCID	
Response	<HEAD>*#ICCID<s><id>	
Exchange channel	GPRS, USB, Bluetooth, SMS	
Designations	**Interpretation**	**Data Format**
<HEAD>	16 byte NTCB packet header with preamble	16*U8
<s>	Parameter separator - space (0x20).	char
*?ICCID	0x2A 0x3F 0x49 0x43 0x43 0x49 0x44	char[7]
*#ICCID	0x2A 0x23 0x49 0x43 0x43 0x49 0x44	char[7]
<id>	Unique SIM serial number	char[]

Query information about firmware version of the GPS receiver

Command	<HEAD>*?VGPS	
Response	<HEAD>*#VGPS<s><n>,<v1>.<v2>.<v3>,<d>.<m>.<y>,<gps_ver>	
Exchange channel	GPRS, USB, Bluetooth, SMS	
Designations	**Interpretation**	**Data Format**
<HEAD>	16 byte NTCB packet header with preamble	16*U8
*?VGPS	0x2A 0x3F 0x56 0x47 0x50 0x53	char[6]
*#VGPS	0x2A 0x23 0x56 0x47 0x50 0x53	char[6]
<s>	Parameter separator - space (0x20).	char
<n>	Device model string (6 characters).	char[6]

<v1>.<v2>.<v3>	Device firmware version indexes of 2 characters.	char[2]
<d>.<m>.<y>	Accordingly, day, month and year of this version of the device firmware are 2 characters each.	char[2]
<gps_ver>	String with model and version of navigation receiver	char[]

Command for microphone listening

Command	<HEAD>*!O<s><phnumber>	
Response	<HEAD>*@O<s><phnumber>	
Exchange channel	GPRS, USB, Bluetooth, SMS	
Designations	**Interpretation**	**Data Format**
<HEAD>	16 byte NTCB packet header with preamble	16*U8
*!O	0x2A 0x21 0x4F	char[3]
*@O	0x2A 0x40 0x4F	char[3]
<s>	Parameter separator - space (0x20).	char
<phnumber>	<phnumber> - number to be called. Phone number (starting with "+").	char[]

Command for time adjustment on device

Command	<HEAD>*!SETTIME<s><offset>	
Response	<HEAD>*@SETTIME<s><ans>	
Exchange channel	GPRS, USB, Bluetooth, SMS	
Designations	**Interpretation**	**Data Format**
<HEAD>	16 byte NTCB packet header with preamble	16*U8
*!SETTIME	0x2A 0x21 0x53 0x45 0x54 0x54 0x49 0x4D 0x45	char[9]
*@SETTIME	0x2A 0x40 0x53 0x45 0x54 0x54 0x49 0x4D 0x45	char[9]
<s>	Parameter separator - space (0x20).	char
<offset>	Offset in seconds with a character in text format. For example: +60 or -30	char
<ans>	Response: "OK" - "time synchronized"; "FAIL" - "synchronization error."	char

Command for diagnostic information query

Command	<HEAD>*?DATA<s><id1>,<id2>, ... ,<idN> Example: <HEAD>*?DATA 21,22	
Response	<HEAD>*#DATA<s><id1>:<data1>,<id2>:<data2>,...,<idN>:<dataN> Example: <HEAD>*#DATA 21:0026,22:0000	
Exchange channel	GPRS, USB, Bluetooth, SMS	
Designations	**Interpretation**	**Data Format**
<HEAD>	16 byte NTCB packet header with preamble	16*U8
*?DATA	0x2A 0x3F 0x44 0x41 0x54 0x41	char[6]
*#DATA	0x2A 0x23 0x44 0x41 0x54 0x41	char[6]
<s>	Parameter separator - space (0x20).	char
<idX>	Parameter ID in text decimal form. 1 .. 255 - parameters correspond to table FLEX 3.0 (Annex A.1. Structure of FLEX telemetry records)	char[]
<dataX>	Value of the parameter in the form of a HEX (ASCII) string with the BigEndian byte order.	char[]

Response to unsupported messages

Response to NTCB message	<HEAD>*Z:<msg>	
Response to FLEX message	~Z:<msg><crc8>	
Exchange channel	GPRS, USB, Bluetooth	
Designations	**Interpretation**	**Data Format**
<HEAD>	16 byte NTCB packet header with preamble	16*U8
*Z:	0x2A 0x5A 0x3A	char[3]
~Z:	0x7E 0x5A 0x3A	char[3]
<msg>	Unsupported FLEX or NTCB message is sent back. For NTCB, entire message is forwarded. For FLEX, first 2 characters after "~" are significant during validation, only they are sent (if any).	char[]
<crc8>	FLEX checksum. See Annex B. Checksum calculation algorithm CRC8	U8

2.2. Outputs control

Command to change output state

Command	<HEAD>*!SETOUT<s><num><new_state>[,<num><new_state>] For example: *!SETOUT 1Y,2N	
Response	<HEAD>*@SETOUT<s><result>,<num><cur_state>[,<num><cur_state>] For example: *@SETOUT OK,1Y,2N или *@SETOUT FAIL,1N,2Y	
Exchange channel	GPRS, USB, Bluetooth, SMS	
Designations	**Interpretation**	**Data Format**
<HEAD>	16 byte NTCB packet header with preamble	16*U8
*!SETOUT	0x2A, 0x21, 0x53, 0x45, 0x54, 0x4F, 0x55, 0x54	char[8]
*@SETOUT	0x2A, 0x40, 0x53, 0x45, 0x54, 0x4F, 0x55, 0x54	char[8]
<s>	Parameter separator - space (0x20).	char
<result>	Command Processing Result (ASCII): "OK" - commands executed; "FAIL" - command execution error;	char[]
<num>	Number of the output to be changed (ASCII). Numbering starts with 1.	char
<new_state>	Output state to be set (ASCII): "Y" - enable; "N" - disable.	char
<cur_state>	Output status after command execution (ASCII): "Y" - on; "N" - off.	char

Command to activate output 1

Command	<HEAD>*!1Y	
Response	<HEAD>*@C<x>	
Exchange channel	GPRS, USB, Bluetooth	
Designations	**Interpretation**	**Data Format**
<HEAD>	16 byte NTCB packet header with preamble	16*U8
*!1Y	0x2A 0x21 0x31 0x59	char[4]
*@C	0x2A 0x40 0x43	char[3]
<x>	Telemetry record with a structure dependent on the storage and exchange protocol.	-

Command to disactivate output 1

Command	<HEAD>*!1N	
Response	<HEAD>*@C<x>	
Exchange channel	GPRS, USB, Bluetooth	
Designations	**Interpretation**	**Data Format**
<HEAD>	16 byte NTCB packet header with preamble	16*U8
*!1N	0x2A 0x21 0x31 0x4E	char[4]
*@C	0x2A 0x40 0x43	char[3]
<x>	Telemetry record with a structure dependent on the storage and exchange protocol.	-

Command to activate output 2

Command	<HEAD>*!2Y	
Response	<HEAD>*@C<x>	
Exchange channel	GPRS, USB, Bluetooth	
Designations	**Interpretation**	**Data Format**
<HEAD>	16 byte NTCB packet header with preamble	16*U8
*!2Y	0x2A 0x21 0x32 0x59	char[4]
*@C	0x2A 0x40 0x43	char[3]
<x>	Telemetry record with a structure dependent on the storage and exchange protocol.	-

Command to disactivate output 2

Command	<HEAD>*!2N	
Response	<HEAD>*@C<x>	
Exchange channel	GPRS, USB, Bluetooth	
Designations	**Interpretation**	**Data Format**
<HEAD>	16 byte NTCB packet header with preamble	16*U8
*!2N	0x2A 0x21 0x32 0x4E	char[4]
*@C	0x2A 0x40 0x43	char[3]
<x>	Telemetry record with a structure dependent on the storage and exchange protocol.	-

Command to activate output 3

Command	<HEAD>*!3Y	
Response	<HEAD>*@C<x>	
Exchange channel	GPRS, USB, Bluetooth	
Designations	**Interpretation**	**Data Format**
<HEAD>	16 byte NTCB packet header with preamble	16*U8
*!3Y	0x2A 0x21 0x33 0x59	char[4]
*@C	0x2A 0x40 0x43	char[3]
<x>	Telemetry record with a structure dependent on the storage and exchange protocol.	-

Command to disactivate output 3

Command	<HEAD>*!3N	
Response	<HEAD>*@C<x>	
Exchange channel	GPRS, USB, Bluetooth	
Designations	**Interpretation**	**Data Format**
<HEAD>	16 byte NTCB packet header with preamble	16*U8
*!3N	0x2A 0x21 0x33 0x4E	char[4]
*@C	0x2A 0x40 0x43	char[3]
<x>	Telemetry record with a structure dependent on the storage and exchange protocol.	-

Command to activate output 4

Command	<HEAD>*!4Y	
Response	<HEAD>*@C<x>	
Exchange channel	GPRS, USB, Bluetooth	
Designations	**Interpretation**	**Data Format**
<HEAD>	16 byte NTCB packet header with preamble	16*U8
*!4Y	0x2A 0x21 0x34 0x59	char[4]
*@C	0x2A 0x40 0x43	char[3]
<x>	Telemetry record with a structure dependent on the storage and exchange protocol.	-

Command to disactivate output 4

Command	<HEAD>*!4N	
Response	<HEAD>*@C<x>	
Exchange channel	GPRS, USB, Bluetooth	
Designations	**Interpretation**	**Data Format**
<HEAD>	16 byte NTCB packet header with preamble	16*U8
*!4N	0x2A 0x21 0x34 0x4E	char[4]
*@C	0x2A 0x40 0x43	char[3]
<x>	Telemetry record with a structure dependent on the storage and exchange protocol.	-

2.3. Inputs control

Command to lock input

Command	<HEAD>*!OFF<s><index>	
Response	<HEAD>*@C<x>	
Exchange channel	GPRS, USB, Bluetooth	
Designations	**Interpretation**	**Data Format**
<HEAD>	16 byte NTCB packet header with preamble	16*U8
*!OFF	0x2A 0x21 0x4F 0x46 0x46	char[5]
*@C	0x2A 0x40 0x43	char[3]
<s>	Parameter separator - space (0x20).	char
<index>	Index of the input to be locked (starting with 1). The parameter is written in symbolic form: "1"... "9" (0x31 - 0x39).	char
<x>	Telemetry record with a structure dependent on the storage and exchange protocol.	-

Command to unlock input

Command	<HEAD>*!ON<s><index>	
Response	<HEAD>*@C<x>	
Exchange channel	GPRS, USB, Bluetooth	
Designations	**Interpretation**	**Data Format**
<HEAD>	16 byte NTCB packet header with preamble	16*U8
*!ON	0x2A 0x21 0x4F 0x4E	char[4]
*@C	0x2A 0x40 0x43	char[3]
<s>	Parameter separator - space (0x20).	char
<index>	Index of the input to be unlocked (starting with 1). The parameter is written in symbolic form: "1"... "9" (0x31 - 0x39).	char
<x>	Telemetry record with a structure dependent on the storage and exchange protocol.	-

2.4. Telemetry commands, queries, messages

FLEX telemetry messages array

Message	~A<size><x[0]-x[size-1]><crc8>	
Response	~A<size><crc8>	
Exchange channel	GPRS	
Designations	**Interpretation**	**Data Format**
~A	0x7E 0x41	char[2]
<size>	Number of telemetry records transferred in the array	U8
<x[0]-x[size-1]>	The number of parameters passed and the size in bytes corresponds to the value in the <bit field[data_size/8+(1)]> field of the FLEX packet. The entries follow each other without any separators.	-
<crc8>	8-bit byte CRC8 ~A characters and fields <size><x[0]-x[size-1]> See Annex B. Checksum calculation algorithm CRC8	U8

Out-of-order FLEX message packet structure

Message	~T<eventindex><x><crc8>	
Response	~T<eventindex><crc8>	
Exchange channel	GPRS	
Designations	**Interpretation**	**Data Format**
~T	0x7E 0x54	char[2]
<x>	Telemetry record with FLEX structure. The number of parameters passed and the size in bytes corresponds to the value in the <bit field[data_size/8+(1)]> field of the FLEX packet.	-
<eventindex>	Index of the received telemetry record.	U32
<crc8>	8-bit byte CRC8 ~T characters and fields <size><x[0]-x[size-1]> See Annex B. Checksum calculation algorithm CRC8	U8

FLEX state packet structure

Message	~C<x><crc8>	
Response	~C<crc8>	
Exchange channel	GPRS	
Designations	**Interpretation**	**Data Format**
~	0x7E 0x43	char[2]
<x>	Telemetry record, with FLEX structure. The number of parameters passed and the size in bytes correspond to the value in the <bit field[data_size/8+(1)]> field of the FLEX packet.	-
<crc8>	8-bit byte CRC8 ~T characters and fields <size><x[0]-x[size-1]> See Annex B. Checksum calculation algorithm CRC8	U8

Array of additional FLEX telemetry messages

Message	~E<count><x[0]-x[count-1]><crc8>	
Response	~E<count><crc8>	
Exchange channel	GPRS	
Designations	**Interpretation**	**Data Format**
<count>	Number of telemetry records transferred in the array	U8
<x[0]-x[size-1]>	Array of additional telemetry records. Records follow each other without any separators.	-
<crc8>	8bit byte CRC8 ~E characters <count><x[0]-x[count-1]> See Annex B. Checksum calculation algorithm CRC8	U

Structure of additional out-of-order FLEX message packet

Message	~X<eventindex><x><crc8>	
Response	~X<eventindex><crc8>	
Exchange channel	GPRS	
Designations	**Interpretation**	**Data Format**
<x>	Additional telemetry record.	-
<eventindex>	Index of the received telemetry record.	U32
<crc8>	8bit byte CRC8 ~X characters <count><x[0]-x[count-1]> See Annex B. Checksum calculation algorithm CRC8	U8

Query the nearest telemetry record before the specified date

Query	<HEAD>*?L<h><mn><s><d><m><y>	
Response	<HEAD>*#L<h><mn><s><d><m><y><page><x>	
Exchange channel	GPRS, USB, Bluetooth	
Designations	**Interpretation**	**Data Format**
<HEAD>	16 byte NTCB packet header with preamble	16*U8
*?L	0x2A 0x3F 0x4C	char[3]
*#L	0x2A 0x23 0x4C	char[3]
<h><mn><s><d><m><y>	Hour, minute, second, day, month, year of the event record. If h=m=s=d=m=y=0, record with a minimum number is searched.	U8 (of each field)
<page>	Page number of the non-volatile memory on which the record is stored.	U32
<x>	Telemetry record, with a structure dependent on the storage and exchange protocol.	-

Query the nearest telemetry record after the specified date

Query	<HEAD>*?R<h><m><s><d><m><y>	
Response	<HEAD>*#R<h><min><s><d><m><y><page><x>	
Exchange channel	GPRS, USB, Bluetooth	
Designations	**Interpretation**	**Data Format**
<HEAD>	16 byte NTCB packet header with preamble	16*U8
*?R	0x2A 0x3F 0x52	char[3]
*#R	0x2A 0x23 0x52	char[3]
<h><mn><s><d><m><y>	Hour, minute, second, day, month, year of the event record. If h=m=s=d=m=y=255, record with a maximum number is searched.	U8 (of each field)
<page>	Page number of the non-volatile memory on which the record is stored.	U32
<x>	Telemetry record, structure depends on the protocol.	-

Query telemetry record by its index

Query	<HEAD>*?I<index>	
Response	<HEAD>*#I<x>	
Exchange channel	GPRS, USB, Bluetooth	
Designations	**Interpretation**	**Data Format**
<HEAD>	16 byte NTCB packet header with preamble	16*U8
*?I	0x2A 0x3F 0x49	char[3]
*#I	0x2A 0x23 0x49	char[3]
<index>	Absolute index of the record in the non-volatile memory.	U32
<x>	Telemetry record, structure depends on the protocol.	-

Command to confirm synchronization of internal memory with the server

Command	<HEAD>*!SYNC<s><index>	
Response	<HEAD>*@C<x>	
Exchange channel	GPRS, USB, Bluetooth	
Designations	**Interpretation**	**Data Format**
<HEAD>	16 byte NTCB packet header with preamble	16*U8
*!SYNC	0x2A 0x21 0x53 0x59 0x4E 0x43	char[6]
*@C	0x2A 0x40 0x43	char[3]
<s>	Parameter separator - space (0x20) or colon (0x3A).	char
<index>	Server index specified in the device settings (starting from 1). The record is binary - 0x01 or symbolic (ASCII) - "1."	U8 / char
<x>	Telemetry record, structure depends on the protocol.	-

Note
This command is issued when data has been removed from the device bypassing the GPRS algorithm (for example, via USB), and is sent in order to cancel the subsequent transfer according to the main operation algorithm.

Query current state of the device in the main packet

Query	<HEAD>*?A	
Response	<HEAD>*#A<x>	
Exchange channel	GPRS, USB, Bluetooth	
Designations	**Interpretation**	**Data Format**
<HEAD>	16 byte NTCB packet header with preamble	16*U8
*?A	0x2A 0x3F 0x41	char[3]
*#A	0x2A 0x23 0x41	char[3]
<x>	Main telemetry record, structure depends on the protocol.	-

Query current state of the device in the additional packet

Query	<HEAD>*?E<s><type>	
Response	<HEAD>*#E<x>	
Exchange channel	GPRS, USB, Bluetooth	
Designations	**Interpretation**	**Data Format**
<HEAD>	16 byte NTCB packet header with preamble	16*U8
*?E	0x2A 0x3F 0x45	char[3]
*#E	0x2A 0x23 0x45	char[3]
<s>	Parameter separator - space (0x20).	char[1]
<type>	Type of the dynamic part of the additional packet: '1' – TouchKey "2" - *not used* '3' - RFID	char[1]
<x>	Additional telemetry record, structure depends on the protocol (see Annex A.2. Structure of additional FLEX 2.0 telemetry records)	-

Command to resend telemetry from non-volatile memory

Command	<HEAD>*!REP_FL<s><srvindex>,<leftdate>[/<lefttime>],<rightdate>[/<righttime>] Example: <HEAD>*!REP_FL 1,09.04.18/13:00:59,10.04.18/03:00:00 <HEAD>*!REP_FL 1,09.04.18,10.04.18	
Response	<HEAD>*@REP_FL<s><result>	
Exchange channel	GPRS, USB, Bluetooth, SMS	
Designations	**Interpretation**	**Data Format**
<HEAD>	16 byte NTCB packet header with preamble	16*U8
*!REP_FL	0x2A 0x21 0x52 0x45 0x50 0x5F 0x46 0x4C	char[8]
*@REP_FL	0x2A 0x40 0x52 0x45 0x50 0x5F 0x46 0x4C	char[8]
<s>	Parameter separator - space (0x20).	char
/	Date/Time field separator (0x2F)	char
<result>	Command processing result (ASCII): "OK" - commands executed; "FAIL" - command execution failed.	char[]
<srvindex>	Server index to repeat in text format: "0" - to all servers; "1.." "3" –server index.	char
<leftdate>	Date of the left border of the interval of the requested telemetry in text format (in UTC): "DD.MM.YY."	char[8]
<lefttime>	***Optional parameter.*** Time of the left border of the interval of the requested telemetry in text format (in UTC) is "HH: MM: SS Absence of a parameter is equal to the value "00:00:00."	char[8]
<rightdate>	Date of the right border of the interval of the requested telemetry in text format (in UTC): "DD.MM.YY."	char[8]
<righttime>	***Optional parameter.*** Time of the right border of the requested telemetry interval in text format (in UTC) is "HH: MM: SS Absence of a parameter is equal to the value "23:59:59."	char[8]

Command to resend telemetry from SD card

Command	<HEAD>*!REP_SD<s><srvindex>,<leftdate>[/<lefttime>],<rightdate>[/<righttime>] Example: < HEAD > *! REP _ SD 1.09.04.18/13: 00: 59.10.04.18/03*: 00:00 (time will not be* < HEAD > *! REP _ SD 1.09.04.18. 10.04.18 (only *09.04.18 will be unloaded*	
Response	<HEAD>*@REP_SD<s><result>	
Exchange channel	GPRS, USB, Bluetooth, SMS	
Designations	**Interpretation**	**Data Format**
<HEAD>	16 byte NTCB packet header with preamble	16*U8
*!REP_SD	0x2A 0x21 0x52 0x45 0x50 0x5F 0x53 0x44	char[8]
*@REP_SD	0x2A 0x40 0x52 0x45 0x50 0x5F 0x53 0x44	char[8]
<s>	Parameter separator - space (0x20).	char
/	Date/Time field separator (0x2F)	char
<result>	Command processing result (ASCII): "OK" - commands executed; "FAIL" - command execution failed.	char[]
<srvindex>	Server index to repeat in text format: "0" - to all servers; "1.." "3" – server index.	char
<leftdate>	Date of the left border of the interval of the requested telemetry in text format (in UTC): "DD.MM.YY."	char[8]
<lefttime>	***Device does not parse the parameter.*** Always equates to "00:00:00." Time of the left border of the interval of the requested telemetry in text format (in UTC) is "HH: MM: SS	char[8]
<rightdate>	Ddate of the right border of the interval of the requested telemetry in text format (in UTC): "DD.MM.YY."	char[8]
<righttime>	***Device does not parse the parameter.*** Always equates to "00:00:00." Time of the right border of the requested telemetry interval in text format (in UTC) is "HH: MM: SS	char[8]

2.5. Device operating modes control

Command to enable security mode

Command	<HEAD>*!GY	
Response	<HEAD>*@C<x>	
Exchange channel	GPRS, USB, Bluetooth	
Designations	**Interpretation**	**Data Format**
<HEAD>	16 byte NTCB packet header with preamble	16*U8
*!GY	0x2A 0x21 0x47 0x59	char[4]
*@C	0x2A 0x40 0x43	char[3]
<x>	Telemetry record	-

Command to disable security mode

Command	<HEAD>*!GN	
Response	<HEAD>*@C<x>	
Exchange channel	GPRS, USB, Bluetooth	
Designations	**Interpretation**	**Data Format**
<HEAD>	16 byte NTCB packet header with preamble	16*U8
*!GN	0x2A 0x21 0x47 0x4E	char[4]
*@C	0x2A 0x40 0x43	char[3]
<x>	Telemetry record	-

Command to change device operating mode

Command	<HEAD>*!M<s><x>	
Response	<HEAD>*@M<s><x>,<e>,<i> Example: *@M G *@M O,5,IN1/IN2/ENG	
Exchange channel	GPRS, USB, Bluetooth, SMS	
Designations	**Interpretation**	**Data Format**
<HEAD>	16 byte NTCB packet header with preamble	16*U8
*!M	0x2A 0x21 0x4D	char[3]
*@M	0x2A 0x40 0x4D	char[3]
<s>	Parameter separator - space (0x20).	char
<x>	Operating mode of the device to be switched to: "G" - security; "O" - surveillance. Response <x> is the current operating mode.	char
<e>	Reason for not switching to security mode: "1" - security mode in the device configuration is disabled; "2" - timeout of the mode change prohibition has not expired; "3" - mode is on: do not switch to the security mode when the ignition is on; "4" - device is already in this mode; "5" - mode is on: do not go into security if one of the security sensors is triggered.	char
<i>	Only if <e> = 5 Aliases of trigerred sensors via **/**. ***For S-25xx, S-23xx devices:*** "IN1," "IN2," "IN3," "IN4," "AIN1," "AIN2" - inputs; "VOLT" - external supply voltage sensor; "ENG" - engine operation sensor. ***For S-26xx, S-24xx devices:*** "IN1," "IN2," "IN3," "IN4," "IN5," "IN6" - inputs; "VOLT" - external supply voltage sensor; "ENG" - engine operation sensor.	char[]

Query current operating mode

Command	<HEAD>*?M	
Response	<HEAD>*#M<s><x>	
Exchange channel	GPRS, USB, Bluetooth, SMS	
Designations	**Interpretation**	**Data Format**
<HEAD>	16 byte NTCB packet header with preamble	16*U8
*?M	0x2A 0x3F 0x4D	char[3]
*#M	0x2A 0x23 0x4D	char[3]
<s>	Parameter separator - space (0x20).	char
<x>	Device operating mode: "G" - security; "O" - surveillance.	char

2.6. Connection to RCS, RFU, DRC services

Device is connected to RCS, RFU servers by sending corresponding commands via SMS channels (5.2.5 RCS, RFU, DRC services) or GPRS.

RCS Remote Configuration Service is designed to provide information interaction between the NTC Configurator and the device via GPRS to change its configuration, update frimware and read telemetry from the device. RCS service is a router server that provides information exchange between the device and the program running on it on the PC. Switching occurs by the unique ID (session ID) of this router, which receives the program that opens the connection to this service. The identifier is passed to the device.

Command to connect to the configurator via RCS

Command	<HEAD>*!CNCT_RCS<s><ip>,<port>,<commID>,<apn>,<login>,<password> **or (for backward compatibility)** <HEAD>*!CNCT_RCS:<ip>:<port>:<commID>:<apn>:<login>:<password>	
Response	<HEAD>*@C<x>	
Exchange channel	GPRS, USB, Bluetooth	
Designations	**Interpretation**	**Data Format**
<HEAD>	16-byte NTCB packet header	U8[16]
*!CNCT_RCS	0x2A 0x21 0x43 0x4E 0x43 0x54 0x5F 0x52 0x43 0x53	char[10]
*@C	0x2A 0x40 0x43	char[3]
<s>	Parameter separator - space (0x20).	char
<ip>	String containing the IP address of the RCS server. For example: 89.208.152.55.	char[] minimum 7, maximum 15
<port>	String containing the port on which the RCS service on the server is routed. For example: 8100.	char[] maximum 5
<commID>	Session ID to connect to the configurator. For example: 43644176.	char[] maximum 8
<apn>	***Optional parameter.*** Access point name of the cellular operator. If the field in the cellular operator settings is empty, the field should be omitted. For example: internet.mts.ru.	char[] maximum 30
<login>	***Optional parameter.*** Login of the cellular operator. If the field in the cellular operator settings is empty, the field should be omitted. For example: mts.	char[] maximum 20
<password>	***Optional parameter.*** Password of the cellular operator. If the field in the cellular operator settings is empty, the field should be omitted For example: mts.	char[] maximum 20
<x>	Telemetry record, with a structure dependent on the storage and exchange protocol.	U8[]

RFU service is designed to automatically change the firmware in the device. Device connects to the RFU server by command: via SMS channel (5.2.5 RCS, RFU services) or GPRS.
Principle of operation of the device with the RFU server is as follows: first, device is remotely connected, then device downloads a new version of the program from the server. Then it reboots, then the work is carried out on the new version of the program.

Command to update device firmware via RFU

Command	<HEAD>*!CNCT_RFU<s><ip>,<port>,<firmware>,<apn>,<login>,<password> **or (for backward compatibility)** <HEAD>*!CNCT_RFU:<ip>:<port>:<firmware>:<apn>:<login>:<password>	
Response	<HEAD>*@C<x>	
Exchange channel	GPRS, USB, Bluetooth	
Designations	**Interpretation**	**Data Format**
<HEAD>	16-byte NTCB packet header	U8[16]
*!CNCT_RFU	0x2A 0x21 0x43 0x4E 0x43 0x54 0x5F 0x52 0x46 0x55	char[10]
*@C	0x2A 0x40 0x43	char[3]
<s>	Parameter separator - space (0x20).	char
<ip>	String containing the IP address of the RCS server. For example: 89.208.152.55.	char[] minimum 7, maximum 15
<port>	String containing the port on which the RCS service on the server is routed. For example: 9100.	char[] maximum 5
<firmware>	Requested firmware version. For example: 02.01.00. If the latest version is needed, the keyword "LAST is used.	char[] maximum 8
<apn>	***Optional parameter.*** Access point name of the cellular operator. If the field in the cellular operator settings is empty, the field should be omitted. For example: internet.mts.ru.	char[] maximum 30
<login>	***Optional parameter.*** Login of the cellular operator. If the field in the cellular operator settings is empty, the field should be omitted. For example: mts.	char[] maximum 20
<password>	***Optional parameter.*** Password of the cellular operator. If the field in the cellular operator settings is empty, the field should be omitted For example: mts.	char[] maximum 20
<x>	Telemetry record, with a structure dependent on the storage and exchange protocol.	U8[]

After receiving the RCS and RFU commands, device generates and transmits response to the telematic server, and then breaks communication with the telematic server and establishes communication with the corresponding RCS and RFU server. Only request packets are used when the device is connected to the RFU service. When connecting to the RCS service, only packet to establish communication is sent. Further, device works the same way as via the USB interface.

DRC service is designed for remote device management.
Devices with a user-defined period or command are connected to the DRC.
When device connects to the service, it checks the task queue for the device. If the task queue is empty, then the service disconnects the device, if there are tasks in the queue, then they are executed one by one (in the order of creation). Tasks queued but not completed can be canceled (deleted) by the user.

Command to connect to DRC

Command	<HEAD>*!CNCT_DRC[<s><time>,<ip>,<port>] Example: *!CNCT_DRC *!CNCT_DRC 300,89.208.152.54,10000	
Response	<HEAD>*@CNCT_DRC<s><ack>	
Exchange channel	GPRS, USB, Bluetooth, SMS	
Designations	**Interpretation**	**Data Format**
<HEAD>	16-byte NTCB packet header	U8[16]
*!CNCT_DRC	0x2A 0x21 0x43 0x4E 0x43 0x54 0x5F 0x44 0x52 0x43	char[10]
*@CNCT_DRC	0x2A 0x40 0x43 0x4E 0x43 0x54 0x5F 0x44 0x52 0x43	char[10]
<time>	*(optional)* Time, in seconds, for which the device is connected.	char[]
<ip>	*(optional)* IP or DNS of the DRC service. If not specified, device uses the server IP address hard-coded in the firmware.	char[]
<port>	*(optional)* PORT of the DRC service. If not specified, device uses the server port hard-coded in the firmware.	char[]
<ack>	Confirmation of command reception: OK - command is accepted for execution.	char[]

2.7. Working with identifiers

Among other telemetry data, server may receive data about the attached Touch Memory keys. When such a key is applied to the Touch Memory reader, its number and current read time are written to the non-volatile memory of the system, and then this message is transmitted to the server. This packet has a higher priority than sending telemetry records array or the current state, but a lower priority than an out-of-order message.

NTCB packet structure of sending unregistered Touch Memory key

<table>
<tr><td>Message</td><td colspan="2"><HEAD>*>TMKEY<datetime>:<code></td></tr>
<tr><td>Response from the server</td><td colspan="2"><HEAD>*<TMKEY</td></tr>
<tr><td>Exchange channel</td><td colspan="2">GPRS</td></tr>
<tr><td>Designations</td><td>Interpretation</td><td>Data Format</td></tr>
<tr><td><HEAD></td><td>16 byte NTCB packet header with preamble</td><td>16*U8</td></tr>
<tr><td>*>TMKEY</td><td>0x2A 0x3E 0x54 0x4D 0x4B 0x45 0x59</td><td>char[7]</td></tr>
<tr><td>*<TMKEY</td><td>0x2A 0x3C 0x54 0x4D 0x4B 0x45 0x59</td><td>char[7]</td></tr>
<tr><td><datetime></td><td>Event time (record generation) on the device.
Time and date of fixing the event:<table><tr><td>Hour</td><td>0 – 23</td></tr><tr><td>Minute</td><td>0 –59</td></tr><tr><td>Second</td><td>0 – 59</td></tr><tr><td>Day</td><td>1 – 1</td></tr><tr><td>Month</td><td>0 – 1</td></tr><tr><td>Year</td><td>0 – 255 (since 2000)</td></tr></table></td><td>U8 U8 U8 U8 U8 U8</td></tr>
<tr><td><code></td><td>Applied TM device ID in its entirety without device type and without checksum.</td><td>U64</td></tr>
</table>

When using the FLEX 2.0 or FLEX 3.0 protocol, key codes, driver cards, RFID tags are sent only in the form of additional telemetry records. See Annex A.2. Structure of additional FLEX 2.0 telemetry records.

Keys registered in the device are transmitted in telemetry packets as an event indicating the number of the slot in the configuration where the corresponding key is written.
See events 0x1900 - 0x2040 in the "Telematic event codes table" file.

Query code of the last Touch Memory key read by the device

Query	<HEAD>*?TM	
Response	<HEAD>*#TM<key>	
Exchange channel	GPRS, USB, Bluetooth	
Designations	**Interpretation**	**Data Format**
<HEAD>	16 byte NTCB packet header with preamble	16*U8
*?TM	0x2A 0x3F 0x54 0x4D	Char[4]
*#TM	0x2A 0x23 0x54 0x4D	Char[4]
<key>	Key code and checksum is zero when the key was not read.	U64

Query the last active radio tag

Query	<HEAD>*?ERFT	
Response	<HEAD>*#ERFT<ID><pwr>	
Exchange channel	GPRS, USB, Bluetooth	
Designations	**Interpretation**	**Data Format**
<HEAD>	16 byte NTCB packet header with preamble	16*U8
*?ERFT	0x2a 0x3f 0x45 0x52 0x46 0x54	6*U8
*#ERFT	0x2a 0x23 0x45 0x52 0x46 0x54	6*U8
ID	Radio tag ID	U64 (little-endian)
pwr	Radiometer signal power in dBm	S8

Command to edit the list of registered Touch Memory keys

ATTENTION!

List with theTouch Memory key parameters must be loaded into the device in advance. Only one key can be edited at a time.

Command	<HEAD>*!SETTM<s><num>,<addr>,<mode>	
Response	<HEAD>*@SETTM<s><res>[,<code>]	
Exchange channel	GPRS, USB, Bluetooth, SMS	
Designations	**Interpretation**	**Data Format**
<HEAD>	16 byte NTCB packet header with preamble	16*U8
*!SETTM	0x2A 0x21 0x53 0x45 0x54 0x54 0x4D	char[7]
*@SETTM	0x2A 0x40 0x53 0x45 0x54 0x54 0x4D	char[7]
<s>	Parameter separator - space (0x20).	char
<num>	Key number in the configuration: 1-510.	char[]
<addr>	Key address (12 characters).	char[12]
<mode>	Key operating mode: "NOACT" - no action; "GUARD" - changes the security mode; "IMMOB" - immobilizer.	char[]
<res>	Command execution result: "OK" - command was executed successfully; "FAIL" - command failed, error.	char[]
<code>	Additional error code (optional): "1" - module is busy; "2" - incorrect TM key number; "3" - no parameters; "4" - incorrect operating mode.	char

Sending a command causes device to reboot

Command:	~O<module><id><msg_length><message><crc8>	
Negative response	~F<module><command><result><crc8>	
Positive response	~R<module><command><crc8>	
Exchange channel	GPRS	
Designation	**Interpretation**	**Data Format**
~O	0x7E 0x4F	2*U8
~F	0x7E 0x46	2*U8
~R	0x7E 0x52	2*U8
<module>	Module ID: 0x7D - settings editor.	U8
<command>	Command Code: 4 – command to edit Touch Memory keys.	U8
<msg_length>	Length of the message. Up to and including 139 characters.	U8 (little-endian)
<message>	Text message in CP1251 encoding. No terminal zero is required at the end of the message. Content: <num><sp><address><sp><mode><nick_name> Decryption: <sp> - space; <num> - number of the key in the configuration (2 characters always, 1-16, for example "01"); <address> - key address (16 characters); <mode> - key operation mode (5 characters); "NOACT" - no action; "GUARD" - changes the security mode; "IMMOB" - "immobilizer"; <nick_name> - key alias (10 characters).	<msg_length>*U8
<result>	Command Result Code: 0x20 - error: parameters are not set; 0x21 - error in parameter No. 1 0x22 - error in parameter No. 2 0x23 - error in parameter No. 3 0x24 - error in parameter No. 4 0x25 - error in parameter No. 5 0x26 - error in parameter No. 6 0x27 - error in parameter No. 7 0x28 - error in parameter No. 8	U8
<crc8>	FLEX checksum. See Annex B. Checksum calculation algorithm CRC8.	U8

2.8. Working with tachographs

2.8.1. Control commands and queries

Obtaining information that is not recorded in telemetry and uploading DDD files is carried out using commands and requests using the FLEX 2.0 protocol. Information about the number of the inserted driver card is transmitted to the server in additional FLEX 2.0 telemetry packets, see Annex A.2. Structure of additional FLEX 2.0 telemetry records.

Information queries:
1. Query information on the current state of the tachograph;
2. Query information on card No. 1
3. Query information on card No. 2
4. Query registration information.

Control commands:
1. Device authorization in the tachograph;
2. Setting a new device authorization key in the tachograph;

Query the current state of the tachograph

<table>
<tr><td>Query</td><td colspan="2">~Q<module><query><crc8></td></tr>
<tr><td>Positive response</td><td colspan="2">~I<module><query><time><state><cards_state><drivercard1><drivercard2>
<mileageTRIP><voltage><drv1AT><drv1DT><drv1CT><drv1RT><drv2AT>
<drv2DT><drv2CT><drv2RT><crc8></td></tr>
<tr><td>Negative response</td><td colspan="2">~U<module><query><result><crc8></td></tr>
<tr><td>Exchange channel</td><td colspan="2">GPRS</td></tr>
<tr><td>Designation</td><td>Interpretation</td><td>Data Format</td></tr>
<tr><td>~Q</td><td>0x7E 0x51</td><td>2*U8</td></tr>
<tr><td>~I</td><td>0x7E 0x49</td><td>2*U8</td></tr>
<tr><td>~U</td><td>0x7E 0x55</td><td>2*U8</td></tr>
<tr><td><module></td><td>Module ID:
0x81 - tachograph</td><td>U8</td></tr>
<tr><td><query></td><td>Request ID:
0x00 - query information about the current state of the tachograph</td><td>U8</td></tr>
<tr><td><time></td><td>Tachograph time in Unix time format</td><td>U32 (little-endian)</td></tr>
<tr><td><state></td><td>CIPF operation mode:
<table>
<tr><td>Bits</td><td>Description</td><td>Values</td></tr>
<tr><td>0…3</td><td>Tachograph operation mode</td><td>Depends on tachograph</td></tr>
<tr><td>4…7</td><td>CIPF operation mode</td><td>Depends on tachograph</td></tr>
</table></td><td>U8</td></tr>
<tr><td><cards_state></td><td>Status of cards:
<table>
<tr><td>Bits</td><td>Description</td><td>Values</td></tr>
<tr><td>0…3</td><td>Status of card No.</td><td>0 = no card, 1 = not authorized, 2 = authorized, 3 = unable to</td></tr>
<tr><td>4…7</td><td>Status of card</td><td>0 = no card, 1 = not authorized, 2 = authorized, 3 = unable to</td></tr>
</table></td><td>U8</td></tr>
<tr><td><drivercard1></td><td>Card type No. 1 and driver activity No. 1:
<table>
<tr><td>Bits</td><td>Description</td><td>Values</td></tr>
</table></td><td>U8</td></tr>
</table>

	0…3	Driver activity	0 = rest, 1 = availability, 2 = work, 3 =	
	4…7	Card Type	0 = No card; 1 = driver; 2 = Master; 3 = Inspector; 4 = Enterprise	
<drivercard2>	Card type No. 2 and driver activity No. 2 (see Card type No. 1 and driver activity No.1)			U8
<mileageTRIP>	TRIP distance 0.005 km			U32 (little-endian)
<voltage>	On-board power system voltage 0.2 V			U8
<drv1AT>	Time spent by driver No. 1 in the current mode (minutes)			U16 (little-endian)
<drv1DT>	Total driving time of driver No. 1 per day (minutes)			U16 (little-endian)
<drv1CT>	Continuous driving time of driver No. 1 (minutes)			U16 (little-endian)
<drv1RT>	Cumulative break times for driver No. 1 (minutes)			U16 (little-endian)
<drv2AT>	Time spent by driver No. 2 in the current mode			U16 (little-endian)
<drv2DT>	Total driving time of driver No. 2 per day (minutes)			U16 (little-endian)
<drv2CT>	Continuous driving time of driver No. 2 (minutes)			U16 (little-endian)
<drv2RT>	Cumulative break times for driver No. 2 (minutes)			U16 (little-endian)
<result>	Command execution result code			U8
<crc8>	Checksum. See Annex B. Checksum calculation algorithm CRC8.			U8

Query information on cards No. 1 and No. 2

Query	~Q<module><query><crc8>	
Positive response	~I<module><query><type_state><reserved1><issuing><number><reserved2><cp1><text1><cp2><text2><cp3><text3><cp4><text4><crc8>	
Negative response	~U<module><query><result><crc8>	
Exchange channel	GPRS	
Designation	**Interpretation**	**Data Format**
~Q	0x7E 0x51	2*U8
~I	0x7E 0x49	2*U8
~U	0x7E 0x55	2*U8
<module>	Module ID: 0x81 - tachograph	U8
<query>	Request ID: 0x01 - request for information on card No. 1 0x02 - request for information on card No. 2	U8
<type_state>	Card status and type: Bits / Description / Values 0…3 / Map Status / 0 = no card, 1 = not authorized, 2 = authorized, 3 = unable to 4…7 / Card Type / 0 = No card; 1 = driver; 1 = Master; 2 = Inspector; 3 = Enterprise	U8
<reserved1>	Authentication type (reserve = 0)	U8
<issuing>	Country code (Russia - 0x2B)	U8
<number>	Card number (for example, "RUD1000002718000")	16*U8
<reserved2>	Card validity period (reserve = 0)	U32 (little-endian)
<cp1>	ISO 8859 code page number in which string No.1 is transmitted	U8
<cp2>	ISO 8859 code page number in which string No.2 is transmitted	U8
<cp3>	ISO 8859 code page number in which string No.3 is transmitted	U8
<cp4>	ISO 8859 code page number in which string No.4 is transmitted	U8
<text1>	String No.1	35*U8
<text2>	String No.2	35*U8

<text3>	String No.3	35*U8
<text4>	String No.4	35*U8
<result>	Command execution result code	U8
<crc8>	Checksum. See Annex B. Checksum calculation algorithm CRC8.	U8

Query registration information

Query	~Q<module><query><crc8>	
Positive response	~I<module><query><version><vin><nation><vrn_cp><vrn><reserved1><speed_limit><next_calib><activation><expiry><serial><reg_no><crc8>	
Negative response	~U<module><query><result><crc8>	
Exchange channel	GPRS	
Designation	**Interpretation**	**Data Format**
~Q	0x7E 0x51	2*U8
~I	0x7E 0x49	2*U8
~U	0x7E 0x55	2*U8
<module>	Module ID: 0x81 - tachograph	U8
<query>	Request ID: 0x03 - request for registration information of the vehicle and tachograph.	U8
<version>	Tachograph version in text representation	32*U8
<vin>	Vehicle identification number (VIN).	17*U8
<nation>	Code of the country in which the vehicle is registered (Russia - 0x2B)	U8
<vrn_cp>	Code page ISO 8859, which contains the registration number of the vehicle	U8
<vrn>	Vehicle registration number.	13*U8
<reserved1>	Reserve = 0	2*U8
<speed_limit>	Vehicle speed limit (km/h)	U8
<next_calib>	Next tachograph calibration time (Unix time)	U32 (little-endian)
<activation>	CIPF activation time (Unix Time)	U32 (little-endian)
<expiry>	CIPF activation end time (Unix time)	U32 (little-endian)
<serial>	CIPF serial number	16*U8
<reg_no>	CIPF registration number	16*U8
<result>	Command execution result code	U8
<crc8>	Checksum. See Annex B. Checksum calculation algorithm CRC8.	U8

Device authorization in the tachograph

Command	~O<module><command><login><psswrd><crc8>	
Positive response	~R<module><command><crc8>	
Negative response	~F<module><command><result><crc8>	
Exchange channel	GPRS	
Designation	**Interpretation**	**Data Format**
~O	0x7E 0x4F	2*U8
~R	0x7E 0x52	2*U8
~F	0x7E 0x46	2*U8
<module>	Module ID: 0x81 - tachograph	U8
<command>	Command code: 0x00 - Authorization:	U8
<login>	User ID.	3*U8
<psswrd>	Password.	16*U8
<result>	Command execution result code	U8
<crc8>	Checksum. See Annex B. Checksum calculation algorithm CRC8.	U8

Setting a new device authorization key in the tachograph

Command	~O<module><command><old_login><old_psswrd><new_login><new_psswrd><crc8>	
Positive response	~R<module><command><crc8>	
Negative response	~F<module><command><result><crc8>	
Exchange channel	GPRS	
Designation	**Interpretation**	**Data Format**
~O	0x7E 0x4F	2*U8
~R	0x7E 0x52	2*U8
~F	0x7E 0x46	2*U8
<module>	Module ID: 0x81 - tachograph	U8
<command>	Command Code: 0x01 - set a new authorization key:	U8
<old_login>	Old user ID.	3*U8
<old_psswrd>	Old password.	16*U8
<new_login>	New User ID.	3*U8
<new_psswrd>	New Password.	16*U8
<result>	Command execution result code	U8
<crc8>	Checksum. See Annex B. Checksum calculation algorithm CRC8.	U8

2.8.2. Generating and transferring DDD file to the server

Device supports generation of an unloading file (DDD file) containing information on the activities of drivers and operation of the vehicle from a tachograph.
To start generation of a DDD file, device is sent a corresponding command, which indicates type of file, the slot number in which the card is installed and additional parameters, if necessary. It is possible to generate a DDD file with automatic sending to the e-mail specified in the device settings.

Command to start DDD file generation

<table>
<tr><td>Command</td><td colspan="2">~O<module><command><type><param><crc8></td></tr>
<tr><td>Positive response</td><td colspan="2">~ R < module > < command > < size > < crc8 > (see
~R<module><command><size><fn_len><fn><crc8></td></tr>
<tr><td>Negative response</td><td colspan="2">~F<module><command><result><crc8></td></tr>
<tr><td>Exchange channel</td><td colspan="2">GPRS</td></tr>
<tr><td>Designation</td><td>Interpretation</td><td>Data Format</td></tr>
<tr><td>~O</td><td>0x7E 0x4F</td><td>2*U8</td></tr>
<tr><td>~R</td><td>0x7E 0x52</td><td>2*U8</td></tr>
<tr><td>~F</td><td>0x7E 0x46</td><td>2*U8</td></tr>
<tr><td><module></td><td>Module ID:
0x81 - tachograph</td><td>U8</td></tr>
<tr><td><command></td><td>Command Code:
0x02 - generation of the upload file;
0x03 - generation of the upload file with sending to email.</td><td>U8</td></tr>
<tr><td><type></td><td>Type of file to be generated:
0x00 -tachograph (overview);
0x01 -tachograph (activity as of the specified date)
0x02 -tachograph (events and faults)
0x03 -tachograph (speed mode details)
0x04 - tachograph (technical data)
0x05 - tachograph (unloading data from the card)</td><td>U8</td></tr>
<tr><td><param></td><td>Parameter depending on the type of file being generated:
- tachograph (activity on the specified date): date in UNIX format time.

- tachograph (data unloading from the card):
<table>
<tr><td>Bits</td><td>Description</td><td>Values</td></tr>
<tr><td>0</td><td>Slot number</td><td>0x01 - slot No. 1
0x02 - Slot #</td></tr>
<tr><td>1…3</td><td>Mask for elementary files</td><td>Reserve = 0xFFFFFF</td></tr>
</table>
Note: Files with codes 0x00... 0x04 are service. Therefore, on some tachographs and for the driver's card, their unloading is impossible</td><td>U32 (little-endian)</td></tr>
<tr><td><fn_len></td><td>Standard file name length for DDD file up to and including 74 bytes</td><td>U8</td></tr>
<tr><td><fn></td><td>Default name for DDD file (no "\0 "at end)</td><td><fn_len>*U8</td></tr>
<tr><td><result></td><td>Command execution result code</td><td>U8</td></tr>
<tr><td><size></td><td>Size of the generated upload file</td><td>U32 (little-endian)</td></tr>
<tr><td><crc8></td><td>Checksum. See Annex B. Checksum calculation algorithm CRC8.</td><td>U8</td></tr>
</table>

Note: *Devices with older firmware versions (up to 06.10.00) can send an abbreviated response to the command to start generating a DDD file that does not contain the standard name for the generated DDD file (~R module><command><size<crc8>).*

Downloading a DDD file takes an average of no more than 10 minutes. The size of the downloaded file does not exceed 64 KB. The downloaded file is stored in the internal memory of the device. If the device repeatedly receives a command to start generating a DDD file within 30 minutes from the moment of its

last generation, it will not download it from the tachograph again, but uses the file previously saved in non-volatile memory.

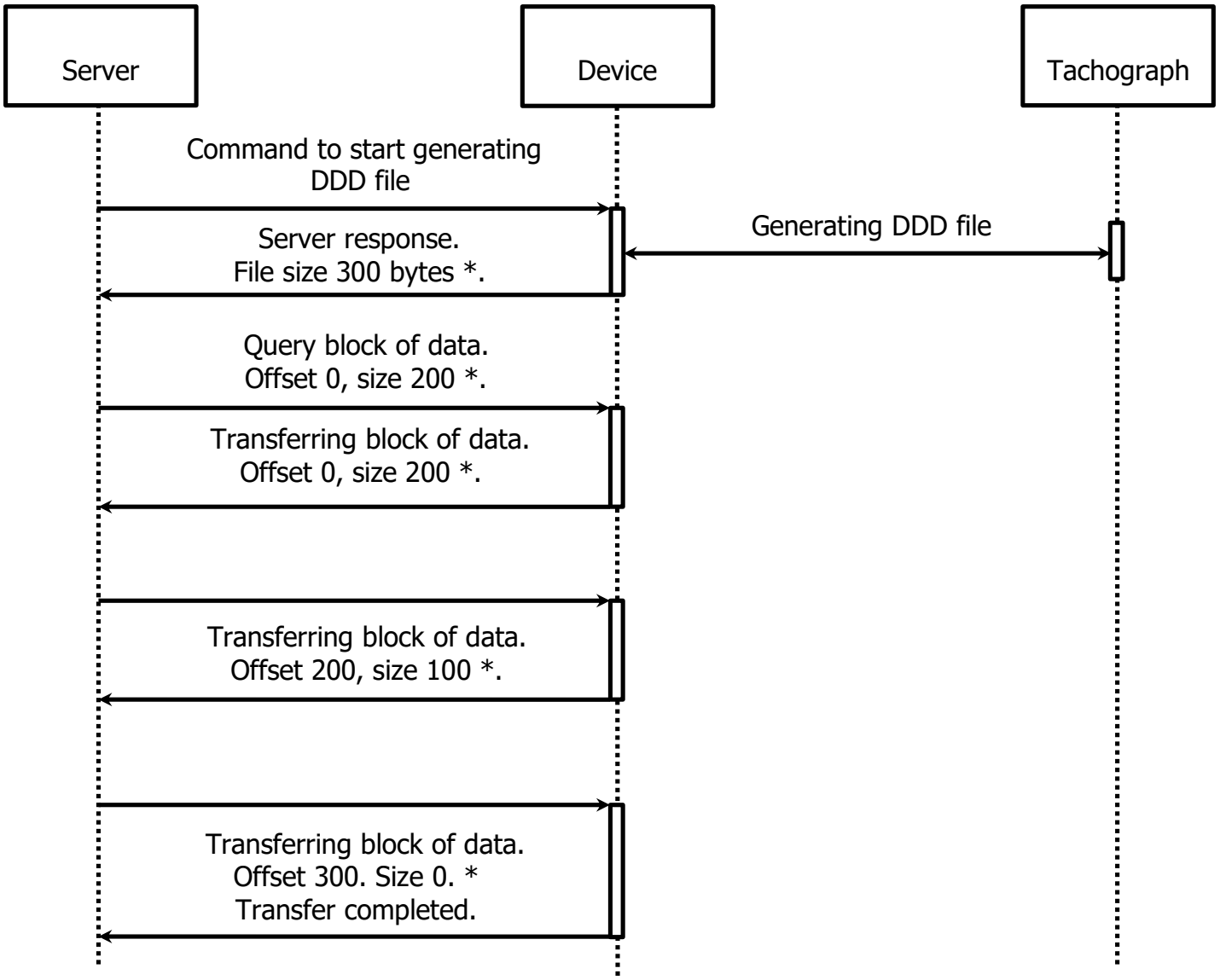
Note: *Non-volatile memory for storing the DDD file will be automatically cleared in the following cases (will download the file again when requesting device generation):*
*- if device recorded the removal of the card from the tachograph (**Important!** Card withdrawal is controlled not by the state of the tachograph, but by the disappearance of the card code).*
- after updating device firmware (due to the fact that one address space is used to store the DDD file and the temporary firmware file).
- after accident file is generated (due to the fact that one address space is used to store the DDD file and accident file).

DDD file transfer is carried out by blocks of arbitrary length at the initiative of the server. The server alternately requests data blocks using an appropriate request, monitors their integrity and transmission order. In each request, the server must specify the number of bytes to read from the DDD file and the offset from its startwhich reading begins. In response, the device sends a block of data indicating the number of bytes that have been read from the file. The maximum block size that a device can transmit at a time is 960 bytes. If the data block was not received by the server, it is possible to request the block again with the same offset and the number of bytes. The file transfer is considered complete when the device sends a block of filewhich the read number of bytes will be 0.

Query DDD file block

Command	~G<module><get_idx><offset><size_need><crc8>	
Positive response	~D<module><get_idx><offset><size_read><data><crc8>	
Negative response	~L<module><get_idx><result><crc8>	
Exchange channel	GPRS	
Designation	**Interpretation**	**Data Format**
~G	0x7E 0x47	2*U8
~D	0x7E 0x44	2*U8
~L	0x7E 0x4C	2*U8
<module>	0x81 - tachograph	U8
<get_idx>	Request ID: 0x00 - query DDD file block	U8
<offset>	Offset from the beginning of the DDD filebytes.	U32 (little-endian)
<size_need>	Number of bytes to read.	U16 (little-endian)
<size_read>	Number of data read in bytes. Read block size may be different from the requested block size, but only downward.	U16 (little-endian)
<data>	Data block	<size_read>*U8
<result>	Command execution result code	U8
<crc8>	Checksum. See Annex B. Checksum calculation algorithm CRC8.	U8

In general, procedure for generating and transferring a DDD file to the server can be represented by the following diagram



2.8.3. Command and query result codes

Code	Description
0x10	command cannot be executed because the previous command has not been completed.
0x20	unknown command/query
0x21	incorrect parameters are specified in the command (for the command to change the authorization key)
0x30	device is not configured to work with tachograph
0x31	no connection with tachograph
0x32	upload file not generated
0x33	failed to log in to the tachograph
0x34	error saving/reading data from non-volatile memory
0x35	DDD generation: trequest contains incorrect parameters (date, card number)
0x36	DDD generation: tachograph cannot generate file in current operating mode
0x37	DDD generation: unsupported file type
0x38	DDD generation: no data to generate file
0x39	DDD Generation error receiving file from tachograph
0x3A	DDD generation: generation aborted (when updating device firmware)
0x3B	DDD generation: file has incorrect structure (failed to generate filename)
0x3C	failed to send file to e-mail
0x70	confirmation of command receipt (for command of upload file generation)

2.9. Working with driver display

Command to send message to driver display via FLEX protocol via GPRS

Command	~O<module><id><index><confirm><msg_length><message><crc8>	
Positive response	~R<module><id><crc8>	
Negative response	~F<module><id><result><crc8>	
Exchange channel	GPRS	
Designation	**Interpretation**	**Data Format**
~O	0x7E 0x4F	2*U8
~F	0x7E 0x46	2*U8
~R	0x7E 0x52	2*U8
<module>	Module ID: 0x82 - driver display DV-01	U8
<id>	Command code: 0 - sending a message to the driver.	U8
<index>	Reserved space for the index assigned to the message by the server (not used at this stage and equal to 0xFFFFFFFF)	U32 (little-endian)
<confirm>	Symbol that determines whether confirmation of message reception is required: ! (0x21) - confirmation required, Other characters - no confirmation required.	U8
<msg_length>	Length of the message. Up to and including 139 characters.	U8 (little-endian)
<message>	Text message to the driver in CP1251 encoding. No terminal zero is required at the end of the message.	<msg_length>*U8
<result>	Command execution result code: 0x01 - command was executed, but the message was truncated due to exceeding the maximum length; 0x10 - device has not yet transmitted the previous message to the display; 0x20 - message length is 0; 0x30 - device is not configured to work with the display; 0x31 - no connection to the display.	U8
<crc8>	FLEX checksum. See Annex B. Checksum calculation algorithm CRC8	U8

Command to send message to driver display via NTCB protocol via GPRS

Command	<HEAD>*!DV<index><confirm><message>	
Response	<HEAD>*@DV<result>	
Exchange channel	GPRS, USB, Bluetooth	
Designation	**Interpretation**	**Data Format**
<HEAD>	16-byte NTCB header	16*U8
*!DV	0x2A 0x21 0x44 0x56	4*U8
*@DV	0x2A 0x40 0x44 0x56	4*U8
<index>	Reserved space for the index assigned to the message by the server (not used at this stage and equal to 0xFFFFFFFF)	U32
<confirm>	A symbol that determines whether confirmation of message reception is required: ! (0x21) - confirmation required, Other characters - no confirmation required.	U8
<message>	Text message to the driver in CP1251 encoding and up to 139 characters inclusive. No terminal zero is required at the end of the message.	N*U8, N is the message length (calculated from the transport layer header)
<result>	Command execution result code:	U8

	0x01 - command was executed, but the message was truncated due to exceeding the maximum length; 0x10 - device has not yet transmitted the previous message to the display; 0x20 - message length is 0; 0x30 - device is not configured to work with the display; 0x31 - no connection to the display.	

2.10. Working with autoinformer

Working with autoinformer is carried out using the NTCB protocol. Notification of events in geozones is transmitted only via USB when the corresponding setting is turned on. Alerts can be used to interact with a mobile application.

Autoinformer Control Command

This command is used both when working via USB and when working via GPRS.

Command	<HEAD>*!AINF<code><data>	
Response	<HEAD>*@AINF<code><cop>	
Exchange channel	GPRS, USB, Bluetooth	
Designations	**Interpretation**	**Data Format**
<HEAD>	16 byte NTCB packet header with preamble	16*U8
*!AINF	0x2A 0x21 0x41 0x49 0x4E 0x46	char[6]
*@AINF	0x2A 0x40 0x41 0x49 0x4E 0x46	char[6]
<code>	Command code: 0x01 - change of the current route; 0x02 - starts playback of the sound file; 0x03 - change of the current movement mode.	U8
<data>	Command dependent data: - change of the current route: route identifier; - start sound file playback: sound file ID; - change of the current driving mode: ID of the driving mode.	U16
<cop>	Parameter indicatingsuccess of the command. Takes the following meanings: 0x00 - command was executed successfully; 0x01 - command was executed, but the settings could not be updated (for route change commands and movement mode); 0x10 - command cannot be executed because execution of the previous command is not completed; 0x20 - unknown command (for command codes greater than 0x03); 0x30 - route cannot be loaded (not in the correct format); 0x31 - failed to load route (route not listed) 0x32 - failed to play sound file; 0x33 - sound file not found; 0x34 - failed to set motion mode (mode not found in the list); 0x35 - failed to load geofence list.	U8

There are text analogues for the specified command:

Command to change the route

Command	<HEAD>*!AINF:<number><char>	
Response	<HEAD>*@AINF:<result>	
Exchange channel	GPRS, USB, Bluetooth	
Designations	**Interpretation**	**Data Format**
<HEAD>	16 byte NTCB packet header with preamble	16*U8
*!AINF	0x2A 0x21 0x41 0x49 0x4E 0x46 0x3A	char[7]
*@AINF	0x2A 0x40 0x41 0x49 0x4E 0x46 0x3A	char[7]
<number>	Route number in text view	cp1251
<char>	Letter of route	cp1251
<result>	Alphabetic code of the command execution result: "S0" (0x53 0x30) - command executed successfully; "S1" (0x53 0x31) - command was executed, but the settings could not be updated (for route change commands and movement mode); "B0" (0x42 0x30) - command cannot be executed because the execution of the previous command is not completed;	char[2]

	"E0" (0x45 0x30) - route cannot be loaded (incorrect format); "E1" (0x45 0x30) - failed to load route (route not listed) "E5" (0x45 0x30) - failed to load geofence list.	

Command to playback sound file

Command	<HEAD>*!AINF!<soundid>	
Response	<HEAD>*@AINF!<result>	
Exchange channel	GPRS, USB, Bluetooth	
Designations	**Interpretation**	**Data Format**
<HEAD>	16 byte NTCB packet header with preamble	16*U8
*!AINF!	0x2A 0x21 0x41 0x49 0x4E 0x46 0x3A 0x21	char[7]
*@AINF!	0x2A 0x40 0x41 0x49 0x4E 0x46 0x3A 0x21	char[7]
<soundid>	Sound file ID in text view	cp1251
<result>	Alphabetic code of the command execution result: "S0" (0x53 0x30) - command executed successfully; "B0" (0x42 0x30) - command cannot be executed because the execution of the previous command is not completed; "E2" (0x45 0x30) - failed to play sound file; "E3" (0x45 0x30) - sound file not found.	char[2]

Command to change speed mode

Command	<HEAD>*!AINF#<spdmodeid>	
Response	<HEAD>*@AINF#<result>	
Exchange channel	GPRS, USB, Bluetooth	
Designations	**Interpretation**	**Data Format**
<HEAD>	16 byte NTCB packet header with preamble	16*U8
*!AINF#	0x2A 0x21 0x41 0x49 0x4E 0x46 0x3A 0x21 0x23	char[7]
*@AINF#	0x2A 0x40 0x41 0x49 0x4E 0x46 0x3A 0x21 0x23	char[7]
<spdmodeid>	Speed mode identifier in text view	cp1251
<result>	Alphabetic code of the command execution result: "S0" (0x53 0x30) - command executed successfully; "S1" (0x53 0x31) - command was executed, but the settings could not be updated (for route change commands and movement mode); "B0" (0x42 0x30) - command cannot be executed because execution of the previous command is not completed; "E4" (0x45 0x30) - failed to set motion mode (mode not found in the list).	char[2]

Autoinformer event alerts

Message	<HEAD>*&AINF<title><id><data>	
Response	Not required	
Exchange channel	GPRS	
Designations	**Interpretation**	**Data Format**
<HEAD>	16 byte NTCB packet header with preamble	16*U8
*&AINF	0x2A 0x26 0x41 0x49 0x4E 0x46	char[6]
<title>	Alert type: 0 - entrance to geofence; 1 - exit from geofence; 2- playback of the sound file.	U8
<id>	Current route ID	U16
<data>	Depends on the type of alert: - entrance to geofence: geofence identifier; - exit to geofence: geofence identifier; - startof sound file playback: sound file ID.	U16

2.10.1. Universal commands for autoinformer

Command to change the route

Message	<HEAD>*!AINF<s>ROUTE,<route>	
Response	<HEAD>*@AINF<s>ROUTE,<result>	
Exchange channel	GPRS, USB, Bluetooth, SMS	
Designations	**Interpretation**	**Format**
<HEAD>	16 byte NTCB packet header	U8[16]
<s>	Field separator - space (0x20).	char
<route>	String containing the number and letter of the route (encoded as windows-1251)	char[]
<result>	Command execution result OK - command executed successfully. BUSY - command cannot be executed because the previous command has not been completed. INVALIDROUTE – route is formed incorrectly. NOROUTE - route not found. NOZONES - failed to load stop list.	char[]

Command to playback sound file

Message	<HEAD>*!AINF<s>PLAY,<sound_id>	
Response	<HEAD>*@AINF<s>PLAY,<result>	
Exchange channel	GPRS, USB, Bluetooth, SMS	
Designations	**Interpretation**	**Format**
<HEAD>	16 byte NTCB packet header	U8[16]
<s>	Field separator - space (0x20).	char
<sound_id>	<sound> - sound file ID (1 to 65535). If 0, stops playing the current sound.	char[]
<result>	OK - command executed successfully. BUSY - command cannot be executed because the previous command has not been completed. INVALID SOUND - failed to play sound file. NO SOUND - sound file not found.	char[]

Command to change autoinformer operating mode

Message	<HEAD>*!AINF<s>MODE,<mode>	
Response	<HEAD>*@AINF<s>MODE,<result>	
Exchange channel	GPRS, USB, Bluetooth, SMS	
Designations	**Interpretation**	**Format**
<HEAD>	16 byte NTCB packet header	U8[16]
<s>	Field separator - space (0x20).	char
<mode>	Autoinformer operating mode: 0 - automatic mode; 1 - manual mode.	char[]
<result>	Command execution result OK - command executed successfully. BUSY - command cannot be executed because the previous command has not been completed. INVALID ROUTE – route is formed incorrectly.	char[]

Command to change the current stop

Message	<HEAD>*!AINF<s>STOP,<stop>	
Response	<HEAD>*@AINF<s>STOP,<result>	
Exchange channel	GPRS, USB, Bluetooth, SMS	
Designations	**Interpretation**	**Format**
<HEAD>	16 byte NTCB packet header	U8[16]
<s>	Field separator - space (0x20).	char
<stop>	Stop ID (1 to 65534). If 0 - exit from the current stop	char[]
<result>	Command execution result OK - command executed successfully. BUSY - command cannot be executed because the previous command has not been completed. INVALID ROUTE – route is formed incorrectly. INVALID MODE - manual mode is not enabled. NOSTOP - stop not found.	char[]

2.11. Working with camera

2.11.1. Control commands and queries

The following commands and queries are used to control the operation of the camera and obtain information about it:

1) Command "Automatic shooting control " - takes the specified number of photos with a specified pause between them;

2) Command "Take a photo" - checks the availability of photos for the specified period of time;

3) Query "Get information about camera" - checks availability of photos for the specified period of time;

Automatic shooting control command

Command	~O<module><cmd_id><param><crc8>	
Positive response	~R<module><cmd_id><crc8>	
Negative response	~F<module><cmd_id><result><crc8>	
Exchange channel	GPRS	
Designation	**Interpretation**	**Data Format**
~O	0x7E 0x4F	2*U8
~F	0x7E 0x46	2*U8
~R	0x7E 0x52	2*U8
<module>	Module ID: 0x80 - digital camera	U8
<cmd_id>	Command code: 0x00 - automatic shooting control.	U8
<param>	Automatic shooting control: 0x00 - enable automatic shooting, 0x01 - disable automatic shooting.	U8
<result>	Command execution result code 2.11.3. Command and query result codes.	U8
<crc8>	Checksum. See Annex B. Checksum calculation algorithm CRC8	U8

Command to take a photo

Command	~O<module><cmd_id><count><delay><crc8>	
Positive response	~R<module><cmd_id><crc8>	
Negative response	~F<module><cmd_id><result><crc8>	
Exchange channel	GPRS	
Designation	**Interpretation**	**Data Format**
~O	0x7E 0x4F	2*U8
~F	0x7E 0x46	2*U8
~R	0x7E 0x52	2*U8
<module>	Module ID: 0x80 - digital camera	U8
<cmd_id>	Command code: 0x01 - take a photo; 0x02 - take a photo and then send it to email.	U8
<count>	Number of photos: 1... 65535	U16 (little-endian)
<delay>	Pause between shots (seconds): 1... 65535	U16 (little-endian)
<result>	Command execution result code 2.11.3. Command and query result codes.	U8
<crc8>	Checksum. See Annex B. Checksum calculation algorithm CRC8	U8

Text analog of commands for working with the camera

Command	<HEAD>*!<cmd>	
Response	<HEAD>*@<Response>	
Exchange channel	GPRS, USB, Bluetooth, SMS	
Designations	**Interpretation**	**Data Format**
<HEAD>	16 byte NTCB packet header with preamble	16*U8
*!	0x2A 0x21	char[2]
*@	0x2A 0x40	char[2]
<cmd>	Command in test form. All commands are supported 4.2.11. Digital camera	char[]
<Response>	Response in text form. See table "Responses to camera control commands" (4.2.11. Digital camera).	char[]

Query to get information about camera

<table>
<tr><td>Query</td><td colspan="2">~Q<module><query><crc8></td></tr>
<tr><td>Positive response</td><td colspan="2">~I<module><query><flags><version><dir><crc8></td></tr>
<tr><td>Negative response</td><td colspan="2">~U<module><query><result><crc8></td></tr>
<tr><td>Exchange channel</td><td colspan="2">GPRS</td></tr>
<tr><td>Designation</td><td>Interpretation</td><td>Data Format</td></tr>
<tr><td>~Q</td><td>0x7E 0x51</td><td>2*U8</td></tr>
<tr><td>~I</td><td>0x7E 0x49</td><td>2*U8</td></tr>
<tr><td>~U</td><td>0x7E 0x55</td><td>2*U8</td></tr>
<tr><td><module></td><td>Module ID:
0x80 - digital camera</td><td>U8</td></tr>
<tr><td><query></td><td>Request ID:
0x00 - get camera information.</td><td>U8</td></tr>
<tr><td><flags></td><td>
<table>
<tr><th>Bits</th><th>Description</th><th>Values</th></tr>
<tr><td>0</td><td>Automatic shooting</td><td>0 - disabled,
1 - enabled</td></tr>
<tr><td>1-7</td><td>Reserve</td><td>0</td></tr>
</table>
</td><td>U8</td></tr>
<tr><td><version></td><td>Camera version (16 characters + "\0 "):
For example, "VC0706 1.00."</td><td>17*U8</td></tr>
<tr><td><dir></td><td>Directory where the photos are stored (8 characters + "\0 "):
For example, "PHOTOS."</td><td>9*U8</td></tr>
<tr><td><result></td><td>Command execution result code 2.11.3. Command and query result codes</td><td>U8</td></tr>
<tr><td><crc8></td><td>Checksum. See attachment Annex B. Checksum calculation algorithm CRC8.</td><td>U8</td></tr>
</table>

2.11.2. Transferring photos to the server

Photos stored on the device are identified by the UTC of their creation time, recorded in unsigned Unix-time format (number of seconds from 00:00:00 on January 1, 1970). Each main photo with size of 640×480 or 320×240, depending on the settings, has a corresponding overview photo with size of 160×120, which is a reduced copy of the main photo.

To quickly notify the server about the appearance of a new photo, device sends "New photo notofocation" to the server every time a new photo is created every 10 minutes and after establishing communication with the server.

New photo notification

Notification	~N<module><id><time><crc8>	
Response	**Not required**	
Exchange channel	GPRS,	
Designation	**Interpretation**	**Data Format**
~N	0x7E 0x4E	2*U8
<module>	Module ID: 0x80 - digital camera.	U8
<id>	Notification code: 0x00 - notification about creation of a new photo.	U8
<time>	Date and time of creation of the last photo: Unix-time (number of seconds from 00:00:00 January 1, 1970) in unsigned format.	U32 (little-endian)
<crc8>	Checksum. See Annex B. Checksum calculation algorithm CRC8	U8

Information on the size of the main and overview photos, on the presence of photos for a certain period of time can be obtained using the query "Query information about the photo." Query specifies the time for which the search is performed and the search mode. Response to the query contains the photo creation time, which is the closest to the search time and meets the search criteria.

Example of getting a list of files for period from A to B:

1. Send a query with search time A and search mode "after the specified time inclusive." If in the response the time of taking a photo C exceeds the time B, then the pictures were not found and search was completed, otherwise the information about the picture is entered in the list of search results;
2. Send a query with search time C and search mode "after the specified time not inclusive." If in the response the time of taking a photo D exceeds the time B, then search is completed, otherwise the information about the photo is entered in the list of search results and the query is repeated, but with the search time D;

Query photo information

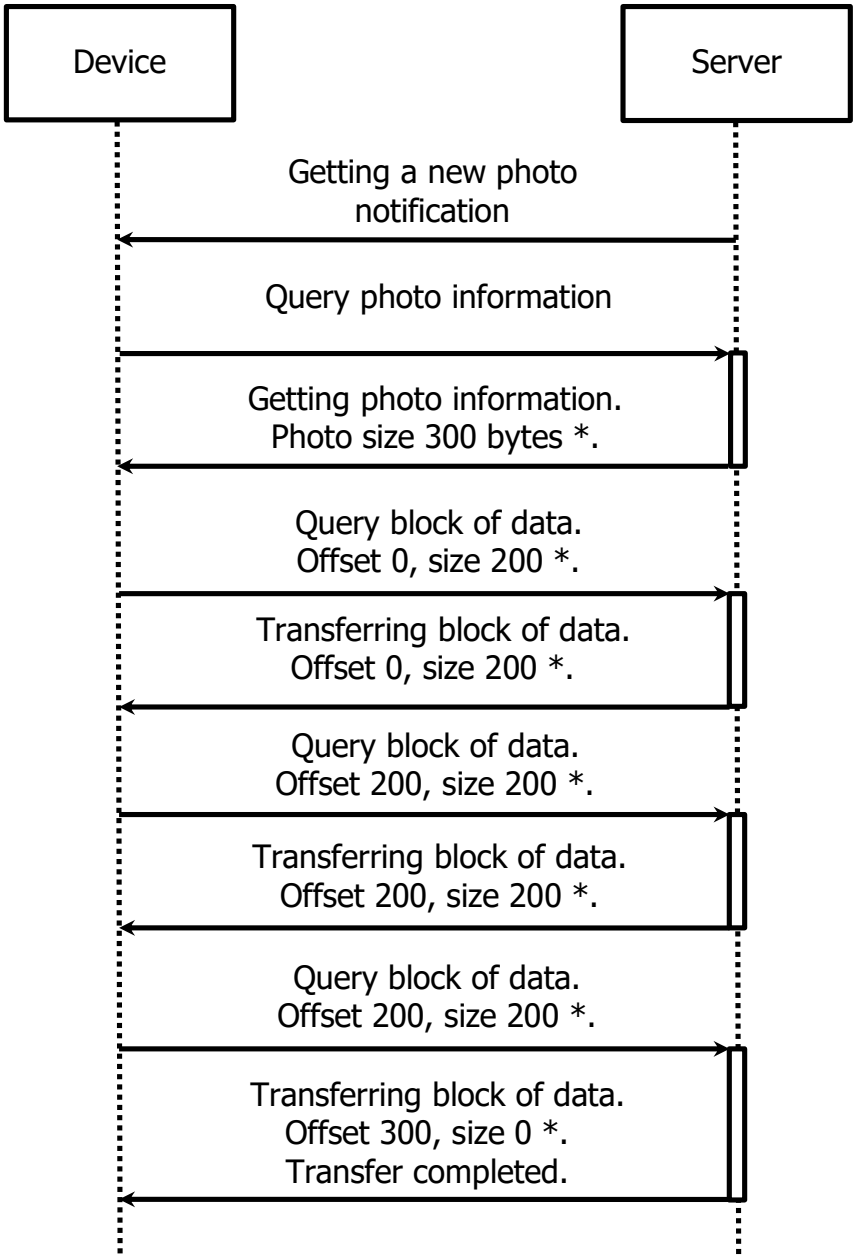
<table>
<tr><td>Query</td><td colspan="2">~Q<module><query><flags><search_time><crc8></td></tr>
<tr><td>Positive response</td><td colspan="2">~I<module><query><time><size_fair><size_rough><crc8></td></tr>
<tr><td>Negative response</td><td colspan="2">~U<module><query><result><crc8></td></tr>
<tr><td>Exchange channel</td><td colspan="2">GPRS</td></tr>
<tr><td>Designation</td><td>Interpretation</td><td>Data Format</td></tr>
<tr><td>~Q</td><td>0x7E 0x51</td><td>2*U8</td></tr>
<tr><td>~I</td><td>0x7E 0x49</td><td>2*U8</td></tr>
<tr><td>~U</td><td>0x7E 0x55</td><td>2*U8</td></tr>
<tr><td><module></td><td>Module ID:
0x80 - digital camera</td><td>U8</td></tr>
<tr><td><query></td><td>Request ID:
0x01-query photo information.</td><td>U8</td></tr>
<tr><td><flags></td><td>
<table>
<tr><td>Bits*</td><td>Description</td><td>Values</td></tr>
<tr><td>0</td><td>Search for a photo whose creation time is the same as the time <search_time></td><td>0 - no,
1 - yes</td></tr>
<tr><td>1</td><td>Search for a photo taken before <search_time> (not inclusive)</td><td>0 - no,
1 - yes</td></tr>
<tr><td>2</td><td>Search for a photo taken after <search_time> (not inclusive)</td><td>0 - no,
1 - yes</td></tr>
<tr><td>3-7</td><td>Reserve</td><td></td></tr>
</table>
* Bit values can be combined to obtain additional search conditions. For example, the combination of bit 0 and bit 1 forms the search condition "Search for a photo taken before <search_time> inclusive".
</td><td>U8</td></tr>
<tr><td><search_time></td><td>Search date and time: Unix-time (number of seconds from 00:00:00 January 1, 1970) in unsigned format.</td><td>U32 (little-endian)</td></tr>
<tr><td><time></td><td>Date and time when the found photo was created. (If a photo is not found, a response is sent ~ U with the appropriate code.)</td><td>U32 (little-endian)</td></tr>
<tr><td><size_fair></td><td>Primary photo size (bytes).</td><td>U16 (little-endian)</td></tr>
<tr><td><size_rough></td><td>Size of the overview photo (bytes).</td><td>U16 (little-endian)</td></tr>
<tr><td><result></td><td>Command execution result code 2.11.3. Command and query result codes.</td><td>U8</td></tr>
<tr><td><crc8></td><td>Checksum. See Annex B. Checksum calculation algorithm CRC8</td><td>U8</td></tr>
</table>

Photos are sent to the server in blocks using the "Query photo data" command. The command specifies the type of photo (main or overview), the time the photo was taken, the offset in bytes from the beginning of the photo, and the size of the block for transmission, so that arbitrary access to the photo data is possible. The latter is necessary when organizing a photo upload and re-requesting photo data blocks. The maximum block size that a device can transmit at a time is 960 bytes.

Query photo data command

Command	~G<module><get_id><utc_time><offset><size_need><crc8>	
Positive response	~D<module><get_id><utc_time><offset><size_read><data><crc8>	
Negative response	~L<module><get_id><result><crc8>	
Exchange channel	GPRS	
Designation	**Interpretation**	**Data Format**
~G	0x7E 0x47	2*U8
~D	0x7E 0x44	2*U8
~L	0x7E 0x4C	2*U8
<module>	Module ID: 0x80 - digital camera	U8
<get_id>	Request ID: 0x00 - query overview photo data; 0x01- querie the main photo data.	U8
<utc_time>	Time to create the required photo: Unix-time (number of seconds from 00:00:00 on January 1, 1970) in unsigned format.	U32 (little-endian)
<offset>	Offset in bytes from the start of the photo file.	U16 (little-endian)
<size_need>	Size of the requested data block in bytes.	U16 (little-endian)
<size_read>	Size of the read photo data block in bytes. Read block size may be different from the requested block size, but only downward.	U16 (little-endian)
<data>	Photo data block.	<size_read>*U8
<result>	Command execution result code 2.11.3. Command and query result codes.	U8
<crc8>	Checksum.	U8

In general, procedure for sending photos to the server can be represented by the following diagram:



*offsets and sizes are given as an example

Completion of the photo transfer is determined based on the size of the photo, offset from the beginning of the photo and the length of the resulting block. When sequentially requesting data, download is considered complete if response is received from the device with a block length of 0.

2.11.3. Command and query result codes

Code	Description
0x10	command cannot be executed because the previous command has not been completed
0x20	unknown command
0x30	device is not configured to work with camera
0x31	no communication with the camera
0x32	automatic shooting is disabled
0x33	SD card error
0x34	photo not found
0x35	error getting picture from camera
0x36	failed to send photo to e-mail

2.12. Communication between external interfaces and server

Data transfer command from device to server via USB

From application to device

Command	<HEAD>*!U2S<data>	
Response	<HEAD>*@U2S<cop>	
Exchange channel	GPRS, USB	
Designations	**Interpretation**	**Data Format**
<HEAD>	16 byte NTCB packet header with preamble	16*U8
*!U2S	0x2A 0x21 0x55 0x32 0x53	char[5]
*@U2S	0x2A 0x40 0x55 0x32 0x53	char[5]
<data>	Array of binary data between 1 and 1003 bytes. Amount of data is calculated from the transport layer header.	U8 * size_of_data
<cop>	Parameter indicating the successfully executed command. Takes the following meanings: 0x31 - command executed, data sent; 0x32 - command cannot be executed, server is not configured; 0x33 - command cannot be executed, server is not available; 0x34 - command cannot be executed, server returned an error, or is not responding to a message.	U8

From device to server

Message	<HEAD>*>U2S<data>	
Response	<HEAD>*<U2S<cop>	
Exchange channel	GPRS, USB	
Designations	**Interpretation**	**Data Format**
<HEAD>	16 byte NTCB packet header with preamble	16*U8
*!U2S	0x2A 0x21 0x55 0x32 0x53	char[5]
*@U2S	0x2A 0x40 0x55 0x32 0x53	char[5]
<data>	Array of binary data between 1 and 1003 bytes. Amount of data is calculated from the transport layer header.	U8 * size_of_data
<cop>	Parameter indicating the successfully executed command. Takes the following meanings: 0x31 - command executed, data received; 0x34 - receive error.	U8

Data transfer command from server via USB

From server to device

Command	<HEAD>*!S2U<data>	
Response	<HEAD>*@S2U<cop>	
Exchange channel	GPRS, USB	
Designations	**Interpretation**	**Data Format**
<HEAD>	16 byte NTCB packet header with preamble	16*U8
*!S2U	0x2A 0x21 0x53 0x32 0x55	char[5]
*@S2U	0x2A 0x40 0x53 0x32 0x55	char[5]
<data>	Array of binary data between 1 and 1003 bytes. Amount of data is calculated from the transport layer header.	U8 * size_of_data
<cop>	Parameter indicating the successfully executed command. Takes the following meanings: 0x31 - command executed, data sent; 0x33 - command cannot be executed, USB is not connected; 0x34 - command cannot be executed, host returned an error, or does not respond to a message.	U8

From device to application

Message	<HEAD>*>S2U<data>	
Response	<HEAD>*<S2U<cop>	
Exchange channel	GPRS, USB	
Designations	**Interpretation**	**Data Format**
<HEAD>	16 byte NTCB packet header with preamble	16*U8
*>S2U	0x2A 0x3E 0x53 0x32 0x55	char[5]
*<S2U	0x2A 0x3C 0x53 0x32 0x55	char[5]
<data>	Array of binary data between 1 and 1003 bytes. Amount of data is calculated from the transport layer header.	U8 * size_of_data
<cop>	Parameter indicating the successfully executed command. Takes the following meanings: 0x31 - command executed; 0x34 - receive error.	U8

Data transfer command from device to server via USB, RS232, RS485

Message	~P<module><put_id><data_length><data><crc8>	
Positive response	~M<module><put_id><crc8>	
Negative response	~S<module><put_id><result><crc8>	
Exchange channel	GPRS	
Designation	**Interpretation**	**Data Format**
~P	0x7E 0x50	2*U8
~S	0x7E 0x53	2*U8
~M	0x7E 0x4D	2*U8
<module>	Module ID: 0x7F - repeater RS232/RS485/USB	U8
<put_id>	Message ID: 0x00 - data transfer in transparent mode from USB. 0x01 - data transfer in transparent mode from RS232. 0x02 - data transfer in transparent mode from RS485. **Note:** Devices with older firmware versions (up to 7.00.00) only support messages with code 0x00.	U8
<data_length>	Size of the data block transmitted by the device. From 1 to 512 bytes inclusive.	U16 (little-endian)
<data>	Data block transmitted by the device.	<data_length>*U8
<result>	Parameter indicating the successfully executed command. Takes the following meanings: 0x31 - successful execution of the command; 0x34 - command cannot be executed, receive failed.	U8
<crc8>	Checksum. See Annex B. Checksum calculation algorithm CRC8	U8

Data transfer command from server to USB, RS232, RS485 devices

Message	~P<module><put_id><data_length><data><crc8>	
Positive response	~M<module><put_id><crc8>	
Negative response	~S<module><put_id><result><crc8>	
Exchange channel	GPRS	
Designation	**Interpretation**	**Data Format**
~P	0x7E 0x50	2*U8
~S	0x7E 0x53	2*U8
~M	0x7E 0x4D	2*U8
<module>	Module ID: 0x7F - repeater RS232/RS485/USB	U8
<put_id>	Message ID: 0x00 - data transfer in transparent mode to USB. 0x01 - data transfer in transparent mode to RS232. 0x02 - data transfer in transparent mode to RS485. **Note:** Devices with older firmware versions (up to 7.00.00) only support messages with code 0x00.	U8
<data_length>	Size of the data block transmitted by the device. From 1 to 512 bytes inclusive.	U16 (little-endian)
<data>	Data block transmitted by the device.	<data_length>*U8
<result>	Parameter indicating the successfully executed command. Takes the following meanings: 0x30 or 0x31 - successful execution of the command; 0x33 - command cannot be executed, USB, RS232, RS485 not connected; 0x34 - command cannot be executed, receive error; 0x35 - command cannot be executed, previous command is processed.	U8
<crc8>	Checksum. See Annex B. Checksum calculation algorithm CRC8.	U8

Custom command

This command allows you to send an arbitrary data string to the device interface or internal module. After sending data, device "listens to" the interface for 500 ms and returns a response without formatting. If no response is received within 500 ms, response field will be empty.

Command	<HEAD>*!UC<s><i>,<msg> Example: <HEAD>*!UC RS485,3101066C	
Response	<HEAD>*@UC<s><i>,<ans>	
Exchange channel	GPRS, USB, Bluetooth, SMS	
Designations	**Interpretation**	**Data Format**
<HEAD>	16 byte NTCB packet header with preamble	16*U8
*!UC	0x2A 0x21 0x55 0x43	char[4]
*@UC	0x2A 0x40 0x55 0x43	char[4]
<s>	Parameter separator - space (0x20).	char
<i>	Interface: – "RS485" - interface of RS-485 device, – "RS232" – interface of RS-232 device, – "GPS" - built-in navigation module.	char[]
<msg>	For RS232, RS485: Packet as HEX string (ASCII). For GPS: Packet in NMEA protocol without checksum (device adds checksum).	char[]
<ans>	If response was received - HEX String (ASCII); If response was not received - empty string.	char[]

2.13. Working with CAN-LOG module

Setting CAN-LOG device program number

Message	<HEAD>*!CANLOG<ver>	
Response from the server	<HEAD>*@CANLOG<cop>	
Exchange channel	GPRS, USB, Bluetooth, SMS	
Designations	**Interpretation**	**Data Format**
<HEAD>	16 byte NTCB packet header with preamble	16*U8
*!CANLOG	0x2A 0x21 0x43 0x41 0x4E 0x4C 0x4F 0x47	char[8]
*@CANLOG	0x2A 0x40 0x43 0x41 0x4E 0x4C 0x4F 0x47	char[8]
<ver>	Text version of the required CAN-LOG program For example, "123"	char[]
<cop>	Result of the program change operation: 0x31 - program is set; 0x32 - incorrect program designation; 0x33 - module does not respond; 0x34 - device is not configured to work with CAN-LOG.	U8

Query CAN-LOG device program number

Message	<HEAD>*?CANLOG	
Response from the server	<HEAD>*#CANLOG<s><result>[,<ver>]	
Exchange channel	GPRS, USB, Bluetooth, SMS	
Designations	**Interpretation**	**Data Format**
<HEAD>	16 byte NTCB packet header with preamble	16*U8
*?CANLOG	0x2A 0x3F 0x43 0x41 0x4E 0x4C 0x4F 0x47	char[8]
*#CANLOG	0x2A 0x23 0x43 0x41 0x4E 0x4C 0x4F 0x47	char[8]
<s>	Parameter separator - space (0x20).	char
<ver>	Text version of the required CAN-LOG program For example, "123"	char[]
<result>	Result of the program change operation: 0x31 - completed; 0x33 - module does not respond; 0x34 - device is not configured to work with CAN-LOG.	U8

2.14. Working with built-in accelerometer

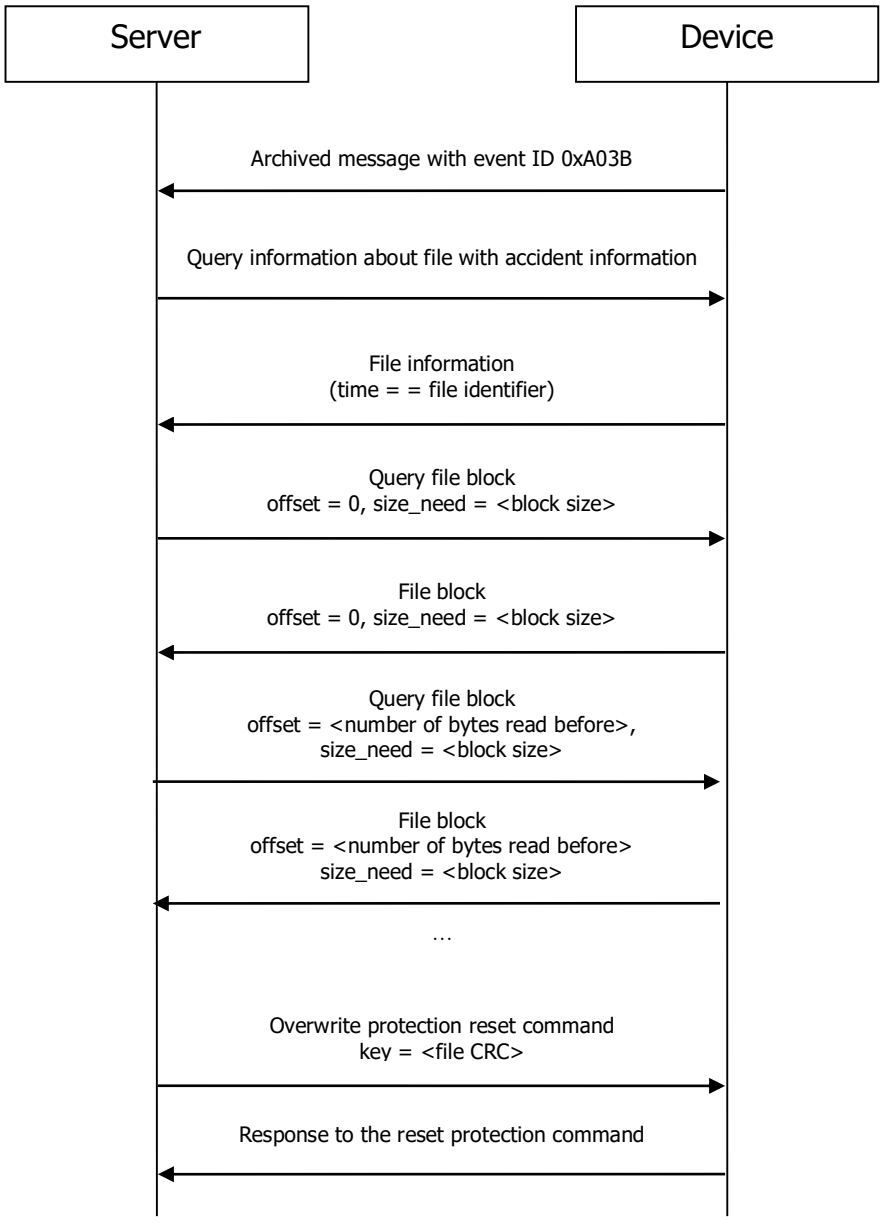
Accelerometer calibration command

Message	<HEAD>*!ACL_C:<command><param>	
Response	<HEAD>*@ACL_C:<command><result>	
Exchange channel	GPRS, USB, Bluetooth	
Designations	**Interpretation**	**Data Format**
<HEAD>	16 byte NTCB packet header with preamble	16*U8
*!ACL_C:	0x2a 0x21 0x41 0x43 0x4c 0x5f 0x43 0x3a	6*U8
*@ACL_C:	0x2a 0x40 0x41 0x43 0x4c 0x5f 0x43 0x3a	6*U8
<command>	"G" (0x47) - calibration (calibration check) of the accelerometer.	U8
<param>	Additional parameter that explains how the command works. 0x30 - start accelerometer calibration; 0x31 - start checking of accelerometer calibration; 0x32 – canceling previously started calibration; 0x33 - start accelerometer calibration according to GNSS data; 0x34 - erasing current calibration.	U8
<result>	Result of the command execution. 0x30 - unknown command; 0x31 - command executed successfully; 0x32 - error: car was moving or vibrating strongly; 0x33 - dynamic calibration started; 0x34 - error: car did not start; 0x35 - calibration completed, check if direction is correct; 0x36 - error: insufficient number and/or intensity of acceleration/braking; 0x37 - dynamic checking started; 0x38 - calibration must be repeated; 0x39 - calibration is correct; 0x3A - error: failed to measure gravity; 0x3B - GNSS calibration started.	U8

2.15. Accident detection

It is possible to understand that the file with information about the accident was generated by the device on two factors:
1. When receiving an archival (alarm) message with an event code 0xA03B - Event of completion of the generation of a file with accident profile;
2. When getting response to the query "Query information about file with information about the accident" that accident was detected (time and length fields are not zero).

Sequence of commands for uploading a file is schematically shown in the figure below:



Query information about detected accident

Message	<HEAD>*?KRAI	
Response	<HEAD>*#KRAI<time><size><flags><filename>	
Exchange channel	GPRS, USB, Bluetooth	
Designations	**Interpretation**	**Data Format**
<HEAD>	16 byte NTCB packet header with preamble	16*U8
*?KRAI	0x2a 0x3f 0x4b 0x52 0x41 0x49	U8[6]
*#KRAI	0x2a 0x23 0x4b 0x52 0x41 0x49	U8[6]
<time>	UTC time of accident detection in unix time format. 0 - accident was not detected.	U32 (little-endian)
<size>	Size of the file with information about the accident. 0 -file not generated.	U32 (little-endian)
<flags>	Overwrite protection mode: 0 - protection mode is disabled; Other values - protection mode is enabled.	U8
<filename>	Name (including extension) with which you want to save the file. Contains a terminal zero at the end. *Size is calculated from the transport layer header*	char[]

Command to reset overwrite protection

Message	<HEAD>*!KRAI<key>	
Response	<HEAD>*@KRAI<result>	
Exchange channel	GPRS, USB, Bluetooth	
Designations	**Interpretation**	**Data Format**
<HEAD>	16 byte NTCB packet header with preamble	16*U8
*!KRAI	0x2a 0x21 0x4b 0x52 0x41 0x49	6*U8
*@KRAI	0x2a 0x40 0x4b 0x52 0x41 0x49	6*U8
<key>	Key for resetting accident information. Equal to the value of block "10. Checksum "from the file.	U32 (little-endian)
<result>	Command execution result: 0x31 - command executed successfully; 0x32 - error working with internal memory; 0x33 - invalid reset key specified; 0x35 - no accident data, no reset required.	U8

Command to read a file page from non-volatile device memory

Command	<HEAD>*!FR<file><time><offset><size>	
Response	<HEAD>*@FR<cop><file><time><offset><size><fileSize><y>	
Exchange channel	GPRS, USB, Bluetooth	
Designations	**Interpretation**	**Data Format**
<HEAD>	16 byte NTCB packet header with preamble	16*U8
*!FR	0x2A 0x21 0x46 0x52	char[4]
*@FR	0x2A 0x40 0x46 0x52	char[4]
<file>	Recorded file type: 3 - accident file	U8
<time>	It is a unique file identifier created by the device. Needed to track the exchange failure when changing a file. The first packet comes with time equal to 0, then time is taken from response.	U32 (little-endian)
<offset>	Offset from file beginning (in bytes)	U32
<size>	Size of the requested data (in bytes). Response returns number of the bytes read. **At the moment, the constant is 512 bytes.**	U16
<fileSize>	Total size of the file in bytes.	U32
<y>	Data read from the non-volatile memory page in the appropriate data storage format (see below).	U8 * <size>
<cop>	Parameter indicating the successfully executed command. Takes the following meanings: 0x31 – reading completed successfully; 0x32 - error occurred while reading page (data integrity is broken); 0x33 - invalid offset in file; 0x34 - file is missing.	U8

Format of file that stores accident information

<table>
<tr><th>File Block</th><th>Size</th><th>Description</th></tr>
<tr><td>1. Prefix</td><td>uint32_t</td><td>“RAIF”</td></tr>
<tr><td>2. File version</td><td>uint16_t</td><td>0x0101</td></tr>
<tr><td>3. File size</td><td>uint16_t</td><td>Total block size 4 - 7.</td></tr>
<tr><td>4. Accident detection time</td><td></td><td></td></tr>
<tr><td>4.1. Accident detection time</td><td>uint32_t</td><td>UTC time of accident detection in unix time format</td></tr>
<tr><td>4.2. Precision timer readings</td><td>uint16_t</td><td>Precision timer readings at the moment of accident detection (0... 65535</td></tr>
<tr><td colspan="3">5. Navigation information</td></tr>
<tr><td>5.1. Status of GPS/GLONASS navigation sensor</td><td>uint8_t</td><td>Bit field:
<table>
<tr><td>Categories</td><td>Values</td></tr>
<tr><td>0</td><td>0 - navigation receiver is off;
1 - navigation receiver is on.</td></tr>
<tr><td>1</td><td>0 - invalid navigation;
1 - valid navigation.</td></tr>
<tr><td>2..7</td><td>Number of navigation satellites 0-31</td></tr>
</table></td></tr>
<tr><td>5.2. Time</td><td>uint32_t</td><td>Unix last valid navigation time</td></tr>
<tr><td>5.3. Latitude</td><td>uint32_t</td><td>Latitude angle recorded when obtaining the last valid coordinates. In ten thousandths of a minute.</td></tr>
<tr><td>5.4. Longitude</td><td>uint32_t</td><td>Longitude angle recorded when obtaining the last valid coordinates. In ten thousandths of a minute.</td></tr>
<tr><td>5.5. Altitude</td><td>int32_t</td><td>Altitude relative to sea level recorded when obtaining the last valid coordinates. In decimeters.</td></tr>
<tr><td>5.6. Speed</td><td>float</td><td>Speed recorded when obtaining the last valid coordinates. In km/h.</td></tr>
<tr><td>5.7. Course</td><td>uint16_t</td><td>Course recorded when obtaining the last valid coordinates. 0° … 360°.</td></tr>
<tr><td colspan="3">6. Accelerometer status</td></tr>
<tr><td>6.1. Current calibration code</td><td>uint8_t</td><td>0xAA - accelerometer is calibrated by displacement;
0XBB - accelerometer is calibrated along axes;
Other values - accelerometer is not calibrated.</td></tr>
<tr><td colspan="3">6.2. Accelerometer basis</td></tr>
<tr><td>6.2.1. X axis</td><td>int16_t[3]</td><td>Along X, Y, Z axes. Scale factor 16 384
To get a real value, the value stored in the file must be divided into a scale factor.</td></tr>
<tr><td>6.2.2. Y axis</td><td>int16_t[3]</td><td>see 6.2.1</td></tr>
<tr><td>6.2.3. Z axis</td><td>int16_t[3]</td><td>see 6.2.1</td></tr>
<tr><td>6.2.4. Offset</td><td>int16_t[3]</td><td>see 6.2.1</td></tr>
<tr><td>6.3. Total number of recorded acceleration points</td><td>uint16_t</td><td></td></tr>
<tr><td>6.4. Accident detection point in the data array from the accelerometer</td><td>uint16_t</td><td></td></tr>
<tr><td>6.5. Axis accelerations</td><td>struct {
int16_t x;
int16_t y;
int16_t z;
} []</td><td>Data array from accelerometer along X, Y, Z axes in m/s^2 with scale factor 128.
To get a real value, the value stored in the file must be divided into a scale factor.</td></tr>
<tr><td colspan="3">7. Speed data</td></tr>
<tr><td>7.1. Total number of recorded velocity points</td><td>uint16_t</td><td></td></tr>
<tr><td>7.2. Accident detection point in the speed data array</td><td>uint16_t</td><td></td></tr>
<tr><td>7.3. Speed data (point validity and speed value)</td><td>struct {
uint8_t valid;
uint8_t speed;
} []</td><td>Speed data array:
valid - validity of point coordinates (1 - valid, 0 - not valid);
speed - speed km/h at a point.</td></tr>
<tr><td>8. Checksum</td><td>uint32_t</td><td>Checksum CRC32 with polynomial 0x77073096, original 0xFFFFFFFF value and without inversion of the result bits. Calculation is performed for units 4...7.</td></tr>
</table>

2.16. Working with temperature sensors 1-Wire

Query temperature sensors

Query	<HEAD>*?TM<s><rt>	
Response	<HEAD>*#TM<s><rt>,<num>:<tkey>,<num>:<tkey>,…	
Exchange channel	GPRS, USB, Bluetooth, SMS	
Designations	**Interpretation**	**Data Format**
<HEAD>	16 byte NTCB packet header with preamble	16*U8
*?TM	0x2A 0x3F 0x54 0x4D	Char[4]
*#TM	0x2A 0x23 0x54 0x4D	Char[4]
<s>	Parameter separator - space (0x20).	char
<rt>	Query type: TSV - query of all visible temperature sensors; TSU - quert of all visible unregistered temperature sensors.	char[]
<num>	Sensor sequence number in text form (for example, "1")	char[2]
<tkey>	Sensor code as HEX string	char[12]
<key>	Key code without family digits and checksum is zero when the key was not read.	U64

Command for temperature sensors

Query	<HEAD>*!TM<s><ct>	
Response	<HEAD>*@TM<s><ct>,<num>:<tkey>,<num>:<tkey>,…	
Exchange channel	GPRS, USB, Bluetooth, SMS	
Designations	**Interpretation**	**Data Format**
<HEAD>	16 byte NTCB packet header with preamble	16*U8
*!TM	0x2A 0x21 0x54 0x4D	Char[4]
*@TM	0x2A 0x40 0x54 0x4D	Char[4]
<s>	Parameter separator - space (0x20).	char
<ct>	Command type: SAVE - save unregistered temperature sensors to configuration.	char[]
<num>	Sensor sequence number in text form (for example, "1")	char[2]
<tkey>	Sensor code as HEX string	char[12]

2.17. Working with passenger flow counters

Command for "Autoconductor" counter

Command	<HEAD>*!ACOND<s><cmd>	
Response	<HEAD>*@ACOND<s><ans>[,<cmd>]	
Exchange channel	GPRS, USB, Bluetooth, SMS	
Designations	**Interpretation**	**Data Format**
<HEAD>	16 byte NTCB packet header with preamble	16*U8
*!ACOND	0x2A, 0x21, 0x41, 0x43, 0x4F, 0x4E, 0x44	char[7]
*@ACOND	0x2A, 0x40, 0x41, 0x43, 0x4F, 0x4E, 0x44	char[7]
<s>	Parameter separator - space (0x20).	char
<cmd>	Command (ASCII): "RESET" - reboot.	char[]
<ans>	Command response (ASCII): "NOINIT" - device is not configured to work with the counter; "FAIL" - unknown command; "OK" - command is accepted.	char

2.18. Working with refrigeration units

Query additional information

Query	~Q<module><query>	
Response	~I<module><query><reftype><ver><btname><sn><refsn><trsn>	
Exchange channel	GPRS	
Designation	**Interpretation**	**Data Format**
~Q	0x7E 0x51	U8[2]
~I	0x7E 0x49	U8[2]
<module>	0x7A - iQFreeze polling module.	U8
<query>	Request ID: 0x00 - query information about iQFreeze.	U8
<reftype>	Installation type	U16
<ver>	iQFreeze firmware version (with terminal zero at the end)	char[19+1]
<btname>	iQFreeze bluetooth name (with terminal zero at the end)	char[19+1]
<sn>	iQFreeze serial number (with terminal zero at the end)	char[19+1]
<refsn>	Installation serial number (with terminal zero at the end)	char[19+1]
<trsn>	Trailer serial number (with terminal zero at the end)	char[19+1]
<crc8>	Checksum (Navtelecom Communication Protocol. Version 5.3. Annex B. Checksum calculation algorithm CRC8)	U8

2.19. Configuration commands

For a detailed description of commands and general rules for working with them, see "6. Working with device configuration".

Writing configuration

Sending a command causes the device to reboot

Command	<HEAD>*!EDITS<s><p>:<t>(<a1>,<a2>,…<aX>) Example: <HEAD>*!EDITS TRANS:SRV1(,,,193.193.165.165,20966)	
Response	<HEAD>*@EDITS<s><result>,<p>:<t>(<a1>,<a2>,…<aX>) Example: <HEAD>*@EDITS OK,TRANS:SRV1(FLEX,0,1,193.193.165.165,20966)	
Exchange channel	GPRS, USB, Bluetooth, SMS	
Designations	**Interpretation**	**Data Format**
<HEAD>	16 byte NTCB packet header with preamble	16*U8
*!EDITS	0x2A 0x21 0x45 0x44 0x49 0x54 0x53	char[7]
*@EDITS	0x2A 0x40 0x45 0x44 0x49 0x54 0x53	char[7]
<s>	Parameter separator - space.	char
<p>	name of the page on which the tag is located. Must end with **":"**.	char[]
<t>	name of the tag you want to edit. Must end with **"("**.	char[]
<aX>	values of parameters written to the device. The last tag parameter must end with **")"**.	char[]
<result>	command execution result: - "OK" - command is executed; - "FAIL" - error in the command.	char[]

Reading configuration

Command	<HEAD>*!READ<s><p>:<t> Example: <HEAD>*!READ TRANS:AP1	
Response	<HEAD>*@READ<s><result>,<p>:<t>(<a1>,<a2>,…<aX>) Example: <HEAD>*@READ OK,TRANS:AP1(internet.mts.ru,mts,mts)	
Exchange channel	GPRS, USB, Bluetooth, SMS	
Designations	**Interpretation**	**Data Format**
<HEAD>	16 byte NTCB packet header with preamble	16*U8
*!EDITS	0x2A 0x21 0x52 0x45 0x41 0x44	char[6]
*@EDITS	0x2A 0x40 0x52 0x45 0x41 0x44	char[6]
<s>	Parameter separator - space.	char
<p>	name of the page on which the tag is located. Must end with **":"**.	char[]
<t>	name of the tag you want to edit. Must end with **"("**.	char[]
<aX>	values of parameters written to the device. The last tag parameter must end with **")"**.	char[]
<result>	command execution result: - "OK" - command executed; - "FAIL" - error in the command.	char[]

FLEX bit mask edit command

Sending this command causes device to reboot

Command	<HEAD>*!SETFM<s><n>:<m>[,<n>:<m>] Example: <HEAD>*!SETFM 7:1111,20:11	
Response	<HEAD>*@SETFM<s><n>:<m>	
Exchange channel	GPRS, USB, Bluetooth, SMS	
Designations	**Interpretation**	**Data Format**
<HEAD>	16 byte NTCB packet header with preamble	16*U8
*!SETFM	0x2A 0x21 0x53 0x45 0x54 0x46 0x4D	char[7]
*@SETFM	0x2A 0x40 0x53 0x45 0x54 0x46 0x4D	char[7]
<s>	Parameter separator - space (0x20).	char
<n>	Index from which bits will be set, starts from 1. Parameter number corresponds to the number from the table "Structure of FLEX telemetry records".	char

3. Encryption AES128

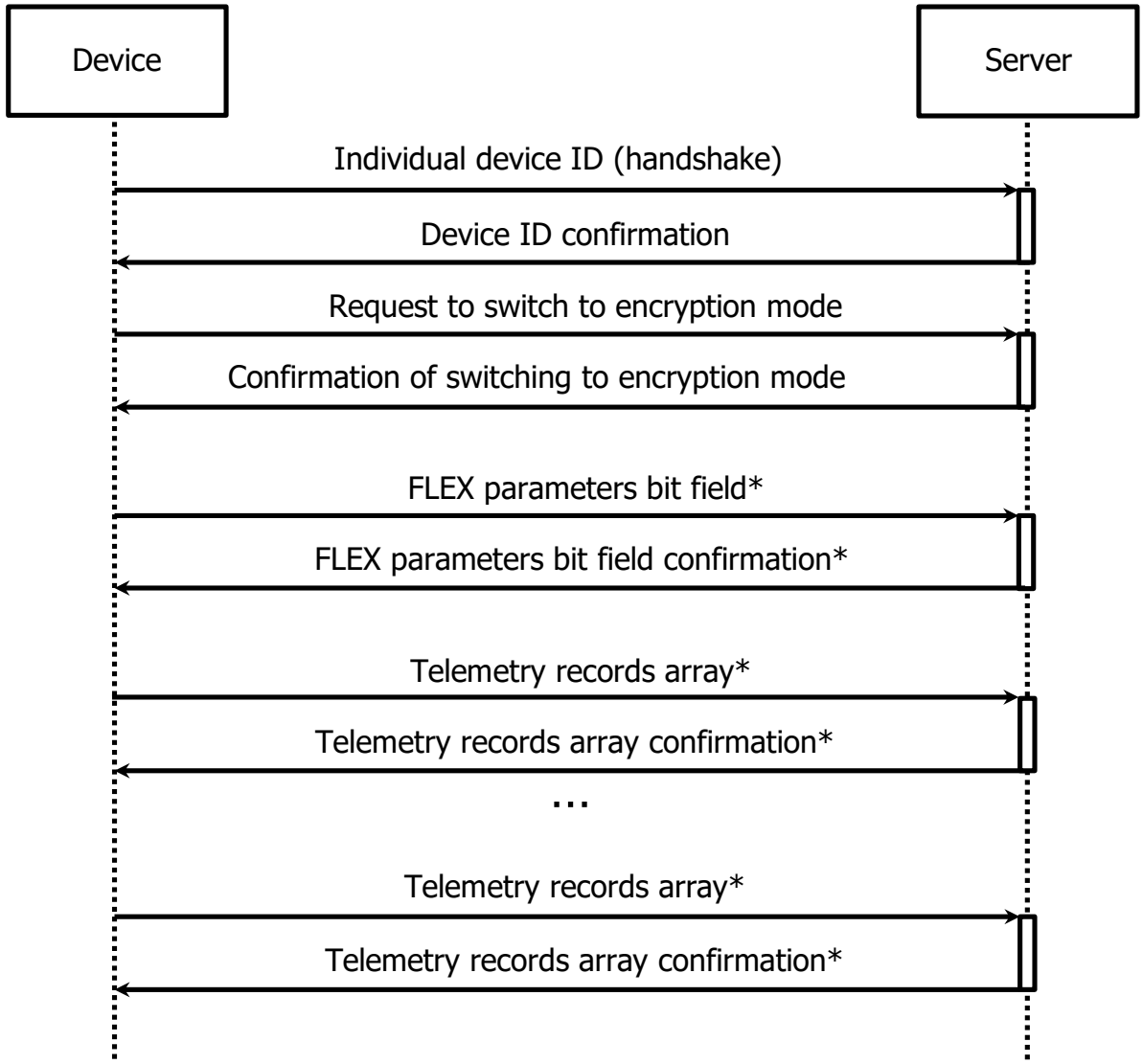
Advanced Encryption Standard (AES) algorithm in Cipher Block Chaining (CBC) mode with a 128-bit key is used for data encryption in device. PKCS7 algorithm described in RFC 5652 is used as an algorithm for padding data to a multiple of the size of the encryption block: padding is done in bytes, each of which is equal to the number of bytes that need to be added to the open data.

Basic definitions

Definition	Meaning
Device key	Cryptic key used to encrypt data transmitted from both the device and the server. This key is generated using the server key from the IMEI of the device, terminated with '*' and encoded in ASCII.
Initialization vector	Vector used to initialize the CBC algorithm. This vector is generated by the device itself when connecting to the server and every 12 hours if the device remains in touch. Initialization vector is generated based on 4 random numbers obtained from a random number generator (RNG).

Procedure for connecting device to the server when encryption mode is enabled

After opening connection, device sends an unencrypted handshake packet. In response, server either sends a corresponding unencrypted response if the device is registered in the database, or breaks the connection.
Upon successful completion of the handshake procedure, device sends to the server an unencrypted command to switch to encryption mode, which indicates the required encryption mode and initialization vector of the data block chaining algorithm.
When receiving a command to switch to encryption mode, server must confirm that the specified encryption mode is supported by sending an unencrypted response. If the server does not confirm encryption, you should disconnect the connection to the device.
If the server confirms support for the required encryption, the device begins data transmission only in encrypted form in a special container packet. At the same time, connection procedure itself remains the same.



* transmitted in encrypted form in a special container packet

Command to switch to encryption mode

Command	<HEAD>*>CODE<mode><vector>	
Response	<HEAD>*<CODE<mode>	
Exchange channel	GPRS	
Designations	**Interpretation**	**Data Format**
<HEAD>	16 byte NTCB packet header with preamble	16*U8
*>CODE	0x2A 0x3E 0x43 0x4F 0x44 0x45	char[5]
*<CODE	0x2A 0x3C 0x43 0x4F 0x44 0x45	char[5]
<mode>	In command: Encryption mode: 0x00 - no encryption; 0x01 – AES128 CBC. In response: Value from the command if the server supports the required encryption, 0 if the server does not support the required encryption.	U8
<vector>	Encryption initialization vector. Generated by the device.	N*U8 N - length of the vector, determined by the transport layer header.

Container packet for encrypted data transferring

Container	#<count><blocks><crc16>	
Exchange channel	GPRS	
Designations	**Interpretation**	**Data Format**
#	0x23	U8
<count>	Number of units to be transferred minus 1.	U8
<blocks>	Blocks of encrypted data.	<blocks_count> × N × U8 N - Encryption block size. For AES128, it is equal to 16.
<crc16>	Checksum. Polynomial 0x1021, initial value 0. See Annex B. Checksum calculation algorithm CRC8	U16 (little-endian)

4. NTCT text protocol

NTCT text protocol messages are used to send information via SMS. Format of a standard SMS message depends on the structure of the telemetry record used in the device.

4.1. Formats of SMS messages coming from device

F2, F5, F5.1F5.2, F6, FLEX records use message format M:111

String #	String content	Interpretation
1	M:111	Message type
2	_ _ _ _ _ _ _ _ _ _	Type of event that occurred, see Event code table file.
3	HH.MM.SS	Event time UTC
4	DD/MM/YY	Event date UTC
5	G:X	Operating mode where X: 0 "surveillance"; 1 "security"; 2 "service mode."
6	I:XXXXXXXX	State of inputs at the moment of fixing the event in the non-volatile memory. From I1 to I8 from left to right. X - not used; Y - triggered; N – not triggered; L - locked by command.
7	O:XXXX	Status of outputs at the moment fixing the event in the non-volatile memory. From O1 to O4 from left to right. X - not used; Y – on; N - off.
8	AK:XX.X XX.X	Voltage at the inputs of main and backup power supply in volts (with tenths)
9	T:ZXX	Temperature in degrees Celsius Z - sign "+" or "-," XX - value
10	AN:XX.X XX.X	Field is left for compatibility with the E-1111 device. It will contain only zero values.
11	HH.MM.SS	Time of obtaining the last valid coordinates according to UTC
12	DD/MM/YY	Date of obtaining the last valid coordinates according to UTC
13	NXXX XX.XXXX	Latitude in degrees, minutes and fractions of minutes. N - northern latitude; S - southern latitude.
14	EXXX XX.XXXX	Longitude in degrees, minutes and fractions of minutes. E - east longitude; W - west longitude.
15	XXX	Speed in km/h
16	XXX	Course in degrees (0 to 359)
17	hhhhhhhh	Record index in non-volatile memory (in hexadecimal)

Standard SMS format M:100

String #	String content	Note
1	M:100	Message type
2	<vendor>	Manufacturing company
3	X-XXXX	Device model string (6 characters)
4	Software version:	Firmware version
5	XX.XX.XX	Version number
6	XX.XX.XX	Version date
7	XX	Localization (RU - Russian version, DE - German version)

Standard SMS format M:101

String #	String content	Note
1	M:101	Message type
2	<text>	Message text generated by the cellular operator response to the USSD request

Standard SMS format M:104

String #	String content	Note
1	M:104	Message type
2	<MODE>	Current operating mode of the device: NOGUARD - surveillence mode; GUARD - security mode; GUARD2 - additional security mode 1; GUARD3 - additional security mode 2;
3	<NAME>	Character-numeric name: IX - inputs, where X = 1... OX - outputs, where X = 1... UG - main power supply; UR - backup power supply; T - temperature; A1 - voltage at analog input of I7/A1; A2 - voltage at analog input of I8/A2.
4	<STATE>	Condition: LOCKED - locked by command (only for I1... I8); OFF - for I1..I8 and O1..O4 means inactive state; ACTIVE - for I1..I8 and O1..O4 means active state; SHORT - for I7..I8 means short circuit on the line; <numerical value> - for voltage and temperature; NOT DEFINED - if parameter is unknown or incorrect.

Standard SMS format M:105

String #	String content	Note
1	M:105	Message type
2	<alert type>	Type of phone list; PVD - list of subscribers for voice alerts; PST - list of subscribers for alerts by standard SMS; PU - list of subscribers for alerts by user SMS.
3	<phone alias 1 >:<flag>	First subscriber name and alert status
4	<phone alias 2>: <flag>	Second subscriber name and alert status
5	<phone alias 3>: <flag>	Third subscriber name and alert status
6	<phone alias 4>: <flag>	Fourth subscriber name and alert status
7	<phone alias 5>: <flag>	Fifth subscriber name and alert status

Standard SMS format M:106

String #	String content	Note
1	M:106	Message type
2	Reset device	Response to RESET command Device will be rebooted

Standard SMS format M:107

String #	String content	Note
1	M:107	Message type
2	Firmware OK	Confirmation that the firmware was successfully downloaded through the RFU service and it will be written to device.
	Firmware <string> error <error code>	Error was detected while downloading the firmware. <string> - string with the firmware version; <error code> - code of the detected error.
	<ip>:<port> not responding	Specified server and port are not responding to communication requests.
	Start connect to <ip>: <port>	Command to update the firmware was accepted, attempt is made to establish connection with the specified server. <ip> - RFU IP address specified in the command. <port> - RFU IP port specified in the command.

Standard SMS format M:112

String #	String content	Interpretation
1	M:112	Message type
2	HH.MM.SS	Event time UTC
3	DD/MM/YY	Event date UTC
4	CX,MCC:Y,MNC:Y, CID:Y, LAC:Y,R:Z	X - station sequence number Y - parameters identifying the station Z - dBm signal attenuation to the station
5	CX,MCC:Y,MNC:Y, CID:Y, LAC:Y,R:Z	X - station sequence number Y - parameters identifying the station Z - dBm signal attenuation to the station
6	CX,MCC:Y,MNC:Y, CID:Y, LAC:Y,R:Z	X - station sequence number Y - parameters identifying the station Z - dBm signal attenuation to the station

Standard SMS format M:114

String #	String content	Note
1	M:114	Message type
2	X-XXXX	Device model string (6 characters)
3	XX.XX.XX	Version number
4	IMEI	International mobile equipment identifier
5	HH.MM.SS	Event time UTC
6	DD/MM/YY	Event fate UTC
7	…	Googlemap hyperlink

Standard SMS format M:115

String #	String content	Note
1	M:115	Message type
2	XXX: hhmmssDDMMYY	XXX - GSM error code hmmssDDMMYY - time of error code generation, where: hh - hour mm - minute ss - second DD - day MM - month YY - year
3	XXX: hhmmssDDMMYY	
4	XXX: hhmmssDDMMYY	
5	XXX: hhmmssDDMMYY	
6	XXX: hhmmssDDMMYY	
7	XXX: hhmmssDDMMYY	
8	XXX: hhmmssDDMMYY	
9	XXX: hhmmssDDMMYY	
10	XXX: hhmmssDDMMYY	

SMS message format according to GOST R 56361-2015

String #	String content	Interpretation
1	IMEI	International mobile equipment identifier
2	X	Location coordinates correspond to the system: 0 - PZ-90; 1 — WGS-84.
3	NXXX XX.XXXX	Latitude in degrees, minutes and fractions of minutes. N - northern latitude; S - southern latitude.
4	EXXX XX.XXXX	Longitude in degrees, minutes and fractions of minutes. E - east longitude; W - west longitude.
5	G:XXXX	Altitude relative to sea level in decimeters.
6	XXX	Speed in km/h.
7	XXX	Course in degrees (0 to 359).
8	HH.MM.SS	Time of obtaining the last valid coordinates according to UTC
9	DD/MM/YY	Date of obtaining the last valid coordinates according to UTC
10	I:XXXXXXXX	State of the inputs at the time the event was committed in the black box. From I1 to I8 from left to right. X - not used; Y - worked; N - did not work; L - blocked by command.

4.2. SMS requests and commands

Via the SMS communication channel it is possible to request information from non-volatile memory of the system, send commands and standard requests.

When sending SMS commands and requests, following rules are applied:
1. All commands are case-sensitive specified in the command description.
2. All Commands are typed in Latin.
3. If password for SMS control is set in the device configuration, then when sending a command, a construction of the form is used:

<password><sp><command> Example *?V with password "123": ***123 *?V***

If password is not set in the configuration, then it is sent like this:

<command> Example *?V without password ****?V***

Device sends a response to the command only if password check is passed, command format is correct and response is not prohibited by the device configuration.

4.2.1. System queries and commands

Query model and firmware version

Command	*?V
Response	*#V:<n>:<v1>.<v2>.<v3>:<d>.<m>.<y>:<loc> Example: *#V:S-2651:01.00.72:13.05.20:RU
Designations	**Interpretation**
<n>	Device model string (6 characters).
<v1>.<v2>.<v3>	Firmware version indexes of 2 characters.
<d>.<m>.<y>	Respectively, day, month and year of this firmware version are 2 characters.
<loc>	Firmware language version 2 characters (RU, DE, EN).

Query unique device ID

Command	*?S
Response	*#S:<s> Example: *#S:123456789012345
Designations	**Interpretation**
<s>	Identifier string.

Query personal SIM account balance

Command	B
Response	M:101 (see 4.1. Formats of SMS messages coming from device)
Designations	**Interpretation**
<n>	Device model string (6 characters).
<v1>.<v2>.<v3>	Firmware version indexes of 2 characters.
<d>.<m>.<y>	Respectively, day, month and year of this firmware version are 2 characters.
<loc>	Firmware language version 2 characters (RU, DE, EN).

Device restart command *(old format)*

Command	RESET
Response	M:106 (see 4.1 Formats of SMS messages coming from device)

Device restart command *(new format)*

Command	*!DEV_RESET
Response	*@DEV_RESET

Arbitrary USSD request

Command	*?USSD<s><code>
Response	*#USSD<s><string>
Designations	**Interpretation**
<s>	Parameter separator - space (0x20).
<code>	USSD code string of the request.
<string>	Response string of the cellular operator.

Command of arbitrary sound indication by buzzer

Command	*!BEEP<s><count>,<group>,<interval>,<freq>
Response	*@BEEP<s><result>
Designations	**Interpretation**
<s>	Parameter separator - space (0x20).
<result>	Command processing result (ASCII): "OK" - commands executed; "FAIL" - command execution error;
<count>	Number of pulses in each pulse group. If >=32, indication is continuous
<group>	Number of groups in audible indication.
<interval>	Duration of each group operating in 1/128 fractions of a second.
<freq>	Frequency of audible indication of each pulse (Hz).

Command to change SIM card

Command	*!CHNGSIM
Response	*@CHNGSIM<s><x>-><y>
Designations	**Interpretation**
<s>	Parameter separator - space (0x20).
->	Separating characters - 0x2D 0x3E
<x>	Current working SIM card: "1" - SIM 1 (External), "2" - SIM 2 (Internal).
<y>	SIM card to switch to: "1" - SIM 1 (External), "2" - SIM 2 (Internal).

Query device status

Command	*?ES
Response	M:114 (see 4.1 Formats of SMS messages coming from device)

Query unique SIM serial number

Command	*?ICCID
Response	*#ICCID<s><id>
Designations	**Interpretation**
<s>	Parameter separator - space (0x20).
<id>	Unique SIM serial number

Query information about firmware version of the GPS receiver

Command	*?VGPS
Response	*#VGPS<s><n>,<v1>.<v2>.<v3>,<d>.<m>.<y>,<gps_ver>
Designations	**Interpretation**
<s>	Parameter separator - space (0x20).
<n>	Device model string (6 characters).
<v1>.<v2>.<v3>	Device firmware version indexes of 2 characters.
<d>.<m>.<y>	Respectively, day, month and year of this version of the device firmware are 2 characters each.
<gps_ver>	String with model and version of navigation receiver

Command for microphone listening

Command	*!O<s><phnumber>
Response	*@O<s><phnumber>
Designations	**Interpretation**
<s>	Parameter separator - space (0x20).
<phnumber>	<phnumber> - number to be called. Phone number (starting with "+").

Command for time adjustment on device

Command	*!SETTIME<s><offset> Example: *!SETTIME +60
Response	*@SETTIME<s><ans>
Designations	**Interpretation**
<s>	Parameter separator - space (0x20).
<offset>	Offset in seconds with a character in text format. For example: + 60 or -30
<ans>	Response: "OK" - "time synchronized"; "FAIL" - "synchronization error."

Command for diagnostic information query

Command	*?DATA<s><id1>,<id2>, ... ,<idN> Example: *?DATA 21,22
Response	*#DATA<s><id1>:<data1>,<id2>:<data2>,...,<idN>:<dataN> Example: *#DATA 21:0026,22:0000
Designations	**Interpretation**
<s>	Parameter separator - space (0x20).
<idX>	Parameter ID in text decimal form. 1 .. 255 - parameters correspond to table FLEX 3.0 (Annex A.1. Structure of FLEX telemetry records)
<dataX>	Parameter value as a HEX string with BigEndian byte order.

4.2.2. Outputs

Command to change output state

Command	*!SETOUT<s><num><new_state>[,<num><new_state>] Example: *!SETOUT 1Y,2N
Response	*@SETOUT<s><result>,<num><cur_state>[,<num><cur_state>] Example: *@SETOUT OK,1Y,2N или *@SETOUT FAIL,1N,2Y
Designations	**Interpretation**
<s>	Parameter separator - space (0x20).
<result>	Command processing result (ASCII): "OK" - commands executed; "FAIL" - command execution error;
<num>	Number of the output to be changed (ASCII). Numbering starts with 1.
<new_state>	Output state to be set (ASCII): "Y" - enable; "N" - disablef.
<cur_state>	Output status after command execution (ASCII): "Y" - on; "N" - off.

Command to enable/disable output

Command	<num><state> Example: 1Y
Response	M:111 (see 4.1 Formats of SMS messages coming from device)
Designations	**Interpretation**
<num>	Output number starting from 1
<state>	Output state to be set (ASCII): "Y" - enable; "N" - disable.

4.2.3. Inputs

Command to lock input

Command	LOCK<s>I<num> Example: LOCK I1
Response	M:105 (see 4.1 Formats of SMS messages coming from device)
Designations	**Interpretation**
<s>	Parameter separator - space (0x20).
<num>	Input number starting from 1

Command to unlock input

Command	UNLOCK<s>I<num> Example: UNLOCK I1
Response	M:105 (see 4.1 Formats of SMS messages coming from device)
Designations	**Interpretation**
<s>	Parameter separator - space (0x20).
<num>	Input number starting from 1

4.2.4. Telemetry information

Query current state of the device

Command	*?A
Response	M:111 (see 4.1 Formats of SMS messages coming from device)

Query current status of <X> sensor

Command	A<x> Example: AUG or AI1
Response	M:111 (see 4.1 Formats of SMS messages coming from device)
Designations	**Interpretation**
<x>	<x> - alphanumeric value of the sensor in the system: I1-I8 - inputs; O1-O4 - outputs; UG - power supply voltage; UR – back-up battery voltage; T1-T4 - temperature; IGN - "Ignition" sensor state (by input or voltage); ACC - disconnection of back-up battery; PWR - disconnection of external power supply.

Query current location as a hyperlink

Command	*?POS
Response	M:114 (see 4.1 Formats of SMS messages coming from device)

Query location-based service (LBS) information

Command	*?LBS
Response	M:112 (see 4.1 Formats of SMS messages coming from device)

Command to confirm synchronization of internal memory with the server

Command	*!SYNC<s><index> Example: *! SYNC 1 or *! SYNC: 1
Response	M:111 (see 4.1 Formats of SMS messages coming from device)
Designations	**Interpretation**
<s>	Parameter separator - space (0x20) or colon (0x3A).
<index>	Server index of the device specified in the settings (starting from 1).

Command to resend telemetry from non-volatile memory

Command	*!REP_FL<s><srvindex>,<leftdate>[/<lefttime>],<rightdate>[/<righttime>] Example: *!REP_FL 1,09.04.18/13:00:59,10.04.18/03:00:00 *!REP_FL 1,09.04.18,10.04.18
Response	*@REP_FL<s><result>
Designations	**Interpretation**
<s>	Parameter separator - space (0x20).
/	Date/Time field separator (0x2F)
<result>	Command processing result (ASCII): "OK" - command executed; "FAIL" - command execution failed.
<srvindex>	Server index to repeat in text format: "0" - to all servers; "1.." "3" – server index.
<leftdate>	Date of the left border of the interval of the requested telemetry in text format (in UTC): "DD.MM.YY."
<lefttime>	***Optional parameter.*** Time of the left border of the interval of the requested telemetry in text format (in UTC): "HH: MM: SS Absence of a parameter is equal to the value "00:00:00."
<rightdate>	Date of the right border of the interval of the requested telemetry in text format (in UTC): "DD.MM.YY."
<righttime>	***Optional parameter.*** Time of the right border of the requested telemetry interval in text format (in UTC): "HH: MM: SS Absence of a parameter is equal to the value "23:59:59."

Command to resend telemetry from SD card

Command	*!REP_SD<s><srvindex>,<leftdate>[/<lefttime>],<rightdate>[/<righttime>] Example: *! REP _ SD 1.09.04.18/13: 00: 59.10.04.18/03: *00:00 (time will not be taken into account* *! REP _ SD 1.09.04.18. 10.04.18 (only *09.04.18 will be unloaded*
Response	*@REP_SD<s><result>
Designations	**Interpretation**
<s>	Parameter separator - space (0x20).
/	Date/Time field separator (0x2F)
<result>	Command processing result (ASCII): "OK" - commands executed; "FAIL" - command execution failed.
<srvindex>	Server index to repeat in text format: "0" - to all servers; "1.." "3" – server index.
<leftdate>	Date of the left border of the interval of the requested telemetry in text format (in UTC): "DD.MM.YY."
<lefttime>	***Device does not parse the parameter.*** Always equates to "00:00:00." Time of the left border of the interval of the requested telemetry in text format (in UTC): "HH: MM: SS
<rightdate>	Date of the right border of the interval of the requested telemetry in text format (in UTC): "DD.MM.YY."
<righttime>	***Device does not parse the parameter.*** Always equates to "00:00:00." Time of the left border of the interval of the requested telemetry in text format (in UTC): "HH: MM: SS

4.2.5. Device operating modes

Command to enable security mode

Command	*!M<s><x> Example: *!M G
Response	*@M<s><x>,<e>,<i> Example: *@M O,5,IN2/VOLT
Designations	**Interpretation**
<x>	Operating mode of the device to be switched to: "G" - security; "O" - surveillance. Response <x> is the current mode of operation
<e>	Reason for not switching to security: "1" - security mode in the device configuration is disabled; "2" - mode change switching timeout has not expired; "3" - mode is on: do not switch to security mode when ignition is on; "4" - device is already in this mode; "5" - mode is on: do not switch to security mode if one of the security sensors is triggered.
<i>	Aliases of the triggered sensors via separator - **/** . ***For S-25xx, S-23xx devices:*** "IN1," "IN2," "IN3," "IN4," "AIN1," "AIN2" – inputs; "VOLT" - external supply voltage sensor; "ENG" - engine operation sensor. ***For S-26xx, S-24xx devices:*** "IN1," "IN2," "IN3," "IN4," "IN5," "IN6" - inputs; "VOLT" - external supply voltage sensor; "ENG" - engine operation sensor.

Query current security mode

Command	*?M
Response	*#M<s><x>
Designations	**Interpretation**
<s>	Parameter separator - space (0x20).
<x>	Device operating mode: "G" - security; "O" - surveillance.

Command to switch security mode *(old format)*

Command	G<mode> Example: G1 or GY
Response	M:111 (see 4.1 Formats of SMS messages coming from device)
Designations	**Interpretation**
<mode>	Operating mode: "Y" or "1" - "Security" mode; "N" or "0" - "Surveillence" mode.

4.2.6. RCS, RFU, DRC Services

To connect device to RCS, RFU, DRC servers, it is necessary to send it an appropriate command via SMS.

Command to connect to the configurator via RCS

Command	*!CNCT_RCS<s><ip>,<port>,<commID>,<apn>,<login>,<password> Example: *!CNCT_RCS 89.208.152.55,8100,43644176 *!CNCT_RCS 89.208.152.55,8100,43644176,internet.mts.ru,mts,mts
Response	M:107 (see 4.1 Formats of SMS messages coming from device)
Designations	**Interpretation**
<s>	Parameter separator - space (0x20).
<ip>	String containing the IP address of the RCS server.
<port>	String containing the port on which the RCS service on the server is routed.
<commID>	Session ID to connect to the configurator.
<apn>	***Optional parameter.*** Access point name of the cellular operator. If the field in the cellular operator settings is empty, the field should be omitted.
<login>	***Optional parameter.*** Login of the cellular operator. If the field in the cellular operator settings is empty, the field should be omitted.
<password>	***Optional parameter.*** Password of the cellular operator. If the field in the cellular operator settings is empty, the field should be omitted

Command to update device firmware via RFU

Command	*!CNCT_RFU<s><ip>,<port>,<firmware>,<apn>,<login>,<password> Example: *!CNCT_RFU 89.208.152.55,9100,LAST *!CNCT_RFU 89.208.152.55,9100,01.00.72 *!CNCT_RFU 89.208.152.55,9100,LAST,internet.mts.ru,mts,mts
Response	M:107 (see 4.1 Formats of SMS messages coming from device)
Designations	**Interpretation**
<s>	Parameter separator - space (0x20).
<ip>	String containing IP address of the RCS server.
<port>	String containing the port on which the RCS service on the server is routed.
<firmware>	Requested firmware version. If the latest version is needed, the keyword "LAST" is used.
<apn>	***Optional parameter.*** Access point name of the cellular operator. If the field in the cellular operator settings is empty, the field should be omitted.
<login>	***Optional parameter.*** Login of the cellular operator. If the field in the cellular operator settings is empty, the field should be omitted.
<password>	***Optional parameter.*** Password of the cellular operator. If the field in the cellular operator settings is empty, the field should be omitted

Old command formats:
(temporarily supported for backward compatibility)

№	Command Text	Command Description	Response Message
1	UPFRM <IP> <PORT> <FIRMWARE> <APN> <LOGIN> <PASSWORD>	RFU service connection command: ***Separator between parameters is space.*** <IP> - IP address of the RFU server; For example: 89.208.152.55 <PORT> - IP port of the RFU server; For example: 9100 <FIRMWARE> - firmware version number (LAST) for the newest; <APN> - cellular operator apn; <LOGIN> - cellular operator login; <PASSWORD> cellular operator password.	M:107
2	NTC_CONNECT <IP> <PORT> <COMID> <APN> <LOGIN> <PASSWORD>	RCS service connection command: ***Separator between parameters is space.*** <IP> - IP address of the RCS server; For example: 89.208.152.55 <PORT> - IP port of the RCS server; For example: 8100 <COMID> - RCS session ID; For example: 43644176 <APN> - cellular operator apn; <LOGIN> - cellular operator login; <PASSWORD> cellular operator password.	M:107

Command to connect to DRC

Command	*!CNCT_DRC[<s><time>,<ip>, <port>] Example: *!CNCT_DRC *!CNCT_DRC 300 *!CNCT_DRC 300,89.208.152.54,10000
Response	*@CNCT_DRC<s><ack>
Designations	**Interpretation**
<time>	*(optional)* Ttime, in seconds, for which the device is connected.
<ip>	*(optional)* IP or DNS of the DRC service. If not specified, device uses the server IP address hard-coded in the firmware.
<port>	*(optional)* PORT of the DRC service. If not specified, device uses the server port hard-coded in the firmware.
<ack>	Confirmation of command reception: OK - command is accepted for execution.

4.2.7. Working with identifiers

Command to edit the list of registered Touch Memory keys

ATTENTION!
List with theTouch Memory key parameters must be loaded into the device in advance. Only one key can be edited at a time.

Command	*!SETTM<s><num>,<addr>,<mode> Example: *!SETTM 1,0002de031122,NOACT
Response	*@SETTM<s><res>[,<code>]
Designations	**Interpretation**
<s>	Parameter separator - space (0x20).
<num>	Key number in the configuration is 1-510.
<addr>	Key address (12 characters).
<mode>	Key operating mode: "NOACT" - no action; "GUARD" - changes the security mode; "IMMOB" - immobilizer.
<res>	Command execution result: "OK" – command executed successfully; "FAIL" - command failed, error.
<code>	*(optional)* Additional error code: "1" - module is busy; "2" - incorrect TM key number; "3" - no parameters; "4" - incorrect operating mode.

4.2.8. Tachograph

Command to generate a DDD file and send it to e-mail

Command	TACH<s>EMAIL<s><card_num>
Response	TACH<s>EMAIL:<s><ans>
Designations	**Interpretation**
<s>	Parameter separator - space (0x20).
<card_num>	Card number to unload: "1" - card from the first slot; "2" - card from the second slot.
<ans>	Command execution results: "OK" - command executed successfully. "BUSY" - command cannot be executed at the moment. "NO TACHOGRAPH" - device is not configured to work with tachograph. "NOT CONNECTED" – no connection to the tachograph. "AUTH ERROR" - failed to log in to the tachograph. "DISK ERROR" - error saving/reading data from non-volatile memory. "DDD WRONG PARAMS" - DDD generation: specified incorrect parameters (date, card number) in the request. "DDD WRONG STATE" - DDD generation: tachograph is not allowed to generate a file in the current operating mode. "DDD NOT SUPPORTED" - DDD generation: unsupported file type. "DDD NO DATA" - DDD generation: no data to generate file. "DDD LOAD ERROR" – DDD generation: error getting file from tachograph. "DDD BUILD ABORTED" - DDD generation: generation interrupted (when updating device firmware). "DDD BROKEN" - DDD generation: file has incorrect structure (file name could not be generated). "EMAIL SEND ERROR" - failed to send file to e-mail.

Query status of tachograph cards

Command	TACH<s>INFO
Response	TACH<s>INFO:<s><err> TACH<s>INFO:<s><ans> Example: TACH INFO: ATOL 1 – RUD0000000002100 2 – NONE
Designations	**Interpretation**
<s>	Parameter separator - space (0x20).
<card_num>	Card number to unload: "1" - card from the first slot; "2" - card from the second slot.
<err>	Error information: "NO TACHOGRAPH" - device is not configured to work with tachograph "NOT CONNECTED" - no connection to the tachograph
<ans>	If successful, format of the <ans> field is as follows <serial> 1 - <card_num> 2 - <card_num> , where <serial> - tachograph model and serial number <card_num> - card number, if any, or "NONE" if none is present

4.2.9. Driver display

Send message without confirmation

Command	DV<msg>
Response	DV:<s><ans>
Designations	**Interpretation**
<s>	Parameter separator - space (0x20).
<msg>	Text message to the driver up to 229 characters inclusive (encoded as CP1251).
<ans>	Command execution result: "OK" - message is accepted. "BUSY" - message was not accepted, because the previous message was not sent to the display. "NO DV" - device is not configured to work with the display. "INVALID MSG" - message length = 0. "NOT CONNECTED" - no connection to the display.

Send message with confirmation request

Command	DV!<msg>
Response	DV:<s><ans>
Designations	**Interpretation**
<s>	Parameter separator - space (0x20).
!	Sign of waiting for confirmation
<msg>	Text message to the driver up to 229 characters inclusive (encoded as CP1251).
<ans>	Command execution result: "OK" - message is accepted. "BUSY" - the message was not accepted, because the previous message was not sent to the display. "NO DV" - device is not configured to work with the display. "INVALID MSG" - message length = 0. "NOT CONNECTED" - no connection to the display.

Read receipt for confirmation messages

Command	is not required, receipt is generated by the device itself
Response	DV:<date>
Designations	**Interpretation**
<s>	Parameter separator - space (0x20).
<date>	Time and date when device received a message for the driver in HH:MM:SS DD/MM/YYYY format

4.2.10. Autoinformer

Route change command *(old format)*

Command	AINF<s>ROUTE<s><n><m>
Response	AINF<s>ROUTE:<s><ans>
Designations	**Interpretation**
<s>	Field separator - space (0x20)
<n>	Route number
<m>	Route letter (CP1251 encoded)
<ans>	Command execution result: "OK" - command executed successfully. "BUSY" - command cannot be executed because the execution of the previous command has not completed. "INVALID ROUTE" - route is formed incorrectly. "NO ROUTE" - route not found. "NO ZONES" – failed to load the list of geofences.

Audio file playback command *(old format)*

Command	AINF<s>PLAY<s><n>
Response	AINF<s> PLAY:<s><ans>
Designations	**Interpretation**
<s>	Field separator - space (0x20)
<n>	Audio file numeric ID
<ans>	Command execution result: "OK" - command executed successfully. "BUSY" - command cannot be executed because the execution of the previous command has not completed. INVALID SOUND - failed to play audio file. "NO SOUND" - audio file not found.

Speed mode change command

Command	AINF<s>SM<s><n>
Response	AINF<s>SM:<s><ans>
Designations	**Interpretation**
<s>	Field separator - space (0x20)
<n>	Speed mode numeric identifier
<ans>	Command execution result: "OK" - command executed successfully. "BUSY" - command cannot be executed because the execution of the previous command has not completed. "NO SM" - speed mode is not found in the mode list.

Route change command

Command	*!AINF<s>ROUTE,<route>
Response	*@AINF<s>ROUTE,<result>
Designations	**Interpretation**
<s>	Field separator - space (0x20)
<route>	String containing the number and letter of the route (encoded as CP1251)
<result>	Command yxecution κesult "OK" - command executed successfully. "BUSY" - command cannot be executed because the previous command has not completed. "INVALID ROUTE" - route is formed incorrectly. "NO ROUTE" - route not found. "NO ZONES" - failed to load stop list.

Audio file playback command

Command	*!AINF<s>PLAY,<sound_id>
Response	*@AINF<s>PLAY,<result>
Designations	**Interpretation**
<s>	Field separator - space (0x20)
<sound_id>	Sound file ID (1 to 65535). If 0, stops playing the current audio file.
<result>	"OK" - command executed successfully. "BUSY" - command cannot be executed because the previous command has not completed. INVALID SOUND - failed to play audio file. "NO SOUND" - audio file not found.

Command to change autoinformer operating mode

Command	*!AINF<s>MODE,<mode>
Response	*@AINF<s>MODE,<result>
Designations	**Interpretation**
<s>	Field separator - space (0x20)
<mode>	Autoinformer operating mode: "0" - automatic mode; "1" - manual mode.
<result>	Command execution result "OK" - command executed successfully. "BUSY" - command cannot be executed because the previous command has not completed. "INVALID ROUTE" - route is formed incorrectly.

Command to change current stop

Command	*!AINF<s>STOP,<stop>
Response	*@AINF<s>STOP,<result>
Designations	**Interpretation**
<s>	Field separator - space (0x20)
<stop>	Stop ID (1 to 65534). If 0 - exit from the current stop
<result>	Command execution result "OK" - command executed successfully. "BUSY" - command cannot be executed because the the previous command has not completed. "INVALID ROUTE" - route is formed incorrectly. "INVALID MODE" - manual mode is not enabled. "NO STOP" - stop not found.

4.2.11. Digital camera

Command to take a photo or series of photo

Command	DCAM<s>SS[<s><n><s><m>] Example: DCAM SS DCAM SS 3 5
Response	DCAM<s>SS:<s><ans>
Designations	**Interpretation**
<s>	Field separator - space (0x20)
<n>	*(optional)* Number of photos. If the parameter is not specified, then it is taken as 1.
<m>	*(optional)* Interval between photos in seconds. If the parameter is not specified, then it is taken as 1.
<ans>	Command execution result: "OK" - command executed successfully "BUSY" - command cannot be executed at the moment "NO CAMERA" - device is not configured to work with the camera "NOT CONNECTED" - no communication with the camera "TURNED OFF" - automatic shooting is disabled "DISK ERROR" - SD card error "SS ERROR" - error retrieving camera photo

Command to take a photo and then send it to e-mail

Command	DCAM<s>EMAIL<s><date>
Response	DCAM<s>EMAIL:<s><ans>
Designations	**Interpretation**
<s>	Field separator - space (0x20)
<date>	Date when the photo is created, if it is not in the command, a new photo is created. Format: HH.MM.SS<space>DD/MM/YY (UTC). For example: "15.30.45 17/09/15"
<ans>	Command execution result: "OK" - command executed successfully "BUSY" - command cannot be executed at the moment "NO CAMERA" - device is not configured to work with the camera "NOT CONNECTED" - no communication with the camera "TURNED OFF" - automatic shooting is disabled "DISK ERROR" - SD card error "NO PHOTO" - photo not found "SS ERROR" - error retrieving camera photo "EMAIL SEND ERROR" - failed to send photo to e-mail <state><s><version><s>DIR<s><dir> - Response to get camera information <state> - automatic shooting mode "ON" - enabled; "OFF" - disabled; <version> - camera version: "VC0706 1.00 <dir> - directory in which the photos are saved: "PHOTOS"

Command to enable/disable automatic shooting

Command	DCAM<s><cmd> Example: DCAM ON DCAM OFF
Response	DCAM<s><cmd>:<s><ans>
Designations	**Interpretation**
<s>	Field separator - space (0x20)
<ans>	Command execution result: "OK" - command executed successfully "BUSY" - command cannot be executed at this time "NO CAMERA" - device is not configured to work with the camera "TURNED OFF" - automatic shooting is disabled

Command to get information about camera

Command	DCAM<s>INFO
Response	DCAM<s>INFO:<s><ans>
Designations	**Interpretation**
<s>	Field separator - space (0x20)
<ans>	Command execution result: <state><s><version><s>DIR<s><dir> Field Description: < state > - automatic shooting mode "ON" - enabled; "OFF" - disabled; <version> - camera version: "VC0706 1.00 <dir> - directory in which the photos are saved: "PHOTOS".

Responses to camera control commands

Note: <cmd> - passed command without parameters

Response	Description
<cmd>: OK	Command executed successfully
<cmd>: BUSY	Command cannot be executed at this time
<cmd>: NO CAMERA	Device is not configured to work with the camera
<cmd>: NOT CONNECTED	No communication with camera
<cmd>: TURNED OFF	Automatic shooting is disabled
<cmd>: DISK ERROR	SD card error
<cmd>: NO PHOTO	Photo not found
<cmd>: SS ERROR	Error getting photo from camera
<cmd>: EMAIL SEND ERROR	Failed to send photo to e-mail
DCAM INFO: <state> <version> DIR <dir>	Response to get camera information command: <state> - automatic shooting mode "ON" - enabled; "OFF" - disabled; <version> - camera version"VC0706 1.00 <dir> - directory in which the photos are saved "PHOTOS"
DCAM PHOTO: <date>	Reply to the nearest photo request regarding <date> <date> - date when photo was created, if not in the command, new photo is created. Format: HH.MM.SS<space>DD/MM/YY (in UTC). "15.30.45 17/09/15"

4.2.12. Working with built-in accelerometer

Query the current calibration information

Command	CALIB<s>ACL<s>?
Response	CALIB<s>ACL:<s><ans>
Designations	**Interpretation**
<s>	Field separator - space (0x20)
<ans>	explanatory message: - "Accelerometer calibration is not supported"; - "Accelerometer is not calibrated"; - "Accelerometer calibrated by displacement"; - "Accelerometer is calibrated along axes."

Command to start offline calibration (fully manual)

Command	CALIB<s>ACL<s>S
Response	CALIB<s>ACL:<s><ans>
Designations	**Interpretation**
<s>	Field separator - space (0x20)
<ans>	explanatory message: - "Accelerometer calibration is not supported"; - "Static calibration started"; - "Dynamic calibration started"; - "Calibration completed"; - "Error: failed to measure gravity"; - "Error: car was moving (vibrating strongly)"; - "Error: car did not move"; - "Error: not enough sharp acceleration/braking."

Command to start GNSS calibration (semi-automatic)

Command	CALIB<s>ACL<s>G
Response	CALIB<s>ACL:<s><ans>
Designations	**Interpretation**
<s>	Field separator - space (0x20)
<ans>	explanatory message: - "Accelerometer calibration is not supported"; - "GNSS static calibration started"; - "GNSS dynamic calibration started"; - "Calibration completed"; - "Error: failed to measure gravity"; - "Error: car was moving (vibrating strongly)."

Command to start calibration status check (fully manual)

Command	CALIB<s>ACL<s>C
Response	CALIB<s>ACL:<s><ans>
Designations	**Interpretation**
<s>	Field separator - space (0x20)
<ans>	explanatory message: - "Accelerometer calibration is not supported"; - "Static check started"; - "Dynamic check started"; - "Error: failed to measure gravity"; - "Error: car did not move"; - "Calibration is correct"; - "Calibration must be repeated.

Command to cancel the previously started calibration

Command	CALIB<s>ACL<s>R
Response	CALIB<s>ACL:<s><ans>
Designations	**Interpretation**
<s>	Field separator - space (0x20)
<ans>	explanatory message: - "Accelerometer calibration is not supported"; - "Calibration interrupted."

Command to reset current calibration

Command	CALIB<s>ACL<s>E
Response	CALIB<s>ACL:<s><ans>
Designations	**Interpretation**
<s>	Field separator - space (0x20)
<ans>	explanatory message: - "Calibration reset."

4.2.13. Communication between external interfaces

This command allows you to send an arbitrary data string to the device interface or internal module. After sending data, device "listens to" the interface for 500 ms and returns a response without formatting. If no response is received within 500 ms, the response field will be empty.

Command	*!UC<s><i>,<msg> Example: *! UC RS485,3101066C
Response	*@UC<s><i>,<ans>
Designations	**Interpretation**
<s>	Parameter separator - space (0x20).
<i>	Interface: – "RS485" - RS-485 device interface, – "RS232" - RS-232 device interface, – "GPS" - built-in navigation module.
<msg>	For RS232, RS485: Packet as HEX string (ASCII). For GPS: Packet in NMEA protocol without checksum (device adds checksum).
<ans>	If response was received - HEX String (ASCII); If response was not received - empty string.

4.3. SMS configuration

4.3.1. EDITS and READ command formats

For detailed description of commands and general rules for working with them, see "6. Working with device configuration".

Writing configuration

Sending a command causes the device to reboot

Command	*!EDITS<s><p>:<t>(<a1>,<a2>,…<aX>) Example: *!EDITS TRANS:SRV1(,,,193.193.165.165,20966)
Response	*@EDITS<s><result>,<p>:<t>(<a1>,<a2>,…<aX>) Example: *@EDITS OK,TRANS:SRV1(FLEX,0,1,193.193.165.165,20966)
Designations	**Interpretation**
<s>	parameter separator - space (0x20).
<p>	name of the page on which the tag is located. Must end with "**:**".
<t>	name of the tag you want to edit. Must end with "**(**".
<aX>	values of parameters written to the device. The last tag parameter must end with "**)**".
<result>	command execution result: - "OK" - command executed; - "FAIL" - error in the command.

Reading configuration

Command	*!READ<s><p>:<t> Example: *!READ TRANS:AP1
Response	*@READ<s><result>,<p>:<t>(<a1>,<a2>,…<aX>) Example: *@READ OK,TRANS:AP1(internet.mts.ru,mts,mts)
Designations	**Interpretation**
<s>	parameter separator - space (0x20).
<p>	name of the page on which the tag is located. Must end with "**:**".
<t>	name of the tag you want to edit. Must end with "**(**".
<aX>	values of parameters written to the device. The last tag parameter must end with "**)**".
<result>	command execution result: - "OK" - command executed; - "FAIL" - error in the command.

FLEX bit mask edit command

Sending a command causes the device to reboot

Command	<HEAD>*!SETFM<s><n>:<m>[,<n>:<m>] Example: <HEAD>*!SETFM 7:1111,20:11
Response	<HEAD>*@SETFM<s><n>:<m>
Designations	**Interpretation**
<s>	Parameter separator - space (0x20).
<n>	Index from which the bits will be set starts from 1. The parameter number corresponds to the number from the table "Structure of FLEX telemetry records".
<m>	Bit mask.

4.3.2. SET and GET command formats

ATTENTION!

This section lists old command formats that are temporarily supported for backward compatibility.

Commands

№	Command text	Command description	Response message
1	SET.<x>.<y> <IP> <port> <IDo> <IDc> <apn> <login> <password> <protocol>	Writes the main parameters necessary to work with the server to the device: <x> - server number from 1 to 3, if not, the settings are edited for the first server <y> - SIM card number from 1 to 2, if not available, the settings are edited for the first card <IP> - IP address or domain name of the server. For example: 90.156.232.36 <port> - port. For example: 4000 < IDo > - identification number of the object <IDc> - personal account number <apn> - name of the cellular operator access point. For example: internet.mts.ru <login> - login of the cellular operator. For example: mts <password> - password of the cellular operator. For example: mts <protocol> - communication protocol with the server. Accepted values: “F6”; “F5.2”; “FLEX”; "EGTS" - without authorization; "EGTS_A" - EGTS with authorization. Separator between parameters is space or row wrap.	Contains the current device settings: CUR. <x>. <y> - message type <IP> - IP address or domain name of the server <port> - server port <IDo> - identification number of the object <IDc> - personal account number <apn> - name of the access point of the cellular operator <login> - login of the cellular operator <password> - password of the cellular operator <protocol> - communication protocol with the server <IMEI> - unique number of the object Comes after the reboot. Where: <x> - server number from 1 to 3, if not, the settings are edited for the first server <y> - SIM card number from 1 to 2, if not available, the settings are edited for the first card
2	SET1.<x>.<y> <IP> <port> <IDo> <IDc>	Writes the main parameters necessary to work with the server to the device: <x> - server number from 1 to 3, if not, the settings are edited for the first server <y> - SIM card number from 1 to 2, if not available, the settings are edited for the first card <IP> - IP address or domain name of the server. For example: 90.156.232.36 <port> - port. For example: 4000 <IDo> - identification number of the object. <IDc> - personal account number Separator between parameters is space or row wrap.	Contains the current device settings: CUR1. <x>. <y> is the message type <IP> - IP address or domain name of the server <port> - server port <IDo> - identification number of the object <IDc> - personal account number <IMEI> - unique number of the object Comes after the reboot. Where: <x> - server number from 1 to 3, if not, the settings are edited for the first server <y> - SIM card number from 1 to 2, if not available, the settings are edited for the first card
3	SET2.<x>.<y> <apn> <login> <password> <protocol>	Writes to the device the parameters necessary for accessing the Internet (GPRS settings): <x> - server number from 1 to 3, if not, the settings are edited for the first server <y> - SIM card number from 1 to 2, if not available, the settings are edited for the first card <apn> - name of the cellular operator access point. For example: internet.mts.ru <login> - login of the cellular operator. For example: mts	Contains the current device settings: CUR2. <x>. <y> - message type <apn> - name of the access point of the cellular operator <login> - login of the cellular operator <password> - password of the cellular operator <protocol> - communication protocol with the server Comes after the reboot. Where:

		<password> - password of the cellular operator. For example: mts <protocol> - communication protocol with the server Accepted values: "F6"; "F5.2"; "FLEX"; "EGTS" - without authorization; "EGTS_A" - EGTS with authorization. Separator between parameters is space or row wrap.	<x> - server number from 1 to 3, if not, the settings are edited for the first server <y> - SIM card number from 1 to 2, if not available, the settings are edited for the first card
4	SETTM<num> <address> <mode> <nick_name>	Edits the TM key settings (the key page must already be in the device): <num> - number of the key in the configuration (2 characters always, 01-16, for example "01"); <address> - key address (12 characters <mode> - key operation mode (5 characters "NOACT" - no action; "GUARD" - changes the security mode; "IMMOB" - immobilizer. <nick_name> - key alias (10 characters, there is no extended key storage format Separator between parameters is space or row wrap.	If command successfully executed: SETTM OK. If there is error in the command, a standard text response for the SET command will be generated.

Queries

№	Text Commands	Command Description	Reply Message
1	GET.<x>.<y>	Gets the current device settings required to work with the server and generated in the template.	Contains the current device settings: CUR. <x>. <y> - message type <IP> - IP address or domain name of the server <port> - server port <IDo> - identification number of the object <IDc> - personal account number <apn> - name of the access point of the cellular operator <login> - login of the cellular operator <password> - password of the cellular operator <protocol> - communication protocol with the server <IMEI> - unique object number Where: <x> - server number from 1 to 3, if not, the settings are edited for the first server <y> - SIM card number from 1 to 2, if not available, the settings are edited for the first card
2	GET1.<x>.<y>	Gets the current device settings required to work with the server and generated in the template.	Contains the current device settings: CUR1. <x>. <y> - message type <IP> - IP address or domain name of the server <port> - server port <IDo> - identification number of the object <IDc> - personal account number <IMEI> - unique number of the object Where:

			<x> - server number from 1 to 3, if not, the settings are edited for the first server <y> - SIM card number from 1 to 2, if not available, the settings are edited for the first card
3	GET2.<x>.<y>	Gets the current device settings required to access the Internet and generated in the template.	CUR2. <x>. <y> - message type <apn> - name of the access point of the cellular operator <login> - login of the cellular operator <password> - password of the cellular operator <protocol> - exchange protocol with the server Where: <x> - server number from 1 to 3, if not, the settings are edited for the first server <y> - SIM card number from 1 to 2, if not available, the settings are edited for the first card

If error occurs while reading the configuration SMS command, SMS message error information will be generated. The settings will not be edited.

Structure of the response SMS error message:

<Title>

<Error type>

<Title> field contains the fixed string "Error!"

<Error type > field can have the following values:

"No settings have been set."

"Parameter # < Parameter number> contains an invalid length".

"Parameter # < Parameter number> contains an invalid character".

"Value for <Parameter number> exceeds the allowed size".

<Parameter number> field is the sequence number of the parameter from the beginning of the SMS, not including the message header

List of excluded characters for parameters of general type: <,>, control characters.

Numeric parameters do not allow all characters except digits.

5. Tone control

When dialing, it is possible to control device using tone commands. The command end sign is always the "*" key (asterisk). If you want to break the link after executing the command, enter the "#" sign (number sign). In this case, the device will disconnect as soon as commands are received. At the same time, commands and password must be entered as additional parameters of a single long phone number after the phone number itself, for example:

+79999999999P1234*1#
where:
+79999999999 - SIM card phone number in the device;
P - Latin letter "P," meaning pause in the set of tones;
1234 * - password with the sign of its completion;
1 - command to activate the first control output O1;
- sign of the end of command entry and disconnecting connection.

Thus, it is possible to enable output control selecting subscriber from the phone book, whose phone number is written in the form: +79999999999P1234*1#, and call him. After receiving the command, the connection will be broken.

№	Command (key sequence)	Description
1	1*	Enable the first control output O1
2	10*	Disable the first control output O1
3	2*	Enable the second control output O2
4	20*	Disable the second control output O2
5	3*	Enable the third control output O3
6	30*	Disable the third control output O3
7	4*	Enable the forth control output O4
8	40*	Disable the forth control output O4
9	54x*	Locking sensor with sequence number **X** = 1..5
10	55x*	Unlocking sensor with sequence number **X** = 1..5
11	549*	Locking internal impact, tilt and movement sensors
12	559*	Unlocking internal impact, tilt and movement sensors

6. Working with device configuration

6.1. Basic principles and description of commands

Remote configuration allows you to remotely change the device configuration without using computer.

Command structure depends on the configuration structure of the device (short example of the structure is given in "6.2. Example of device configuration structure".

Parameters are accessed using the following scheme: Page -> Tag -> Parameters.
Parameters are arranged in strict order according to the configuration structure. The command specifies the parameters from the first to the last to be changed.

If the command is incomplete, then the filled fields will be replaced, and empty (two commas in a row) will be ignored.

If "!" is entered instead of the parameter in the command then the parameter in the device configuration is reset to zero (if the parameter is text - it is erased, if numeric, it is set to 0).

Writing configuration

Sending a command causes the device to reboot

Command	*!EDITS<s><p>:<t>(<a1>,<a2>,…<aX>) Example: *!EDITS TRANS:SRV1(,,,193.193.165.165,20966)
Response	*@EDITS<s><result>,<p>:<t>(<a1>,<a2>,…<aX>) Example: *@EDITS OK,TRANS:SRV1(FLEX,0,1,193.193.165.165,20966)
Exchange channel	GPRS, USB, Bluetooth, SMS
Designations	**Interpretation**
<s>	parameter separator - space.
<p>	name of the page on which the tag is located. Must end with "**:**".
<t>	name of the tag you want to edit. Must end with "**(**".
<aX>	values of parameters written to the device. The last tag parameter must end with "**)**".
<result>	command execution result: - "OK" - command executed; - "FAIL" - error in the command.

Reading configuration

Command	*!READ<s><p>:<t> Example: *!READ TRANS:AP1
Response	*@READ<s><result>,<p>:<t>(<a1>,<a2>,…<aX>) Example: *@READ OK,TRANS:AP1(internet.mts.ru,mts,mts)
Exchange channel	GPRS, USB, Bluetooth, SMS
Designations	**Interpretation**
<s>	parameter separator - space.
<p>	name of the page on which the tag is located. Must end with "**:**".
<t>	name of the tag you want to edit. Must end with "**(**".
<aX>	values of parameters written to the device. The last tag parameter must end with "**)**".
<result>	command execution result: - "OK" - command executed; - "FAIL" - error in the command.

The command format supports accessing multiple pages and tags in one command, for this you need to write the name of the following page or tag separated by a comma:

*!EDITS TRANS:SRV1(FLEX,,,193.193.165.165,20966),AP1(internet.mts.ru,mts,mts)
*!READ TRANS:SRV1,AP1

FLEX bit mask edit command

Sending a command causes the device to reboot

Command	<HEAD>*!SETFM<s><n>:<m>[,<n>:<m>] Example: <HEAD>*!SETFM 7:1111,20:00	
Response	<HEAD>*@SETFM<s><n>:<m>	
Exchange channel	GPRS, USB, Bluetooth, SMS	
Designations	**Interpretation**	**Data Format**
<HEAD>	16 byte NTCB packet header with preamble	16*U8
*!SETFM	0x2A 0x21 0x53 0x45 0x54 0x46 0x4D	char[7]
*@SETFM	0x2A 0x40 0x53 0x45 0x54 0x46 0x4D	char[7]
<s>	Parameter separator - space (0x20).	char
<n>	Index from which the bits will be set, starts with 1. Parameter number corresponds to the number in the "FLEX telemetry record structure" table.	char
<m>	Bit mask.	char

Additionally, to be able to edit the FLEX mask using SMS, the service symbol "**&**" is entered. If you use the construction described below in the EDITS command in the parameter field, this will allow you to edit the parameter bitwise.

Field format	Description of parameters
&<n>[<m>]…<n>[<m>]	Format for bitwise editing FLEX mask settings: <n> - index from which the bits will be set, starts with 1. Parameter number corresponds to the number in the "FLEX telemetry record structure" table. <m> - bit mask.

Example of a command that includes 4 bits in 1 starting with the 6th bit (7th parameter from the FLEX Telemetry Record Structure table) and disables 124 and 125 bits (125th and 126th parameters from the "FLEX telemetry record structure" table):

*!EDITS PROTOCOL:FLEX(,&7[1111]20[00])

6.2. Example of device configuration structure

For a complete configuration structure, see the document "24xx. 26xx. Device Configuration”

Parameter	Value	Type	Zeroing
Page: **TRANS**	Parameters that determine the order in which data is transferred.		
Tag: **OBJECT**	Object parameters.		
name	Object name (up to 64 characters). By default, "NONAME."	Char[24]	Yes
Tag: **SRV1**	Transfer parameters for the 1st server.		
protocol	Data transfer protocol: 1. “F5.2”, 2. “F6”, 3. “FLEX”, 4. "EGTS" - EGST without authorization, 5. "EGTSAUTH" - EGST with authorization. The default is FLEX.	Char[8]	Yes
objID	Object ID. The default is 0.	U32	Yes
dcID	ID of the dispatch center (personal account number). The default is 1.	U32	Yes
addr	IP address or domain name of the server. For example, "89.208.152.54." The default is "0.0.0.0" (This value indicates that the server is not in use).	Char[35]	Yes
port	Port. For example: "5100." The default is "0."	Char[7]	Yes
transProto	Transport layer protocol: – “TCP”, – “UDP”. The default is "TCP."	Char[3]	Yes
Tag: **SRV2**	Parameters of the 2nd server.		
See tag: SRV1			
Tag: **SRV3**	Parameters of the 3rd server.		
See tag: SRV1			
Tag: **AP1**	Access point settings for SIM1.		
name	Name of the access point of the cellular operator. For example: "internet.mts.ru" The default is "". Blank entry in the name, login, pass parameters indicates that the access point is set to Automatic.	Char[30]	Yes
login	Login of the mobile operator. For example: "mts" The default is "". Blank entry in the name, login, pass parameters indicates that the access point is set to Automatic.	Char[20]	Yes
pass	Password of the cellular operator. For example: "mts" The default is "". Blank entry in the name, login, pass parameters indicates that the access point is set to Automatic.	Char[20]	Yes
pin	SIM card PIN. The default is. "" This value disables PIN verification on the SIM card	Char[9]	Yes
Tag: **AP2**	Access point settings for SIM2.		
See tag: AP1			

Annex A. Telemetry record structures

Annex A.1. Structure of FLEX telemetry records

This table contains the current structure of the telemetry packet with separation by versions.

<table>
<tr><th>№</th><th>Record field</th><th>Size of record item</th><th>Data format</th><th>Accepted value</th></tr>
<tr><td colspan="5">FLEX 1.0</td></tr>
<tr><td>1</td><td>End-to-end record number in non-volatile memory</td><td>4</td><td>U32</td><td>Starts from zero, increments on each record. Never decreases.</td></tr>
<tr><td>2</td><td>Event ID corresponding to this record</td><td>2</td><td>U16</td><td>The codes specified in the table are recorded in the protocol</td></tr>
<tr><td>3</td><td>Event time</td><td>4</td><td>U32</td><td>Number of seconds since 1970</td></tr>
<tr><td>4</td><td>Device status</td><td>1</td><td>U8</td><td>Bit field:
<table>
<tr><td>Bits</td><td>Value</td></tr>
<tr><td>0</td><td>Test mode
1 - test mode
0 - operating mode</td></tr>
<tr><td>1</td><td>Alarm alert
1 - enabled
0 - disabled</td></tr>
<tr><td>2</td><td>Alarm
1 - enabled
0 - disabled</td></tr>
<tr><td>3-4</td><td>Operating mode:
0 - surveillance
1 - security
2 - additional security
3 - service</td></tr>
<tr><td>5</td><td>Evacuation:
0 - not fixed
1 - fixed</td></tr>
<tr><td>6</td><td>Power saving mode:
0 - no;
1 - yes.</td></tr>
<tr><td>7</td><td>Accelerometer calibration:
0 - not calibrated
1 - calibrated</td></tr>
</table></td></tr>
</table>

<table>
<tr><td>5</td><td>Status of function modules 1</td><td>1</td><td>U8</td><td>Bit field:<table><tr><th>Categories</th><th>Value</th></tr><tr><td>0</td><td>0 - GSM is on
1 - GSM is off</td></tr><tr><td>1</td><td>0 - USB disabled
1 - USB enabled</td></tr><tr><td>2</td><td>0 - additional high-precision navigation receiver is disabled
1 - additional high-precision navigation receiver is connected</td></tr><tr><td>3</td><td>0 - clock is not synchronized by GPS
1- clock synchronized by GPS</td></tr><tr><td>4</td><td>0 - the first SIM card works
1 - the second SIM card works</td></tr><tr><td>5</td><td>0 - no registration in the cellular network
1- registration in the cellular network</td></tr><tr><td>6</td><td>0 - home cellular network
1- roaming</td></tr><tr><td>7</td><td>0 - engine (generator) is off
1 - engine (generator) is on</td></tr></table></td></tr>
<tr><td>6</td><td>Status of function modules 2</td><td>1</td><td>U8</td><td>Bit field:<table><tr><th>Bits</th><th>Value</th></tr><tr><td>0-1</td><td>0 - no GSM jamming
1 - GSM jamming detected
2 - industrial interference detected</td></tr><tr><td>2</td><td>0 - no GNSS jamming
1 - GNSS jamming detected</td></tr><tr><td>3</td><td>GNSS coordinate averaging:
0 - no;
1 - yes.</td></tr><tr><td>4</td><td>Accelerometer status:
0 - regular operation;
1 - error.</td></tr><tr><td>5</td><td>Bluetooth module status:
0 - off
1 - on</td></tr><tr><td>6</td><td>Wi-Fi module status:
1 - on
0 - off</td></tr><tr><td>7</td><td>Reserve</td></tr></table></td></tr>
<tr><td>7</td><td>GSM layer</td><td>1</td><td>U8</td><td>0: -113 dB/m or less
1: - 111 dB/m
2.. 30: -109.. -53 Db/m
31: -51 dB/m or more
99: no cellular signal.</td></tr>
<tr><td>8</td><td>Status of GPS/GLONASS navigation sensor</td><td>1</td><td>U8</td><td>Bit field:<table><tr><th>Bits</th><th>Values</th></tr><tr><td>0</td><td>0 - navigation receiver is off;
1 - navigation receiver is on.</td></tr><tr><td>1</td><td>0 - not valid navigation;
1 - valid navigation.</td></tr><tr><td>2..7</td><td>Number of navigation satellites 0-32</td></tr></table></td></tr>
<tr><td>9</td><td>Time of last valid coordinates</td><td>4</td><td>U32</td><td>Number of seconds since 1970</td></tr>
</table>

10	Last valid latitude	4	I32	Latitude angle recorded when obtaining the last valid coordinates. ten thousandths of a minute. For example, 55 ° 42,2389'will be represented as 33422389
11	Last valid longitude	4	I32	Longitude angle recorded when obtaining the last valid coordinates. ten thousandths of a minute. For example, 37 ° 41,6063'will be represented as 22616063
12	Last valid height	4	I32	Altitude relative to sea level recorded when obtaining the last valid coordinates: decimeters For example, 205 meters will be presented as 2050 decimeters
13	Speed	4	Float	Speed recorded when obtaining the last valid coordinates. km/h
14	Course	2	U16	Course recorded when obtaining the last valid coordinates. 0° … 360°
15	Current mileage	4	Float	Mileage recorded at the time of the event, calculated during the arrival of valid navigation data. km.
16	Last track segment	4	Float	Mileage calculated between this event and the previous one. (between two track points) km.
17	Total number of seconds in the last track segment	2	U16	Total number of computation points by the navigation receiver at a rate of once per second.
18	Number of seconds on the last track segment along which the mileage was calculated (valid navigation)	2	U16	Number of calculation points by the navigation receiver of coordinates at a rate of once per second with valid navigation data.
19	Main power supply voltage	2	U16	millivolts 0-65535 mV
20	Backup power supply voltage	2	U16	millivolts 0-65535 mV
21	Voltage at analog input 1 (Ain1)	2	U16	millivolts 0-65535 mV
22	Voltage at analog input 2 (Ain2)	2	U16	millivolts 0-65535 mV
23	Voltage at analog input 3 (Ain3)	2	U16	millivolts 0-65535 mV
24	Voltage at analog input 4 (Ain4)	2	U16	millivolts 0-65535 mV
25	Voltage at analog input 5 (Ain5)	2	U16	millivolts 0-65535 mV
26	Voltage at analog input 6 (Ain6)	2	U16	millivolts 0-65535 mV
27	Voltage at analog input 7 (Ain7)	2	U16	millivolts 0-65535 mV
28	Voltage at analog input 8 (Ain8)	2	U16	millivolts 0-65535 mV

<table>
<tr><td>29</td><td>Current readings of discrete sensors 1</td><td>1</td><td>U8</td><td>Bit field:<table><tr><td>Bits</td><td>Values</td></tr><tr><td>0</td><td>input In1</td></tr><tr><td>1</td><td>input In2</td></tr><tr><td>2</td><td>input In3</td></tr><tr><td>3</td><td>input In4</td></tr><tr><td>4</td><td>input In5</td></tr><tr><td>5</td><td>input In6</td></tr><tr><td>6</td><td>input In7</td></tr><tr><td>7</td><td>input In8</td></tr></table>0 - sensor in normal condition
1 - sensor triggered</td></tr>
<tr><td>30</td><td>Current readings of discrete sensors 2</td><td>1</td><td>U8</td><td>Bit field:<table><tr><td>Bits</td><td>Values</td></tr><tr><td>0</td><td>input In9</td></tr><tr><td>1</td><td>input In10</td></tr><tr><td>2</td><td>input In11</td></tr><tr><td>3</td><td>input In12</td></tr><tr><td>4</td><td>input In13</td></tr><tr><td>5</td><td>input In14</td></tr><tr><td>6</td><td>input In15</td></tr><tr><td>7</td><td>input In16</td></tr></table>0 - sensor in normal condition
1 - sensor triggered</td></tr>
<tr><td>31</td><td>Current state of outputs 1</td><td>1</td><td>U8</td><td>Bit field:<table><tr><td>Bits</td><td>Values</td></tr><tr><td>0</td><td>1st output</td></tr><tr><td>1</td><td>2nd output</td></tr><tr><td>2</td><td>3rd output</td></tr><tr><td>3</td><td>4th output</td></tr><tr><td>4</td><td>5th output</td></tr><tr><td>5</td><td>6th output</td></tr><tr><td>6</td><td>7th output</td></tr><tr><td>7</td><td>8th output</td></tr></table>0- output disabled
1- output enabled</td></tr>
<tr><td>32</td><td>Current state of outputs 2</td><td>1</td><td>U8</td><td>Bit field:<table><tr><td>Bits</td><td>Values</td></tr><tr><td>0</td><td>9th output</td></tr><tr><td>1</td><td>10th output</td></tr><tr><td>2</td><td>11th output</td></tr><tr><td>3</td><td>12th output</td></tr><tr><td>4</td><td>13th output</td></tr><tr><td>5</td><td>14th output</td></tr><tr><td>6</td><td>15th output</td></tr><tr><td>7</td><td>16th output</td></tr></table>0- output disabled
1- output enabled</td></tr>
<tr><td>33</td><td>Readings of pulse counter 1</td><td>4</td><td>U32</td><td>Number of pulses counted at the time of the event 0-232</td></tr>
<tr><td>34</td><td>Readings of pulse counter 2</td><td>4</td><td>U32</td><td>Number of pulses counted at the time of the event 0-232</td></tr>
<tr><td>35</td><td>Frequency on analog-frequency fuel level sensor 1</td><td>2</td><td>U16</td><td>Frequency value 0-20000 Hz</td></tr>
<tr><td>36</td><td>Frequency on analog-frequency fuel level sensor 2</td><td>2</td><td>U16</td><td>Frequency value 0-20000 Hz</td></tr>
<tr><td>37</td><td>Engine hours counted at generator operation sensor trigger</td><td>4</td><td>U32</td><td>0-2^{32} sec</td></tr>
</table>

38	Fuel level measured by fuel level sensor 1 RS-485	2	U16	Relative level value for the digital sensor 0 - 65499 - level, 65500 - 65535 - error code (see Annex E) **Notes:** 1. If calibration for the sensor is loaded into the device, then 1 level unit corresponds to 0.1 l of fuel. 2. Depending on the setting, the entire range 0 to 65535 can be used to transfer level.
39	Fuel level measured by fuel level sensor 2 RS-485	2	U16	Relative level value for the digital sensor 0 - 65499 - level, 65500 - 65535 - error code (see Annex E) **Notes:** 1. If calibration for the sensor is loaded into the device, then 1 level unit corresponds to 0.1 l of fuel. 2. Depending on the setting, the entire range 0 to 65535 can be used to transfer level.
40	Fuel level measured by fuel level sensor 3 RS-485	2	U16	Relative level value for the digital sensor 0 - 65499 - level, 65500 - 65535 - error code (see Annex E) **Notes:** 1. If calibration for the sensor is loaded into the device, then 1 level unit corresponds to 0.1 l of fuel. 2. Depending on the setting, the entire range 0 to 65535 can be used to transfer level.
41	Fuel level measured by fuel level sensor 4 RS-485	2	U16	Relative level value for the digital sensor 0 - 65499 - level, 65500 - 65535 - error code (see Annex E) **Notes:** 1. If calibration for the sensor is loaded into the device, then 1 level unit corresponds to 0.1 l of fuel. 2. Depending on the setting, the entire range 0 to 65535 can be used to transfer level.
42	Fuel level measured by fuel level sensor 5 RS-485	2	U16	Relative level value for the digital sensor 0 - 65499 - level, 65500 - 65535 - error code (see Annex E) **Notes:** 1. If calibration for the sensor is loaded into the device, then 1 level unit corresponds to 0.1 l of fuel. 2. Depending on the setting, the entire range 0 to 65535 can be used to transfer level.
43	Fuel level measured by fuel level sensor 6 RS-485	2	U16	Relative level value for the digital sensor 0 - 65499 - level, 65500 - 65535 - error code (see Annex E) **Notes:** 1. If calibration for the sensor is loaded into the device, then 1 level unit corresponds to 0.1 l of fuel. 2. Depending on the setting, the entire range 0 to 65535 can be used to transfer level.
44	Fuel level measured by fuel level sensor RS-232	2	U16	Relative level value for the digital sensor 0 - 65499 - level, 65500 - 65535 - error code (see Annex E) **Notes:** 1. If calibration for the sensor is loaded into the device, then 1 level unit corresponds to 0.1 l of fuel. 2. Depending on the setting, the entire range 0 to 65535 can be used to transfer level.

45	Temperature from digital sensor 1 (in degrees Celsius)	1	I8	-55 ° C... + 125 (-128 ° C - sensor not connected
46	Temperature from digital sensor 2 (in degrees Celsius)	1	I8	-55 ° C... + 125 (-128 ° C - sensor not connected
47	Temperature from digital sensor 3 (in degrees Celsius)	1	I8	-55 ° C... + 125 (-128 ° C - sensor not connected
48	Temperature from digital sensor 4 (in degrees Celsius)	1	I8	-55 ° C... + 125 (-128 ° C - sensor not connected
49	Temperature from digital sensor 5 (in degrees Celsius)	1	I8	-55 ° C... + 125 (-128 ° C - sensor not connected
50	Temperature from digital sensor 6 (in degrees Celsius)	1	I8	-55 ° C... + 125 (-128 ° C - sensor not connected
51	Temperature from digital sensor 7 (in degrees Celsius)	1	I8	-55 ° C... + 125 (-128 ° C - sensor not connected
52	Temperature from digital sensor 8 (in degrees Celsius)	1	I8	-55 ° C... + 125 (-128 ° C - sensor not connected
53	CAN Fuel level in the tank	2	U16	<table><tr><th>Bits</th><th>Values</th></tr><tr><td>0-14</td><td>If bit 15 is equal to one 0-100 in% (accuracy up to 1%) If bit 15 is zero 0-32766 in 0.1 liters (0-3276.6 liters) 32767 (0x7FFF) – parameter not read</td></tr><tr><td>15</td><td>1 - percentage of volume 0 - volume in tenths of Liters</td></tr></table>
54	CAN Total fuel consumption	4	Float	Liters 0 - (3.4 • 10^{38}) L If the value is negative, the parameter is not read
55	CAN Engine RPM	2	U16	Revolutions per minute 0-65534 65535 (0xFFFF) - parameter not read
56	CAN Coolant (engine) temperature	1	I8	In degrees Celsius -127 to 127 ° C -128 ° C (0x80) - parameter not read
57	CAN Total vehicle mileage	4	Float	Kilometers 0 - (3,4 • 10^{38}) km If value is negative, parameter is not read
58	CAN Axle load 1	2	U16	kilograms 0-65534 kg 65535 (0xFFFF) - parameter not read
59	CAN Axle load 2	2	U16	kilograms 0-65534 kg 65535 (0xFFFF) - parameter not read
60	CAN Axle load 3	2	U16	kilograms 0-65534 kg 65535 (0xFFFF) - parameter not read
61	CAN Axle load 4	2	U16	kilograms 0-65534 kg 65535 (0xFFFF) - parameter not read
62	CAN Axle load 5	2	U16	kilograms 0-65534 kg 65535 (0xFFFF) - parameter not read
63	CAN Gas pedal position	1	U8	0-100% 255 (0xFF) - parameter not read
64	CAN Brake pedal position	1	U8	0-100% 255 (0xFF) - parameter not read
65	CAN Engine load	1	U8	0-100% 255 (0xFF) - parameter not read

<table>
<tr><td>66</td><td>CAN
Liquid level in the diesel exhaust filter</td><td>2</td><td>U16</td><td><table><tr><td>Bits</td><td>Value</td></tr><tr><td>0-14</td><td>If bit 15 is equal to one
0-100 in% (accuracy up to 1%)

If bit 15 is zero
0-32766 in 0.1 liters (0-3276.6 liters)
32767 (0x7FFF) - parameter not read</td></tr><tr><td>15</td><td>1 - percentage of volume
0 - volume in tenths of a liter</td></tr></table></td></tr>
<tr><td>67</td><td>CAN
Total engine hours</td><td>4</td><td>U32</td><td>seconds
0-232 sec</td></tr>
<tr><td>68</td><td>CAN
Distance up to maintenance</td><td>2</td><td>I16</td><td>kilometers
0-32767 * 5km
-1 (0xFFFF) - parameter not read</td></tr>
<tr><td>69</td><td>CAN
Vehicle speed</td><td>1</td><td>U8</td><td>kilometers per hour
0-254
255 (0xFF) - parameter not read</td></tr>
<tr><td colspan="5">FLEX 2.0</td></tr>
<tr><td rowspan="8">70</td><td rowspan="8">Navigation information</td><td rowspan="8">8</td><td>U8</td><td>Number of visible GLONASS satellites
0-32</td></tr>
<tr><td>U8</td><td>Number of visible GPS satellites
0-32</td></tr>
<tr><td>U8</td><td>Number of visible Galileo satellites
0-32</td></tr>
<tr><td>U8</td><td>Number of visible Compass satellites
0-32</td></tr>
<tr><td>U8</td><td>Number of visible Beidou satellites
0-32</td></tr>
<tr><td>U8</td><td>Number of visible DORIS satellites
0-32</td></tr>
<tr><td>U8</td><td>Number of visible IRNSS satellites
0-32</td></tr>
<tr><td>U8</td><td>Number of visible QZSS satellites
0-32</td></tr>
<tr><td rowspan="2">71</td><td>HDOP of the operational receiver</td><td rowspan="2">2</td><td>U8</td><td>0.1... 25.0 (1-250, value multiplied by 10)</td></tr>
<tr><td>PDOP of the operational receiver</td><td>U8</td><td>0.1... 25.0 (1-250, value multiplied by 10)</td></tr>
<tr><td>72</td><td>State of additional high-precision navigation receiver</td><td>1</td><td>U8</td><td>Bit field:<table><tr><td>Bits</td><td>Values</td></tr><tr><td>0</td><td>0 - navigation information in float point
1 - navigation information in fixed point</td></tr><tr><td>1</td><td>0 - receiver does not operate in RTK mode;
1 - receiver operates in RTK mode (float or fixed)</td></tr><tr><td>2..7</td><td>Number of navigation satellites 0-32</td></tr></table></td></tr>
<tr><td rowspan="2">73</td><td>Latitude of coordinate from high-precision receiver</td><td rowspan="2">16</td><td>I64</td><td>Latitude angle recorded when obtaining the last valid coordinates.
destimilational fractions of a minute.</td></tr>
<tr><td>Longitude of coordinate from high-precision receiver</td><td>I64</td><td>Longitude angle recorded when obtaining the last valid coordinates.
destimilational fractions of a minute.</td></tr>
</table>

74	Altitude from high-precision receiver	4	I32	Altitude relative to sea level recorded when obtaining the last valid coordinates: millimeters For example, 205 meters will be represented by 205,000 millimeters
75	Course from high-precision receiver	2	U16	Course recorded when obtaining the last valid coordinates. hundredths of a degree. For example, 270 ° will be represented as 27,000
76	Speed from high-precision receiver	4	Float	Speed recorded when obtaining the last valid coordinates. km/h
77	Location-based service (LBS) information	37	U32	Cell id
			U16	Local area code (lac)
			U16	Mobile country code where the base station is located (mcc)
			U16	Mobile network code (mnc)
			S8	Level of radio signal received via this channel at the telephone receiver input. Measured in "decibels to milliwatt" (dBm). It ranges – about 35 dBm - – 111 dBm
	Nearby station #1 (LBS) information		U32	Cell id
			U16	Local area code (lac)
			U16	Mobile country code where the base station is located (mcc)
			U16	Mobile network code (mnc)
			S8	Level of radio signal received via this channel at the telephone receiver input. Measured in "decibels to milliwatt" (dBm). It ranges – about 35 dBm - – 111 dBm
	Nearby station #2 (LBS) information		U32	Cell id
			U16	Local area code (lac)
			U16	Mobile country code where the base station is located (mcc)
			U16	Mobile network code (mnc)
			S8	Level of radio signal received via this channel at the telephone receiver input. Measured in "decibels to milliwatt" (dBm). It ranges – about 35 dBm - – 111 dBm
	Last data time received from LBS		U32	Number of seconds since 1970
78	Temperature measured by fuel level sensor 1 RS485	1	I8	Temperature value for SCL -128 ° C... + 127
79	Temperature measured by fuel level sensor 2 RS485	1	I8	Temperature value for SCL -128 ° C... + 127
80	Temperature measured by fuel level sensor 3 RS485	1	I8	Temperature value for SCL -128 ° C... + 127
81	Temperature measured by fuel level sensor 4 RS485	1	I8	Temperature value for SCL -128 ° C... + 127
82	Temperature measured by fuel level sensor 5 RS485	1	I8	Temperature value for SCL -128 ° C... + 127
83	Temperature measured by fuel level sensor 6 RS485	1	I8	Temperature value for SCL -128 ° C... + 127
84	Fuel level measured by fuel level sensor 7 RS485	3	U16	Relative level value for the digital sensor 0 - 65499 - level, 65500 - 65535 - error code (see Annex E) **Notes:** 1. If calibration for the sensor is loaded into the device, then 1 level unit corresponds to 0.1 l of fuel. 2. Depending on the setting, the entire range 0 to 65535 can be used to transfer level.
	Temperature measured by fuel level sensor 7 RS485		I8	Temperature value for SCL -128 ° C... + 127

85	Fuel level measured by fuel level sensor 8 RS485	3	U16	Relative level value for the digital sensor 0 - 65499 - level, 65500 - 65535 - error code (see Annex E) **Notes:** 1. If calibration for the sensor is loaded into the device, then 1 level unit corresponds to 0.1 l of fuel. 2. Depending on the setting, the entire range 0 to 65535 can be used to transfer level.
	Temperature measured by fuel level sensor 8 RS485		I8	Temperature value for SCL -128 ° C... + 127
86	Fuel level measured by fuel level sensor 9 RS485	3	U16	Relative level value for the digital sensor 0 - 65499 - level, 65500 - 65535 - error code (see Annex E) **Notes:** 1. If calibration for the sensor is loaded into the device, then 1 level unit corresponds to 0.1 l of fuel. 2. Depending on the setting, the entire range 0 to 65535 can be used to transfer level.
	Temperature measured by fuel level sensor 9 RS485		I8	Temperature value for SCL -128 ° C... + 127
87	Fuel level measured by fuel level sensor 10 RS485	3	U16	Relative level value for the digital sensor 0 - 65499 - level, 65500 - 65535 - error code (see Annex E) **Notes:** 1. If calibration for the sensor is loaded into the device, then 1 level unit corresponds to 0.1 l of fuel. 2. Depending on the setting, the entire range 0 to 65535 can be used to transfer level.
	Temperature measured by fuel level sensor 10 RS485		I8	Temperature value for SCL -128 ° C... + 127
88	Fuel level measured by fuel level sensor 11 RS485	3	U16	Relative level value for the digital sensor 0 - 65499 - level, 65500 - 65535 - error code (see Annex E) **Notes:** 1. If calibration for the sensor is loaded into the device, then 1 level unit corresponds to 0.1 l of fuel. 2. Depending on the setting, the entire range 0 to 65535 can be used to transfer level.
	Temperature measured by fuel level sensor 11 RS485		I8	Temperature value for SCL -128 ° C... + 127
89	Fuel level measured by fuel level sensor 12 RS485	3	U16	Relative level value for the digital sensor 0 - 65499 - level, 65500 - 65535 - error code (see Annex E) **Notes:** 1. If calibration for the sensor is loaded into the device, then 1 level unit corresponds to 0.1 l of fuel. 2. Depending on the setting, the entire range 0 to 65535 can be used to transfer level.
	Temperature measured by fuel level sensor 12 RS485		I8	Temperature value for SCL -128 ° C... + 127

90	Fuel level measured by fuel level sensor 13 RS485	3	U16	Relative level value for the digital sensor 0 - 65499 - level, 65500 - 65535 - error code (see Annex E) **Notes:** 1. If calibration for the sensor is loaded into the device, then 1 level unit corresponds to 0.1 l of fuel. 2. Depending on the setting, the entire range 0 to 65535 can be used to transfer level.
	Temperature measured by fuel level sensor 13 RS485		I8	Temperature value for SCL -128 ° C... + 127
91	Fuel level measured by fuel level sensor 14 RS485	3	U16	Relative level value for the digital sensor 0 - 65499 - level, 65500 - 65535 - error code (see Annex E) **Notes:** 1. If calibration for the sensor is loaded into the device, then 1 level unit corresponds to 0.1 l of fuel. 2. Depending on the setting, the entire range 0 to 65535 can be used to transfer level.
	Temperature measured by fuel level sensor 14 RS485		I8	Temperature value for SCL -128 ° C... + 127
92	Fuel level measured by fuel level sensor 15 RS485	3	U16	Relative level value for the digital sensor 0 - 65499 - level, 65500 - 65535 - error code (see Annex E) **Notes:** 1. If calibration for the sensor is loaded into the device, then 1 level unit corresponds to 0.1 l of fuel. 2. Depending on the setting, the entire range 0 to 65535 can be used to transfer level.
	Temperature measured by fuel level sensor 15 RS485		I8	Temperature value for SCL -128 ° C... + 127
93	Fuel level measured by fuel level sensor 16 RS485	3	U16	Relative level value for the digital sensor 0 - 65499 - level, 65500 - 65535 - error code (see Annex E) **Notes:** 1. If calibration for the sensor is loaded into the device, then 1 level unit corresponds to 0.1 l of fuel. 2. Depending on the setting, the entire range 0 to 65535 can be used to transfer level.
	Temperature measured by fuel level sensor 16 RS485		I8	Temperature value for SCL -128 ° C... + 127
94	Information about 1st tire pressure sensor	6	U8	Wheel No. If no sensor = 0
			U8	Pressure in 0.1 bar If no sensor = 0
			I8	Temperature in degrees, if no sensor = -128
	Information about 2nd tire pressure sensor		U8	Wheel No. If no sensor = 0
			U8	Pressure in 0.1 bar If no sensor = 0
			I8	Temperature in degrees, if no sensor = -128
95	Information about 3rd tire pressure sensor	12	U8	Wheel No. If no sensor = 0
			U8	Pressure in 0.1 bar If no sensor = 0
			I8	Temperature in degrees, if no sensor = -128

	Information about 4th tire pressure sensor		U8	Wheel No. If no sensor = 0
			U8	Pressure in 0.1 bar If no sensor = 0
			I8	Temperature in degrees, if no sensor = -128
	Information about 5th tire pressure sensor		U8	Wheel No. If no sensor = 0
			U8	Pressure in 0.1 bar If no sensor = 0
			I8	Temperature in degrees, if no sensor = -128
	Information about 6th tire pressure sensor		U8	Wheel No. If no sensor = 0
			U8	Pressure in 0.1 bar If no sensor = 0
			I8	Temperature in degrees, if no sensor = -128
96	Information about 7th tire pressure sensor	24	U8	Wheel No. If no sensor = 0
			U8	Pressure in 0.1 bar If no sensor = 0
			I8	Temperature in degrees, if no sensor = -128
	Information about 8th tire pressure sensor		U8	Wheel No. If no sensor = 0
			U8	Pressure in 0.1 bar If no sensor = 0
			I8	Temperature in degrees, if no sensor = -128
	Information about 9th tire pressure sensor		U8	Wheel No. If no sensor = 0
			U8	Pressure in 0.1 bar If no sensor = 0
			I8	Temperature in degrees, if no sensor = -128
	Information about 10th tire pressure sensor		U8	Wheel No. If no sensor = 0
			U8	Pressure in 0.1 bar If no sensor = 0
			I8	Temperature in degrees, if no sensor = -128
	Information about 11th tire pressure sensor		U8	Wheel No. If no sensor = 0
			U8	Pressure in 0.1 bar If no sensor = 0
			I8	Temperature in degrees, if no sensor = -128
	Information about 12th tire pressure sensor		U8	Wheel No. If no sensor = 0
			U8	Pressure in 0.1 bar If no sensor = 0
			I8	Temperature in degrees, if no sensor = -128
	Information about 13th tire pressure sensor		U8	Wheel No. If no sensor = 0
			U8	Pressure in 0.1 bar If no sensor = 0
			I8	Temperature in degrees, if no sensor = -128
	Information about 14th tire pressure sensor		U8	Wheel No. If no sensor = 0

			U8	Pressure in 0.1 bar If no sensor = 0
			I8	Temperature in degrees, if no sensor = -128
97	Information about 15th tire pressure sensor	48	U8	Wheel No. If no sensor = 0
			U8	Pressure in 0.1 bar If no sensor = 0
			I8	Temperature in degrees, if no sensor = -128
	Information about 16th tire pressure sensor		U8	Wheel No. If no sensor = 0
			U8	Pressure in 0.1 bar If no sensor = 0
			I8	Temperature in degrees, if no sensor = -128
	Information about 17th tire pressure sensor		U8	Wheel No. If no sensor = 0
			U8	Pressure in 0.1 bar If no sensor = 0
			I8	Temperature in degrees, if no sensor = -128
	Information about 19th tire pressure sensor		U8	Wheel No. If no sensor = 0
			U8	Pressure in 0.1 bar If no sensor = 0
			I8	Temperature in degrees, if no sensor = -128
	Information about 20th tire pressure sensor		U8	Wheel No. If no sensor = 0
			U8	Pressure in 0.1 bar If no sensor = 0
			I8	Temperature in degrees, if no sensor = -128
	Information about 21st tire pressure sensor		U8	Wheel No. If no sensor = 0
			U8	Pressure in 0.1 bar If no sensor = 0
			I8	Temperature in degrees, if no sensor = -128
	Information about 22nd tire pressure sensor		U8	Wheel No. If no sensor = 0
			U8	Pressure in 0.1 bar If no sensor = 0
			I8	Temperature in degrees, if no sensor = -128
	Information about 23rd tire pressure sensor		U8	Wheel No. If no sensor = 0
			U8	Pressure in 0.1 bar If no sensor = 0
			I8	Temperature in degrees, if no sensor = -128
	Information about 24th tire pressure sensor		U8	Wheel No. If no sensor = 0
			U8	Pressure in 0.1 bar If no sensor = 0
			I8	Temperature in degrees, if no sensor = -128
	Information about 25th tire pressure sensor		U8	Wheel No. If no sensor = 0
			U8	Pressure in 0.1 bar If no sensor = 0

			I8	Temperature in degrees, if no sensor = -128
	Information about 26th tire pressure sensor		U8	Wheel No. If no sensor = 0
			U8	Pressure in 0.1 bar If no sensor = 0
			I8	Temperature in degrees, if no sensor = -128
	Information about 27th tire pressure sensor		U8	Wheel No. If no sensor = 0
			U8	Pressure in 0.1 bar If no sensor = 0
			I8	Temperature in degrees, if no sensor = -128
	Information about 28th tire pressure sensor		U8	Wheel No. If no sensor = 0
			U8	Pressure in 0.1 bar If no sensor = 0
			I8	Temperature in degrees, if no sensor = -128
	Information about 29th tire pressure sensor		U8	Wheel No. If no sensor = 0
			U8	Pressure in 0.1 bar If no sensor = 0
			I8	Temperature in degrees, if no sensor = -128
	Information about 30th tire pressure sensor		U8	Wheel No. If no sensor = 0
			U8	Pressure in 0.1 bar If no sensor = 0
			I8	Temperature in degrees, if no sensor = -128
	Information about 31st tire pressure sensor		U8	Wheel No. If no sensor = 0
			U8	Pressure in 0.1 bar If no sensor = 0
			I8	Temperature in degrees, if no sensor = -128
98	Tachograph data: Driver activity and status of card slots.	1	U8	Bit field: Bits: Values 0..1: Driver activity 1: 0 - Rest, 1 – Ready to work, 2 - Work is not related to vehicle driving, 3 - Vehicle driving 2..3: Driver card slot 1: 0 - No card, 1 - Not authorized, 2 - Authorized, 3 - Failed to eject 4..5: Driver activity 2: 0 - Rest, 1 – Ready to work, 2 - Work is not related to vehicle driving, 3 - Vehicle driving 6..7: Driver card slot 2: 0 - No card, 1 - Not authorized, 2 - Authorized, 3 - Failed to eject

<table>
<tr><td>99</td><td>Tachograph operation mode/card</td><td>1</td><td>U8</td><td>0 - Tachograph disabled,
1 - Driver,
2 - Master,
3 - Inspector,
4 - Enterprise,
5 - Crew.</td></tr>
<tr><td>100</td><td>Status flags from tachograph</td><td>2</td><td>U16</td><td>Bit field:<table><tr><th>Bits</th><th>Values</th></tr><tr><td>0</td><td>Ignition:
0 - off;
1 - on.</td></tr><tr><td>1</td><td>GND disabled:
0 - no;
1 - yes.</td></tr><tr><td>2</td><td>"Ferry/Train" mode:
0 - off;
1 - on.</td></tr><tr><td>3</td><td>"Not applicable" mode:
0 - off;
1 - on.</td></tr><tr><td>4</td><td>Backlight:
0 - off;
1 - on.</td></tr><tr><td>5</td><td>Tachograph communication error:
0 - no;
1 - yes.</td></tr><tr><td>6-8</td><td>Driver 1 working modes defined in time:
0 - no time limit warnings found;
1 - limit #1:15 min before the onset of 4.5 hours of continuous driving;
2 - limit #2: exceeding 4.5 hours of continuous driving
3 - limit #3: 15 min to additional Warning 1
4 - constraint #4: Warning 1 occurred
5 - limit #5: 15 min to additional Warning 2
6 - Constraint # 6: Warning 2 occurred
7 - reserve.</td></tr><tr><td>9-11</td><td>Driver 2 working modes defined in time:
See "Driver 1 working modes defined in time”</td></tr><tr><td>12-15</td><td>Reserve</td></tr></table></td></tr>
<tr><td>101</td><td>Speed by tachograph</td><td>1</td><td>U8</td><td>0-254 km/h
No value - 255.</td></tr>
<tr><td>102</td><td>Odometer by tachograph</td><td>4</td><td>U32</td><td>Value of 1/10 km
No value - 0xFFFFFFFF.</td></tr>
<tr><td>103</td><td>Time by tachograph</td><td>4</td><td>U32</td><td>Number of seconds since 1970</td></tr>
</table>

104	Current status of the driver received from display module	1	U8	0 - driver status unknown (no display); 1 - "On call"; 2 - "On the road"; 3 - "Available"; 4 - "Waiting"; 5 - "Return"; 6 - "Reserve"; 7 - "Working"; 8 - "Break"; 9 - "Ready"; 10 - "Lunch"; 11 - "Rest"; 12 - "Repair"; 13 - "Loading"; 14 - "Unloading"; 15 - "Breakdown"; 16 - "Accident"
105	Index of the last message received/read on the display module.	4	U32	0 - no messages received/read; 0xFFFFFFFF - received/read message sent by NTCT command; Other values correspond to the index of the last received/read message
106	Time increment relative to previous record	2	U16	0-65534 - 1/100 second 0xFFFF - value irrelevant, overflow occurred
107	X-axis linear acceleration	6	I16	-24000.. + 24000 (-24.. + 24 g multiplied by 1000) No value: -32768
	Y-axis linear acceleration		I16	-24000.. + 24000 (-24.. + 24 g multiplied by 1000) No value: -32768
	Z-axis linear acceleration		I16	-24000.. + 24000 (-24.. + 24 g multiplied by 1000) No value: -32768
108	EcoDriving. Threshold exceeding duration	2	U16	0-65534 - 0.01 seconds 0xFFFF - value is irrelevant, duration is more than 655.34 sec.
109	EcoDriving. Maximum value of positive acceleration for the period	6	I16	0..+24000 (-24000..+24000 mg)
	EcoDriving. Maximum value of negative acceleration (braking) for the period		I16	0..+24000 (-24000..+24000 mg)
	EcoDriving. Maximum value of lateral acceleration for the period		I16	0..+24000 (-24000..+24000 mg)
110	Data of passenger flow counters 1	2	U8	0-254 255 - no data
	Data of passenger flow counters 2		U8	0-254 255 - no data
111	Data of passenger flow counters 3	2	U8	0-254 255 - no data
	Data of passenger flow counters 4		U8	0-254 255 - no data
112	Data of passenger flow counters 5	2	U8	0-254 255 - no data
	Data of passenger flow counters 6		U8	0-254 255 - no data
113	Data of passenger flow counters 7	2	U8	0-254 255 - no data
	Data of passenger flow counters 8		U8	0-254 255 - no data
114	Data of passenger flow counters 9	2	U8	0-254 255 - no data
	Data of passenger flow counters 10		U8	0-254 255 - no data

<table>
<tr><td rowspan="2">115</td><td>Data of passenger flow counters 11</td><td rowspan="2">2</td><td>U8</td><td>0-254
255 - no data</td></tr>
<tr><td>Data of passenger flow counters 12</td><td>U8</td><td>0-254
255 - no data</td></tr>
<tr><td rowspan="2">116</td><td>Data of passenger flow counters 13</td><td rowspan="2">2</td><td>U8</td><td>0-254
255 - no data</td></tr>
<tr><td>Data of passenger flow counters 14</td><td>U8</td><td>0-254
255 - no data</td></tr>
<tr><td rowspan="2">117</td><td>Data of passenger flow counters 15</td><td rowspan="2">2</td><td>U8</td><td>0-254
255 - no data</td></tr>
<tr><td>Data of passenger flow counters 16</td><td>U8</td><td>0-254
255 - no data</td></tr>
<tr><td>118</td><td>Autoinformer status</td><td>1</td><td>U8</td><td>Bit field:<table><tr><td>Bits</td><td>Values</td></tr><tr><td>0</td><td>0 - autoinformer disabled
1 - autoinformer enabled</td></tr><tr><td>1</td><td>0 - object outside the geofence
1 - object inside the geofence</td></tr><tr><td>2</td><td>0 - geofence does not correspond to the route
1 - geofence corresponds to the route</td></tr><tr><td>3</td><td>0 - no errors
1 - error on the route</td></tr><tr><td>4</td><td>0 - no errors
1 - error when working with SD card</td></tr><tr><td>5</td><td>0 - no violation
1 - violation of driving mode</td></tr><tr><td>6</td><td>0 - automatic mode
1 - manual mode</td></tr><tr><td>7</td><td>reserve</td></tr></table></td></tr>
<tr><td>119</td><td>Last geofence ID</td><td>2</td><td>U16</td><td>1-65535
0 - was not in the geofence</td></tr>
<tr><td>120</td><td>Last stop ID</td><td>2</td><td>U16</td><td>1-65535
0 - did not arrive at the stop</td></tr>
<tr><td>121</td><td>Current route ID</td><td>2</td><td>U16</td><td>1-65535
0 - route not set</td></tr>
<tr><td>122</td><td>Camera status</td><td>1</td><td>U8</td><td>Bit field:<table><tr><td>Bits</td><td>Values</td></tr><tr><td>0</td><td>0 - camera not available
1 - camera available</td></tr><tr><td>1</td><td>0 - automatic shooting is off
1 - automatic shooting is on</td></tr><tr><td>2</td><td>0 - normal operation mode
1 - not enough card space</td></tr><tr><td>3</td><td>0 - no errors
1 - error when working with SD card</td></tr><tr><td>4..7</td><td>reserve</td></tr></table></td></tr>
</table>

<table>
<tr><th colspan="5">FLEX 3.0</th></tr>
<tr><td>123</td><td>Device 2 status</td><td>1</td><td>U8</td><td>Bit field:<table><tr><th>Bits</th><th>Value</th></tr><tr><td>0</td><td>Tamper button:
0 - normal;
1 - case is opened.</td></tr><tr><td>1-2</td><td>Navigation antenna status:
0 - no measurements are made
1 - closing in GNSS antenna
2 - GNSS antenna not connected/break
3 - GNSS antenna connected</td></tr><tr><td>3-7</td><td>Reserve</td></tr></table></td></tr>
<tr><td>124</td><td>Status of function modules 3</td><td>1</td><td>U8</td><td>Bit field:<table><tr><th>Bits</th><th>Value</th></tr><tr><td>0</td><td>Iridium module status:
1 - on;
0 - off.</td></tr><tr><td>1</td><td>Inertial navigation module status:
1 - on;
0 - off.</td></tr><tr><td>2-7</td><td>Reserve</td></tr></table></td></tr>
<tr><td>125</td><td>Communication status</td><td>1</td><td>U8</td><td>Bit Field:<table><tr><th>Bits</th><th>Value</th></tr><tr><td>0-2</td><td>Communication status:
0 - missing;
1 - via GSM;
2 - via GPRS;
3 - via WiFi;
4 - via Iridium;
5 – WCDMA (3G);
6.. 7 - reserve</td></tr><tr><td>3</td><td>Working with server 1
0 - not connected;
1 - connected.</td></tr><tr><td>4</td><td>Working with server 2
0 - not connected;
1 - connected.</td></tr><tr><td>5</td><td>Working with server 3
0 - not connected;
1 - connected.</td></tr><tr><td>6</td><td>Working with RCS/RFU
0 - not connected;
1 - connected.</td></tr><tr><td>7</td><td>Working with configurator via Bluetooth:
0 - not connected;
1 - connected.</td></tr></table></td></tr>
</table>

<table>
<tr><td>126</td><td>Current readings of discrete sensors 3</td><td>1</td><td>U8</td><td>Bit field:<table><tr><td>Bits</td><td>Values</td></tr><tr><td>0</td><td>input In17</td></tr><tr><td>1</td><td>input In18</td></tr><tr><td>2</td><td>input In19</td></tr><tr><td>3</td><td>input In20</td></tr><tr><td>4</td><td>input In21</td></tr><tr><td>5</td><td>input In22</td></tr><tr><td>6</td><td>input In23</td></tr><tr><td>7</td><td>input In24</td></tr></table>0 - sensor in normal state;
1 – sensor triggered.</td></tr>
<tr><td>127</td><td>Pulse counter 3 readings</td><td>4</td><td>U32</td><td>Number of pulses counted at the time of the event 0-232</td></tr>
<tr><td>128</td><td>Pulse counter 4 readings</td><td>4</td><td>U32</td><td>Number of pulses counted at the time of the event 0-232</td></tr>
<tr><td>129</td><td>Pulse counter 5 readings</td><td>4</td><td>U32</td><td>Number of pulses counted at the time of the event 0-232</td></tr>
<tr><td>130</td><td>Pulse counter 6 readings</td><td>4</td><td>U32</td><td>Number of pulses counted at the time of the event 0-232</td></tr>
<tr><td>131</td><td>Pulse counter 7 readings</td><td>4</td><td>U32</td><td>Number of pulses counted at the time of the event 0-232</td></tr>
<tr><td>132</td><td>Pulse counter 8 readings</td><td>4</td><td>U32</td><td>Number of pulses counted at the time of the event 0-232</td></tr>
<tr><td>133</td><td>Frequency on analog frequency sensor 3</td><td>2</td><td>U16</td><td>Frequency value 0-20000 Hz</td></tr>
<tr><td>134</td><td>Frequency on analog frequency sensor 4</td><td>2</td><td>U16</td><td>Frequency value 0-20000 Hz</td></tr>
<tr><td>135</td><td>Frequency on analog frequency sensor 5</td><td>2</td><td>U16</td><td>Frequency value 0-20000 Hz</td></tr>
<tr><td>136</td><td>Frequency on analog frequency sensor 6</td><td>2</td><td>U16</td><td>Frequency value 0-20000 Hz</td></tr>
<tr><td>137</td><td>Frequency on analog frequency sensor 7</td><td>2</td><td>U16</td><td>Frequency value 0-20000 Hz</td></tr>
<tr><td>138</td><td>Frequency on analog frequency sensor 8</td><td>2</td><td>U16</td><td>Frequency value 0-20000 Hz</td></tr>
<tr><td>139</td><td>Accelerometer virtual sensor status</td><td>1</td><td>U8</td><td>Bit field:<table><tr><td>Bits</td><td>Value</td></tr><tr><td>0</td><td>SH1 sensor status</td></tr><tr><td>1</td><td>SH2 sensor status</td></tr><tr><td>2</td><td>SH3 sensor status</td></tr><tr><td>3</td><td>SH4 sensor status</td></tr><tr><td>4</td><td>WAKEUP sensor state</td></tr><tr><td>5..7</td><td>Reserve</td></tr></table>0 - normal;
1 - triggered.</td></tr>
<tr><td>140</td><td>Internal tilt angle sensor. Angle of inclination relative to local vertical</td><td>1</td><td>U8</td><td>degrees from 0 to 25 in 0.25 degree increments.</td></tr>
<tr><td rowspan="2">141</td><td rowspan="2">Internal tilt sensor. Inclination angles relative to the vertical line</td><td rowspan="2">2</td><td>I8</td><td>Pitch angle degrees
-90 to 90 in increments of 1
forward tilt < 0
tilt back ></td></tr>
<tr><td>I8</td><td>Roll angle degrees
-180 to 180 in 1.5 increments
left roll < 0
roll to the right > 0</td></tr>
<tr><td rowspan="3">142</td><td rowspan="3">External tilt angle sensor. Axis deviations</td><td rowspan="3">3</td><td>U8</td><td>X axis in degrees
to 180 in increments of 1</td></tr>
<tr><td>U8</td><td>Y-axis in degrees
to 180 in increments of 1</td></tr>
<tr><td>U8</td><td>Z-axis in degrees
to 180 in increments of 1</td></tr>
</table>

<table>
<tr><td>143</td><td>EcoDriving. Maximum value of vertical acceleration per period</td><td>2</td><td>I16</td><td>0..+24000 (0..+24000 mg)</td></tr>
<tr><td>144</td><td>EcoDriving. Maximum speed per period</td><td>1</td><td>U8</td><td>0.. 254 km/h</td></tr>
<tr><td>145</td><td>EcoDriving. Speed thresholds state</td><td>1</td><td>U8</td><td>Bit field:<table><tr><td>Bits</td><td>Value</td></tr><tr><td>0</td><td>Threshold 1 state</td></tr><tr><td>1</td><td>Threshold 2 state</td></tr><tr><td>2</td><td>Threshold 3 state</td></tr><tr><td>3</td><td>Threshold 4 state</td></tr><tr><td>4</td><td>Threshold 5 state</td></tr><tr><td>5</td><td>Threshold 6 state</td></tr><tr><td>6..7</td><td>Reserve</td></tr></table>0 – threshold in normal;
1 - threshold triggered.</td></tr>
<tr><td rowspan="3">146</td><td rowspan="3">EcoDriving. Acceleration thresholds state</td><td rowspan="3">3</td><td>U8</td><td>Bit field:<table><tr><td>Bits</td><td>Value</td></tr><tr><td>0</td><td>Acceleration threshold 1 state</td></tr><tr><td>1</td><td>Acceleration threshold 2 state</td></tr><tr><td>2</td><td>Acceleration threshold 3 state</td></tr><tr><td>3</td><td>Acceleration threshold 4 state</td></tr><tr><td>4</td><td>Braking threshold 1 state</td></tr><tr><td>5</td><td>Braking threshold 2 state</td></tr><tr><td>6</td><td>Braking threshold 3 state</td></tr><tr><td>7</td><td>Braking threshold 4 state</td></tr></table>0 – threshold in normal;
1 - threshold triggered.</td></tr>
<tr><td>U8</td><td>Bit field:<table><tr><td>Bits</td><td>Value</td></tr><tr><td>0</td><td>Leftward acceleration threshold 1 state</td></tr><tr><td>1</td><td>Leftward acceleration threshold 2 state</td></tr><tr><td>2</td><td>Leftward acceleration threshold 3 state</td></tr><tr><td>3</td><td>Leftward acceleration threshold 4 state</td></tr><tr><td>4</td><td>Rightward acceleration threshold 1 state</td></tr><tr><td>5</td><td>Rightward acceleration threshold 1 state</td></tr><tr><td>6</td><td>Rightward acceleration threshold 1 state</td></tr><tr><td>7</td><td>Rightward acceleration threshold 1 state</td></tr></table>0 – threshold in normal;
1 - threshold triggered.</td></tr>
<tr><td>U8</td><td>Bit field:<table><tr><td>Bits</td><td>Value</td></tr><tr><td>0</td><td>Vertical acceleration threshold 1 state</td></tr><tr><td>1</td><td>Vertical acceleration threshold 2 state</td></tr><tr><td>2</td><td>Vertical acceleration threshold 3 state</td></tr><tr><td>3</td><td>Vertical acceleration threshold 4 state</td></tr><tr><td>4..7</td><td>Reserve</td></tr></table>0 – threshold in normal;
1 - threshold triggered.</td></tr>
<tr><td>147</td><td>Frequency at output of FLS 485 #1</td><td>2</td><td>U16</td><td>0... 65534 Hz
65535 - no data</td></tr>
<tr><td>148</td><td>Frequency at output of FLS 485 #2</td><td>2</td><td>U16</td><td>0... 65534 Hz
65535 - no data</td></tr>
<tr><td>149</td><td>Frequency at output of FLS 485 #3</td><td>2</td><td>U16</td><td>0... 65534 Hz
65535 - no data</td></tr>
<tr><td>150</td><td>Frequency at output of FLS 485 #4</td><td>2</td><td>U16</td><td>0... 65534 Hz
65535 - no data</td></tr>
<tr><td>151</td><td>Frequency at output of FLS 485 #5</td><td>2</td><td>U16</td><td>0... 65534 Hz
65535 - no data</td></tr>
</table>

152	Frequency at output of FLS 485 #6	2	U16	0... 65534 Hz 65535 - no data
153	Frequency at output of FLS 485 #7	2	U16	0... 65534 Hz 65535 - no data
154	Frequency at output of FLS 485 #8	2	U16	0... 65534 Hz 65535 - no data
155	Frequency at output of FLS 485 #9	2	U16	0... 65534 Hz 65535 - no data
156	Frequency at output of FLS 485 #10	2	U16	0... 65534 Hz 65535 - no data
157	Frequency at output of FLS 485 #11	2	U16	0... 65534 Hz 65535 - no data
158	Frequency at output of FLS 485 #12	2	U16	0... 65534 Hz 65535 - no data
159	Frequency at output of FLS 485 #13	2	U16	0... 65534 Hz 65535 - no data
160	Frequency at output of FLS 485 #14	2	U16	0... 65534 Hz 65535 - no data
161	Frequency at output of FLS 485 #15	2	U16	0... 65534 Hz 65535 - no data
162	Frequency at output of FLS 485 #16	2	U16	0... 65534 Hz 65535 - no data
163	High-precision temperature sensor 1	2	I16	− 273.15 to + 1638.35 in 0.05 increments 0x8000 - no data
164	High-precision temperature sensor 2	2	I16	− 273.15 to + 1638.35 in 0.05 increments 0x8000 - no data
165	High-precision temperature sensor 3	2	I16	− 273.15 to + 1638.35 in 0.05 increments 0x8000 - no data
166	High-precision temperature sensor 4	2	I16	− 273.15 to + 1638.35 in 0.05 increments 0x8000 - no data
167	High-precision humidity sensor 1	1	U8	0 to 100 in 0.5% increments 0xFF - No Data
168	High-precision humidity sensor 2	1	U8	0 to 100 in 0.5% increments 0xFF - No Data
169	High-precision humidity sensor 3	1	U8	0 to 100 in 0.5% increments 0xFF - No Data
170	High-precision humidity sensor 4	1	U8	0 to 100 in 0.5% increments 0xFF - No Data
171	Fuel flow sensor. Sensor status	2	U16	Bit field: Bits: Value 0..3: Fuel feed line operating mode 4..7: Fuel return line operating mode 8..11: Engine operating mode by flow rate 12..13: Power status 14..15: Reserve Operating modes of chambers and engine: 0 - idle run 1 - nominal mode 2 - overload 3 - tampering 4 - negative 5 - interference 6.. 15 - reserve
172	Fuel flow sensor. Fault information	4	U32	Bit field: Bits: Value 0..31: Reserve
173	Fuel flow sensor. Total fuel consumption	4	U32	0.01 l. No value: 0xFFFFFFFF
174	Fuel flow sensor. Fuel consumption per trip	4	U32	0.01 l. No value: 0xFFFFFFFF
175	Fuel flow sensor. Current flow rate	2	I16	0.1 l/h. No value: 0x8000

<table>
<tr><td>176</td><td>Fuel flow sensor. Total fuel volume of the feed line</td><td>4</td><td>U32</td><td>0.01 l.
No value: 0xFFFFFFFF</td></tr>
<tr><td>177</td><td>Fuel flow sensor. Current flow rate of the feed line</td><td>2</td><td>I16</td><td>0.1 l/h
No value: 0x8000</td></tr>
<tr><td>178</td><td>Fuel flow sensor. Temperature of the feed line</td><td>2</td><td>I16</td><td>0.1 ° C
No value: 0x8000</td></tr>
<tr><td>179</td><td>Fuel flow sensor. Total fuel volume of the return line</td><td>4</td><td>U32</td><td>0.01 L
No value: 0xFFFFFFFF</td></tr>
<tr><td>180</td><td>Fuel flow sensor. Current flow rate of the return line</td><td>2</td><td>I16</td><td>0.1 l/h
No value: 0x8000</td></tr>
<tr><td>181</td><td>Fuel flow sensor. Temperature of the return line</td><td>2</td><td>I16</td><td>0.1 ° C
No value: 0x8000</td></tr>
<tr><td rowspan="2">182</td><td rowspan="2">Refrigeration unit.
Unit status</td><td rowspan="2">2</td><td>U8</td><td>Bit field:<table><tr><td>Bit</td><td>Value</td></tr><tr><td>0</td><td>Communication with refrigeration unit:
0 – no communication, 1 - in communication</td></tr><tr><td>1</td><td>Door status:
0 - closed, 1 - open.</td></tr><tr><td>2..4</td><td>Unit type:
0 - unknown;
1 - ThermoKing SLX series;
2 – Carrier Standard32;
3 – Zanotti;
4 – ThermalMaster;
5 – Carrier NDP33LN6FB.</td></tr><tr><td>5..7</td><td>Reserve</td></tr></table></td></tr>
<tr><td>U8</td><td>Operating mode:
0 - no data
1 - engine is off;
2 - heating;
3 - cooling;
4 – thawing;
129... 170 - one of the states of the ThermoKing SLX series installation (see Annex A4);

Other values - reserve.</td></tr>
<tr><td>183</td><td>Refrigeration unit. Refrigeration temperature in section 1 (Refrigeration temperature)</td><td>2</td><td>I16</td><td>-273.15 to 327.67 in 0.01 °C increments
No value: 0x8000</td></tr>
<tr><td>184</td><td>Refrigeration unit. Refrigeration temperature in section 2</td><td>2</td><td>I16</td><td>-273.15 to 327.67 in 0.01 °C increments
No value: 0x8000</td></tr>
<tr><td>185</td><td>Refrigeration unit. Refrigeration temperature in section 3</td><td>2</td><td>I16</td><td>-273.15 to 327.67 in 0.01 °C increments
No value: 0x8000</td></tr>
<tr><td>186</td><td>Refrigeration unit. Temperature set 1</td><td>2</td><td>I16</td><td>-273.15 to 327.67 in 0.01 °C increments
No value: 0x8000</td></tr>
<tr><td>187</td><td>Refrigeration unit. Temperature set 2</td><td>2</td><td>I16</td><td>-273.15 to 327.67 in 0.01 °C increments
No value: 0x8000</td></tr>
<tr><td>188</td><td>Refrigeration unit. Temperature set 3</td><td>2</td><td>I16</td><td>-273.15 to 327.67 in 0.01 °C increments
No value: 0x8000</td></tr>
<tr><td>189</td><td>Refrigeration unit. Ambient air temperature</td><td>2</td><td>I16</td><td>-273.15 to 327.67 in 0.01 °C increments
No value: 0x8000</td></tr>
<tr><td>190</td><td>Refrigeration unit. Coolant temperature</td><td>2</td><td>I16</td><td>-273.15 to 327.67 in 0.01 °C increments
No value: 0x8000</td></tr>
<tr><td>191</td><td>Refrigeration unit. Battery voltage</td><td>2</td><td>U16</td><td>0 to 65534 in 1mV increments
No value: 0xFFFF</td></tr>
<tr><td>192</td><td>Refrigeration unit. Battery current</td><td>2</td><td>U16</td><td>0 to 65534 in 1 mA increments
No value: 0xFFFF</td></tr>
<tr><td>193</td><td>Refrigeration unit. Engine hours of operation from the engine</td><td>4</td><td>U32</td><td>0 to 42949672.94 in 0.01 h increments
No value: 0xFFFFFFFF</td></tr>
</table>

<table>
<tr><td>194</td><td>Refrigeration unit. Engine hours of operation from the network</td><td>4</td><td>U32</td><td>0 to 42949672.94 in 0.01 h increments
No value: 0xFFFFFFFF</td></tr>
<tr><td rowspan="2">195</td><td>Refrigeration unit. Number of errors</td><td rowspan="2">4</td><td>U16</td><td>0…65535</td></tr>
<tr><td>Refrigeration unit. Most important error code</td><td>U16</td><td>0…65535</td></tr>
<tr><td rowspan="2">196</td><td>Refrigeration unit. Code 2nd most important error</td><td rowspan="2">4</td><td>U16</td><td>0…65535</td></tr>
<tr><td>Refrigeration unit. Code 3rd most important error</td><td>U16</td><td>0…65535</td></tr>
<tr><td rowspan="3">197</td><td>Refrigeration unit. Code 4th most important error</td><td rowspan="3">6</td><td>U16</td><td>0…65535</td></tr>
<tr><td>Refrigeration unit. Code 5th most important error</td><td>U16</td><td>0…65535</td></tr>
<tr><td>Refrigeration unit. Code 6th most important error</td><td>U16</td><td>0…65535</td></tr>
<tr><td rowspan="2">198</td><td rowspan="2">Refrigeration unit. Engine condition</td><td rowspan="2">3</td><td>U8</td><td>Operating mode:
0 - no data;
1 - no RPM, engine is stopped;
2 - diesel engine works, low speed;
3 - diesel engine works, high speed;
4 – powered by the network.</td></tr>
<tr><td>U16</td><td>Engine RPM:
From 0 to 65534 rpm.
0xFFFF - no data.</td></tr>
<tr><td>199</td><td>Refrigeration unit. Compressor configuration</td><td>1</td><td>U8</td><td>0 - no data;
1 - Start/Stop - mode with engine shutdown;
2 - Continuous - continuous operating mode
129... 156 - one of the ThermoKing SLX series unit modes of the (See Annex A4);

Other values - reserve.</td></tr>
<tr><td>200</td><td>Information about location in geofences</td><td>2</td><td>U16</td><td>Bit field:
<table>
<tr><td>Bits</td><td>Value</td></tr>
<tr><td>0</td><td>Geofence 1</td></tr>
<tr><td>1</td><td>Geofence 2</td></tr>
<tr><td>2</td><td>Geofence 3</td></tr>
<tr><td>3</td><td>Geofence 4</td></tr>
<tr><td>4</td><td>Geofence 5</td></tr>
<tr><td>5</td><td>Geofence 6</td></tr>
<tr><td>6</td><td>Geofence 7</td></tr>
<tr><td>7</td><td>Geofence 8</td></tr>
<tr><td>8</td><td>Geofence 9</td></tr>
<tr><td>9</td><td>Geofence 10</td></tr>
<tr><td>10..15</td><td>reserve</td></tr>
</table>
0 - device outside the geofence;
1 - device inside the geofence.</td></tr>
</table>

<table>
<tr><td>201</td><td>CAN. Security status flags</td><td>2</td><td>U16</td><td>Status flags.
Bit field:<table><tr><th>Bits</th><th>Value</th></tr><tr><td>0</td><td>Ignition on</td></tr><tr><td>1</td><td>Alarm system is switched to security mode (in alarm mode)</td></tr><tr><td>2</td><td>Car is locked using a key fob</td></tr><tr><td>3</td><td>Key is in the ignition</td></tr><tr><td>4</td><td>Dynamic ignition 2 enabled</td></tr><tr><td>5</td><td>Front passenger door opened</td></tr><tr><td>6</td><td>Rear passenger doors open</td></tr><tr><td>7</td><td>Reserved</td></tr><tr><td>8</td><td>Driver’s door open</td></tr><tr><td>9</td><td>Passenger doors open</td></tr><tr><td>10</td><td>Trunk open</td></tr><tr><td>11</td><td>Hood open</td></tr><tr><td>12</td><td>Tightened hand brake lever</td></tr><tr><td>13</td><td>Foot brake pressed</td></tr><tr><td>14</td><td>Engine is running</td></tr><tr><td>15</td><td>Webasto</td></tr></table></td></tr>
<tr><td>202</td><td>CAN. Security status events</td><td>1</td><td>U8</td><td>Last event:
0 - No event;
1 - Car is locked using a key fob;
2 - Car is opened using a key fob;
3 - Trunk is opened using a key fob;
4 - Module sent a reset signal to the alarm
5 - Reserved
6 - Reserved
7 - Module switched to "sleep mode" energy saving mode
8... 255 - Reserved</td></tr>
</table>

<table>
<tr><td>203</td><td>CAN. Emergency controllers</td><td>4</td><td>U32</td><td>Bit field:<table><tr><th>Bit</th><th>Value</th></tr><tr><td>0</td><td>STOP</td></tr><tr><td>1</td><td>Oil pressure/level</td></tr><tr><td>2</td><td>Refrigerant temperature/level</td></tr><tr><td>3</td><td>Hand brake system</td></tr><tr><td>4</td><td>Battery charging</td></tr><tr><td>5</td><td>AIRBAG (airbag)</td></tr><tr><td>6</td><td>Reserved</td></tr><tr><td>7</td><td>Reserved</td></tr><tr><td>8</td><td>Check the engine</td></tr><tr><td>9</td><td>Lighting malfunction</td></tr><tr><td>10</td><td>Low air pressure in the tire</td></tr><tr><td>11</td><td>Worn brake pads</td></tr><tr><td>12</td><td>Prevention</td></tr><tr><td>13</td><td>ABS (anti-block system)</td></tr><tr><td>14</td><td>Low fuel level</td></tr><tr><td>15</td><td>Approaching Service</td></tr><tr><td>16</td><td>ESP (electronic stability program)</td></tr><tr><td>17</td><td>Glow plug indicator</td></tr><tr><td>18</td><td>FAP (particulate filter)</td></tr><tr><td>19</td><td>EPC (electrical pressure control)</td></tr><tr><td>20</td><td>Overall lights</td></tr><tr><td>21</td><td>Dipped headlights</td></tr><tr><td>22</td><td>High beam headlights</td></tr><tr><td>23</td><td>Reserved</td></tr><tr><td>24</td><td>Ready to start moving</td></tr><tr><td>25</td><td>Cruise control</td></tr><tr><td>26</td><td>Retarder automatic</td></tr><tr><td>27</td><td>Hand retarder</td></tr><tr><td>28</td><td>Air conditioner on</td></tr><tr><td>29</td><td>Reserved</td></tr><tr><td>30</td><td>Driver's belt</td></tr><tr><td>31</td><td>Passenger belt</td></tr></table></td></tr>
<tr><td rowspan="2">204</td><td rowspan="2">CAN. Fault information</td><td rowspan="2">5</td><td>U8</td><td>Status of fault indication lamps.
Bit field:<table><tr><th>Bit</th><th>Value</th></tr><tr><td>0</td><td>Fault indication lamp is on</td></tr><tr><td>1</td><td>Red "Stop" lamp is on</td></tr><tr><td>2</td><td>Yellow "Warning" lamp is on</td></tr><tr><td>3</td><td>"Protection" lamp is on</td></tr><tr><td>4</td><td>Fault lamp flashes</td></tr><tr><td>5</td><td>Red "Stop" lamp flashes</td></tr><tr><td>6</td><td>"Warning" yellow lamp flashes</td></tr><tr><td>7</td><td>"Protection" lamp flashes</td></tr></table></td></tr>
<tr><td>U32</td><td>Diagnostics Trouble Code.
0x00000000…0xFFFFFFFF</td></tr>
<tr><td>205</td><td>Custom engine hours 1 (on-load operation)</td><td>4</td><td>U32</td><td>0-232 sec</td></tr>
<tr><td>206</td><td>Custom engine hours 2 (no load operation)</td><td>4</td><td>U32</td><td>0-232 sec</td></tr>
<tr><td>207</td><td>User parameter #1 1 byte</td><td>1</td><td>U8, I8</td><td rowspan="7">A universal field can be represented as:
1. Unsigned integer 0... 255
2. Signed integer -128... 127
3. Bit field with bits 0.. 7</td></tr>
<tr><td>208</td><td>User parameter #2 1 byte</td><td>1</td><td>U8, I8</td></tr>
<tr><td>209</td><td>User parameter #3 1 byte</td><td>1</td><td>U8, I8</td></tr>
<tr><td>210</td><td>User parameter #4 1 byte</td><td>1</td><td>U8, I8</td></tr>
<tr><td>211</td><td>User parameter #5 1 byte</td><td>1</td><td>U8, I8</td></tr>
<tr><td>212</td><td>User parameter #6 1 byte</td><td>1</td><td>U8, I8</td></tr>
<tr><td>213</td><td>User parameter #7 1 byte</td><td>1</td><td>U8, I8</td></tr>
</table>

214	User parameter #8 1 byte	1	U8, I8	
215	User parameter #9 1 byte	1	U8, I8	
216	User parameter #10 1 byte	1	U8, I8	
217	User parameter #11 1 byte	1	U8, I8	
218	User parameter #12 1 byte	1	U8, I8	
219	User parameter #13 1 Byte	1	U8, I8	
220	User parameter #14 1 Byte	1	U8, I8	
221	User parameter #15 1 byte	1	U8, I8	
222	User parameter #16 1 byte	1	U8, I8	
223	User parameter #1 2 bytes	2	U16, I16	Universal field can be represented as: 1. Unsigned integer 0... 65535; 2. Signed integer -32768... 32767; 3. Bit field with bits 0..15.
224	User parameter #2 2 Bytes	2	U16, I16	
225	User parameter #3 2 Bytes	2	U16, I16	
226	User parameter #4 2 Bytes	2	U16, I16	
227	User parameter #5 2 Bytes	2	U16, I16	
228	User parameter #6 2 Bytes	2	U16, I16	
229	User parameter #7 2 Bytes	2	U16, I16	
230	User parameter #8 2 Bytes	2	U16, I16	
231	User parameter #9 2 bytes	2	U16, I16	
232	User parameter #10 2 bytes	2	U16, I16	
233	User parameter #11 2 bytes	2	U16, I16	
234	User parameter #12 2 bytes	2	U16, I16	
235	User parameter #13 2 bytes	2	U16, I16	
236	User parameter #14 2 bytes	2	U16, I16	
237	User parameter #15 2 bytes	2	U16, I16	
238	User parameter #1 4 bytes	4	U32, I32, float	Universal field can be represented as: 1. Unsigned integer 0...4294967295 2. Signed integer -2147483648…2147483647; 3. Bit field with bits 0.. 31 4. Real number -3.40282347e+38…-1.17549435e-38, 0, 1.17549435E-38e …3.40282347e+38
239	User parameter #2 4 bytes	4	U32, I32, float	
240	User parameter #3 4 bytes	4	U32, I32, float	
241	User parameter #4 4 bytes	4	U32, I32, float	
242	User parameter #5 4 bytes	4	U32, I32, float	
243	User parameter #6 4 bytes	4	U32, I32, float	
244	User parameter #7 4 bytes	4	U32, I32, float	
245	User parameter #8 4 bytes	4	U32, I32, float	
246	User parameter #9 4 bytes	4	U32, I32, float	
247	User parameter #10 4 bytes	4	U32, I32, float	
248	User parameter #11 4 bytes	4	U32, I32, float	
249	User parameter #12 4 bytes	4	U32, I32, float	
250	User parameter #13 4 bytes	4	U32, I32, float	
251	User parameter #14 4 bytes	4	U32, I32, float	
252	User parameter #15 4 bytes	4	U32, I32, float	

253	User parameter #1 8 bytes	8	U64, I64, double	A universal field can be represented as: 1. Unsigned integer 0...18446744073709551615 2. Signed integer -9223372036854775808… 9223372036854775807; 3. Bit field with bits 0.. 63 4. Real number -1.7976931348623158e+308… -2.2250738585072014e-308, 0, 2.2250738585072014e-308e… 1.7976931348623158e+308
254	User parameter #2 8 bytes	8	U64, I64, double	
255	User parameter #3 8 bytes	8	U64, I64, double	

Annex A.2. Structure of additional FLEX 2.0 telemetry records

№	Record field	Record item size	Data format	Accepted value
1	Data length	2	U16	Total size of packet fields (numbered 2-15)
2	Data structure version	1	U8	Version of the data structure of the static part of the Packet to identify the compatibility of the format of the transmitted data on the server and in the device. For version 1.0 10 (0x0A)
3	Data length	1	U8	Length of the static part of the packet (fields 4-14)
4	End-to-end record number in the non-volatile memory	4	U32	Starts from zero, increments with each record. Never decreases.
5	Event code corresponding to this record	2	U16	Codes specified in the table are recorded in the protocol
6	Event time	4	U32	Number of seconds since 1970
7	Status of GPS/GLONASS navigation sensor	1	U8	Bit field: Bits: Values 0: 0 - navigation receiver is off; 1 - navigation receiver is on. 1: 0 - invalid navigation; 1 - valid navigation. 2..7: Number of navigation satellites 0-32
8	Time of the last valid coordinates (before the event)	4	U32	Number of seconds since 1970
9	Last valid latitude	4	I32	Latitude angle recorded when obtaining the last valid coordinates. In ten thousandths of a minute. For example 55° 42,2389' will be presented as 33422389
10	Last valid longitude	4	I32	Longitude angle recorded when obtaining the last valid coordinates. In ten thousandths of a minute. For example 37° 41,6063' will be presented as 22616063
11	Last valid altitude	4	I32	Altitude relative to sea level recorded when obtaining the last valid coordinates: In decimeters For example 205 meters will be submitted by any 2050 decimeters
12	Speed	4	Float	Speed recorded when obtaining the last valid coordinates. km/h
13	Course	2	U16	Course recorded when obtaining the last valid coordinates. 0° … 360°
14	Current mileage	4	Float	Mileage recorded at the time of the event, calculated during the arrival of valid navigation data. km.
15	Dynamic part of the packet	-	U8	Dataset depends on the event that occurred.

Structure of the dynamic part of the FLEX 2.0 additional packet

Dynamic part of the packet contains a set of fields of various formats and contents.

General field format

№	Record field	Record item size	Data format	Accepted value
1	Field type	1	U8	Variable that defines the format of the next field.
2	Field length	1	U8	Field length
3	Field contents	-	-	Data dependent on field type and size specified by field length

List of possible field types for the dynamic part of the packet

0x01 - Touch Memory key.

№	Record field	Record item size	Data format	Accepted value
3	Touch Memory Key	8	U64	Touch Memory key code read by the device at the time of the event.

0x02 - driver card 1,2 (tachograph). Card number depends on the event code.

№	Record field	Record item size	Data format	Accepted value
3	Driver card number	16	U8[16]	Card identification number for the tachograph.

Event codes for which this packet is generated allow you to determine which card was installed.

Code (in hexadecimal)	Code (in decimal)	Interpretation
2530	9520	Installing the card in slot #1
2531	9521	Installing the card in slot #2

0x03 - RFID radio tag code.

№	Record field	Record item size	Data format	Accepted value
3	RFID tag code	8	U64	RFID tag code read by the device at the time of the event.

0x04 ... 0xFF - Reserved for further expansion.

Annex A.3. Examples of basic FLEX packets

FLEX 2.0 Protocol negotiation packet example

Byte number	Byte	Value
1	0x40	@ NTC – preamble.
2	0x4e	
3	0x54	
4	0x43	
5	0x01	1 - recipient ID.
6	0x00	
7	0x00	
8	0x00	
9	0x00	0 - sender ID.
10	0x00	
11	0x00	
12	0x00	
13	0x1a	26 - number of bytes of data in the packet.
14	0x00	
15	0x1b	Checksum of data.
16	0x19	Header checksum.
17	0x2a	*>FLEX
18	0x3e	
19	0x46	
20	0x4c	
21	0x45	
22	0x58	
23	0xb0	Symbol of FLEX protocol.
24	0x14	2.0 - protocol version.
25	0x14	2.0 - data structure version.
26	0x7a	122 - configuration field size.
27	0xf2	Bit array with information about the transmitted fields of the data structure. In this case, the following fields are selected: - number of record in non-volatile memory; - event code corresponding to this record; - event time; - device status; - GSM layer; - voltage at the main power supply.
28	0x00	
29	0x20	
30	0x00	
31	0x00	
32	0x00	
33	0x00	
34	0x00	
35	0x00	
36	0x00	
37	0x00	
38	0x00	
39	0x00	
40	0x00	
41	0x00	
42	0x00	

FLEX 2.0 telemetry packet example

Byte number	Byte	Value
1	0x7e	~A
2	0x41	
3	0x01	1 – telemetry record in a packet.
4	0x1e	30 - number of the record in the non-volatile memory.
5	0x00	
6	0x00	
7	0x00	
8	0xb0	6064 - code of the incident.
9	0x17	
10	0xe1	1450450401 - event time in UTC (18.12.2015 @ 17:53:21 MSK GTM + 03:00)
11	0x1d	
12	0x74	
13	0x56	
14	0x00	Device status.
15	0x1f	31 - GSM level.
16	0x14	12052 - voltage at the main power supply.
17	0x2f	
18	0x24	8-bit byte CRC8 of the packet.

FLEX 2.0 additional telemetry packet example

Byte number	Byte	Value
1	0x7e	~E
2	0x45	
3	0x01	1 - one packet in the archive message.
4	0x39	Total size of the first packet - 57 bytes.
5	0x00	
6	0x0a	1.0 - version of the static part of the packet.
7	0x25	Length of the static part of the packet - 37 bytes.
8	0x10	End-to-end record number - 16.
9	0x00	
10	0x00	
11	0x00	
12	0x30	Event code - 2530 (installing card in slot #1).
13	0x25	
14	0x05	Event time.
15	0x37	
16	0xd7	
17	0x55	
18	0x33	Status of the navigation sensor.
19	0xec	Time of the last valid coordinates.
20	0x36	
21	0xd7	
22	0x55	
23	0x7e	Last valid latitude.
24	0x2d	
25	0xf9	

26	0x01	
27	0x3b	Last valid longitude.
28	0xcc	
29	0x14	
30	0x01	
31	0x0f	Last valid altitude.
32	0x07	
33	0x00	
34	0x00	
35	0x00	Speed.
36	0x00	
37	0x00	
38	0x00	
39	0x00	Course.
40	0x00	
41	0x00	Current mileage.
42	0x00	
43	0x00	
44	0x00	
45	0x02	Subpacket of type 2 - driver's card.
46	0x10	Length of the subpacket - 16 bytes.
47	0x12	Driver card number.
48	0x00	
49	0x00	
50	0x00	
51	0x00	
52	0x00	
53	0x00	
54	0x00	
55	0x00	
56	0x00	
57	0x00	
58	0x00	
59	0x00	
60	0x00	
61	0x00	
62	0x34	
63	0x9a	8-bit byte CRC8 of the packet.

Annex A.4. Compressor configurations and conditions for ThermoKing units

Compressor configurations for ThermoKing units

Value	Configuration
129	"Diesel, Cycle-Sentry" (Standard RA Diesel Cycle-Sentry Fresh Recip)
130	"Diesel, Cycle-Sentry" (Standard RA Diesel Cycle-Sentry Frozen Recip)
131	"Diesel, Continuous" (Standard RA Diesel Continuous Fresh Recip)
132	"Diesel, Continuous" (Standard RA Diesel Continuous Frozen Recip)
133	"Electric, Cycle-Sentry" (Standard RA Electric Cycle-Sentry Fresh Recip)
134	"Electric, Cycle-Sentry" (Standard RA Electric Cycle-Sentry Frozen Recip)
135	"Electric, Continuous" (Standard RA Electric Continuous Fresh Recip)
136	"Electric, Continuous" (Standard RA Electric Continuous Frozen Recip)
137	"Diesel, Cycle-Sentry" (Standard DA Diesel Cycle-Sentry Fresh Recip)
138	"Diesel, Cycle-Sentry" (Standard DA Diesel Cycle-Sentry Frozen Recip)
139	"Diesel, Continuous" (Standard DA Diesel Continuous Fresh Recip)
140	"Diesel, Continuous" (Standard DA Diesel Continuous Frozen Recip)
141	"Electric, Cycle-Sentry" (Standard DA Electric Cycle-Sentry Fresh Recip)
142	"Electric, Cycle-Sentry" (Standard DA Electric Cycle-Sentry Frozen Recip)
143	"Electric, Continuous" (Standard DA Electric Continuous Fresh Recip)
144	"Electric, Continuous" (Standard DA Electric Continuous Frozen Recip)
145	"Diesel, Cycle-Sentry" (Economy RA Diesel Cycle-Sentry Fresh Recip)
146	"Diesel, Cycle-Sentry" (Economy RA Diesel Cycle-Sentry Frozen Recip)
147	"Diesel, Continuous" (Economy RA Diesel Continuous Fresh Recip)
148	"Diesel, Continuous" (Economy RA Diesel Continuous Frozen Recip)
149	"Electric, Cycle-Sentry" (Economy RA Electric Cycle-Sentry Fresh Recip)
150	"Electric, Cycle-Sentry" (Economy RA Electric Cycle-Sentry Frozen Recip)
151	"Electric, Continuous" (Economy RA Electric Continuous Fresh Recip)
152	"Electric, Continuous" (Economy RA Electric Continuous Frozen Recip)
153	"Diesel, Continuous" (Standard RA Diesel Modulation Recip)
154	"Diesel, Continuous" (Standard DA Diesel Modulation Recip)
155	"Electric, Continuous" (Standard RA Electric Modulation Recip)
156	"Electric, Continuous" (Standard DA Electric Modulation Recip)

States for ThermoKing units

Value	State
129	"INVAL" (Invalid)
130	"COOL " (Cool)
131	"NULL " (Normal Null)
132	"HEAT " (Heat)
133	"DFRST" (Defrost)
134	"SHTDN" (Shutdown)
135	"LKCOL" (Lockout Cool)
136	"FACOL" (Frozen Algorithm Cool)
137	"FANUL" (Frozen Algorithm Null)

138	"PULUP" (Pull Up)
139	"PULDN" (Pull Down)
140	"STABL" (Stabilize)
141	"PMPDN" (Pump Down)
142	"OVLSH" (Overload Shutdown)
143	"DSABL" (Disable Controller)
144	"LPTRP" (Long Pretrip)
145	"SPTRP" (Short Pretrip)
146	"NORML" (Normal)
147	"PWROF" (Power Off)
148	"WSTRT" (Warm Restart)
149	"RACTL" (Return Air Control)
150	"DACTL" (Discharge Air Control)
151	"FACTL" (Frozen Algorithm Control)
152	"DSANP" (Disable Controller No Protect)
153	"FACHI" (Frozen Algorithm Cool High)
154	"FACLO" (Frozen Algorithm Cool Low)
155	"COOLH" (Cool High)
156	"DSCHK" (Dual Scroll Compressor Check)
157	"RRACL" (Remote Return Air Control)
158	"OTCTL" (Evaporator Control)
159	"SLEEP" (Sleep Mode)
160	"EVAC" (Evacuation Mode)
161	"SERVICE" (Service Test Mode)
162	"RLYBD" (Relay Board Test Mode)
163	"TEMP CTL" (Temperature Control)
164	"ENGSTRT" (Diesel Engine Start)
165	"MOTSTRT" (Electric Motor Start)
166	"EM FLASH" (EM Flash Load)
167	"NO ZONE" (Zone Not Configured)
168	Countdown Logging Mode
169	Conservative Logging Mode
170	Profile Upload Mode

Annex B. Checksum calculation algorithm CRC8

To check the integrity of data in FLEX messages, a checksum calculated using the CRC8 algorithm is used. The checksum is counted from the beginning of the packet (including the '~' character) to the last byte of data and is usually written at the end of the packet. You can use the table function (programming language C) to calculate the CRC8.

```
const unsigned char crc8_table[256] =
{
    0x00, 0x31, 0x62, 0x53, 0xC4, 0xF5, 0xA6, 0x97, 0xB9, 0x88, 0xDB, 0xEA, 0x7D, 0x4C, 0x1F, 0x2E,
    0x43, 0x72, 0x21, 0x10, 0x87, 0xB6, 0xE5, 0xD4, 0xFA, 0xCB, 0x98, 0xA9, 0x3E, 0x0F, 0x5C, 0x6D,
    0x86, 0xB7, 0xE4, 0xD5, 0x42, 0x73, 0x20, 0x11, 0x3F, 0x0E, 0x5D, 0x6C, 0xFB, 0xCA, 0x99, 0xA8,
    0xC5, 0xF4, 0xA7, 0x96, 0x01, 0x30, 0x63, 0x52, 0x7C, 0x4D, 0x1E, 0x2F, 0xB8, 0x89, 0xDA, 0xEB,
    0x3D, 0x0C, 0x5F, 0x6E, 0xF9, 0xC8, 0x9B, 0xAA, 0x84, 0xB5, 0xE6, 0xD7, 0x40, 0x71, 0x22, 0x13,
    0x7E, 0x4F, 0x1C, 0x2D, 0xBA, 0x8B, 0xD8, 0xE9, 0xC7, 0xF6, 0xA5, 0x94, 0x03, 0x32, 0x61, 0x50,
    0xBB, 0x8A, 0xD9, 0xE8, 0x7F, 0x4E, 0x1D, 0x2C, 0x02, 0x33, 0x60, 0x51, 0xC6, 0xF7, 0xA4, 0x95,
    0xF8, 0xC9, 0x9A, 0xAB, 0x3C, 0x0D, 0x5E, 0x6F, 0x41, 0x70, 0x23, 0x12, 0x85, 0xB4, 0xE7, 0xD6,
    0x7A, 0x4B, 0x18, 0x29, 0xBE, 0x8F, 0xDC, 0xED, 0xC3, 0xF2, 0xA1, 0x90, 0x07, 0x36, 0x65, 0x54,
    0x39, 0x08, 0x5B, 0x6A, 0xFD, 0xCC, 0x9F, 0xAE, 0x80, 0xB1, 0xE2, 0xD3, 0x44, 0x75, 0x26, 0x17,
    0xFC, 0xCD, 0x9E, 0xAF, 0x38, 0x09, 0x5A, 0x6B, 0x45, 0x74, 0x27, 0x16, 0x81, 0xB0, 0xE3, 0xD2,
    0xBF, 0x8E, 0xDD, 0xEC, 0x7B, 0x4A, 0x19, 0x28, 0x06, 0x37, 0x64, 0x55, 0xC2, 0xF3, 0xA0, 0x91,
    0x47, 0x76, 0x25, 0x14, 0x83, 0xB2, 0xE1, 0xD0, 0xFE, 0xCF, 0x9C, 0xAD, 0x3A, 0x0B, 0x58, 0x69,
    0x04, 0x35, 0x66, 0x57, 0xC0, 0xF1, 0xA2, 0x93, 0xBD, 0x8C, 0xDF, 0xEE, 0x79, 0x48, 0x1B, 0x2A,
    0xC1, 0xF0, 0xA3, 0x92, 0x05, 0x34, 0x67, 0x56, 0x78, 0x49, 0x1A, 0x2B, 0xBC, 0x8D, 0xDE, 0xEF,
    0x82, 0xB3, 0xE0, 0xD1, 0x46, 0x77, 0x24, 0x15, 0x3B, 0x0A, 0x59, 0x68, 0xFF, 0xCE, 0x9D, 0xAC
};

unsigned char crc8_calc
(
    unsigned char   *lp_block,    / * (in) pointer to the data buffer */
    unsigned int    len           / * (in) number of bytes to count */
)
{
    unsigned char crc = 0xFF;
    while (len--)
    {
        crc = crc8_table[crc ^ *lp_block++];
    }
    return crc;
}
```

Or use the following function (java language):

```
public static byte crc8 (byte[] buffer)
{
     byte crc = (byte) 0xFF;
     for (byte b : buffer) {
          crc ^= b;
          for (int i = 0; i < 8; i++) {
              crc = (crc & 0x80) != 0 ? (byte) ((crc << 1) ^ 0x31) : (byte) (crc << 1);
         }
     }
     return crc;
}
```

Annex C. Checksum calculation algorithm CRC16

To check the integrity of data in encrypted messages, a checksum is used, calculated using the CRC16 algorithm with the polynomial 0x1021the initial value 0. The checksum is counted from the beginning of the packet (including the '#' character) to the last byte of data and is written at the end of the packet. You can use the table function (programming language C) to calculate the CRC16.

```
static const unsigned short crc16_poly1021_table[256] =
{
     0x0000, 0x1021, 0x2042, 0x3063, 0x4084, 0x50a5, 0x60c6, 0x70e7, 0x8108, 0x9129, 0xa14a, 0xb16b,
     0xc18c, 0xd1ad, 0xe1ce, 0xf1ef, 0x1231, 0x0210, 0x3273, 0x2252, 0x52b5, 0x4294, 0x72f7, 0x62d6,
     0x9339, 0x8318, 0xb37b, 0xa35a, 0xd3bd, 0xc39c, 0xf3ff, 0xe3de, 0x2462, 0x3443, 0x0420, 0x1401,
     0x64e6, 0x74c7, 0x44a4, 0x5485, 0xa56a, 0xb54b, 0x8528, 0x9509, 0xe5ee, 0xf5cf, 0xc5ac, 0xd58d,
     0x3653, 0x2672, 0x1611, 0x0630, 0x76d7, 0x66f6, 0x5695, 0x46b4, 0xb75b, 0xa77a, 0x9719, 0x8738,
     0xf7df, 0xe7fe, 0xd79d, 0xc7bc, 0x48c4, 0x58e5, 0x6886, 0x78a7, 0x0840, 0x1861, 0x2802, 0x3823,
     0xc9cc, 0xd9ed, 0xe98e, 0xf9af, 0x8948, 0x9969, 0xa90a, 0xb92b, 0x5af5, 0x4ad4, 0x7ab7, 0x6a96,
     0x1a71, 0x0a50, 0x3a33, 0x2a12, 0xdbfd, 0xcbdc, 0xfbbf, 0xeb9e, 0x9b79, 0x8b58, 0xbb3b, 0xab1a,
     0x6ca6, 0x7c87, 0x4ce4, 0x5cc5, 0x2c22, 0x3c03, 0x0c60, 0x1c41, 0xedae, 0xfd8f, 0xcdec, 0xddcd,
     0xad2a, 0xbd0b, 0x8d68, 0x9d49, 0x7e97, 0x6eb6, 0x5ed5, 0x4ef4, 0x3e13, 0x2e32, 0x1e51, 0x0e70,
     0xff9f, 0xefbe, 0xdfdd, 0xcffc, 0xbf1b, 0xaf3a, 0x9f59, 0x8f78, 0x9188, 0x81a9, 0xb1ca, 0xa1eb,
     0xd10c, 0xc12d, 0xf14e, 0xe16f, 0x1080, 0x00a1, 0x30c2, 0x20e3, 0x5004, 0x4025, 0x7046, 0x6067,
     0x83b9, 0x9398, 0xa3fb, 0xb3da, 0xc33d, 0xd31c, 0xe37f, 0xf35e, 0x02b1, 0x1290, 0x22f3, 0x32d2,
     0x4235, 0x5214, 0x6277, 0x7256, 0xb5ea, 0xa5cb, 0x95a8, 0x8589, 0xf56e, 0xe54f, 0xd52c, 0xc50d,
     0x34e2, 0x24c3, 0x14a0, 0x0481, 0x7466, 0x6447, 0x5424, 0x4405, 0xa7db, 0xb7fa, 0x8799, 0x97b8,
     0xe75f, 0xf77e, 0xc71d, 0xd73c, 0x26d3, 0x36f2, 0x0691, 0x16b0, 0x6657, 0x7676, 0x4615, 0x5634,
     0xd94c, 0xc96d, 0xf90e, 0xe92f, 0x99c8, 0x89e9, 0xb98a, 0xa9ab, 0x5844, 0x4865, 0x7806, 0x6827,
     0x18c0, 0x08e1, 0x3882, 0x28a3, 0xcb7d, 0xdb5c, 0xeb3f, 0xfb1e, 0x8bf9, 0x9bd8, 0xabbb, 0xbb9a,
     0x4a75, 0x5a54, 0x6a37, 0x7a16, 0x0af1, 0x1ad0, 0x2ab3, 0x3a92, 0xfd2e, 0xed0f, 0xdd6c, 0xcd4d,
     0xbdaa, 0xad8b, 0x9de8, 0x8dc9, 0x7c26, 0x6c07, 0x5c64, 0x4c45, 0x3ca2, 0x2c83, 0x1ce0, 0x0cc1,
     0xef1f, 0xff3e, 0xcf5d, 0xdf7c, 0xaf9b, 0xbfba, 0x8fd9, 0x9ff8, 0x6e17, 0x7e36, 0x4e55, 0x5e74,
     0x2e93, 0x3eb2, 0x0ed1, 0x1ef0
};

unsigned short eval_crc16(unsigned short crc, const unsigned char *msg, unsigned int msg_len)
{
   /* Calculation of 16-bit cyclic redundancy code (CRC) by the table method
    * DESCRIPTION
    * Calculates 16-bit cyclic redundancy using a table.
    * PARAMETERS
    * crc - initial CRC value, 0 on the first call;
    * msg - indicator of the message block;
    * msg_len - number of bytes in the block.
    * RETURN VALUE
    * 16-bit CRC.
    */

     unsigned char temp;

     while (msg_len--)
     {
          temp = (*msg++ ^ (crc >> 8)) & 0xFF;
          crc = crc16_poly1021_table[temp] ^ (crc << 8);
     }
     return (crc);
}
```

Annex D. Summary of NTCB messages

Packet	Appointment
System	
*?V	Query model and version of the device.
*?VGPS	Query information about the firmware version of the GPS receiver
*?S	Query unique device ID.
*>S	Message with individual device ID string.
*!DEV_RESET	Device restart command.
*?USSD	Arbitrary USSD request
*!BEEP	Command of arbitrary sound indication by buzzer
*!CHNGSIM	Command to change SIM card
*?ICCID	Query unique SIM serial number
*!O	Command for microphone listening
*!SETTIME	Command for time adjustment on device
*?DATA	Command for diagnostic information query
*Z	Response to unsupported messages
~Z	Response to unsupported messages
Outputs	
*!1Y	Command to activate output 1
*!1N	Command to disactivate output 1
*!2Y	Command to activate output 2
*!2N	Command to disactivate output 2
*!3Y	Command to activate output 3
*!3N	Command to disactivate output 3
*!4Y	Command to activate output 4
*!4N	Command to disactivate output 4
*!SETOUT	Command to change output state
Inputs	
*!OFF	Command to lock input
*!ON	Command to unlock input
Telemetry	
~A	Array of telemetry messages in FLEX format.
~T	Structure of out-of-order message sending packet in FLEX format.
~C	Structure of the current state sending packet in FLEX format.
~E	Array of additional FLEX 2.0 telemetry messages.
~X	Structure of additional FLEX 2.0 telemetry messages sending packet.
*?L	Query the nearest telemetry record before the specified date
*?R	Query the nearest telemetry record after the specified date
*?I	Query telemetry record by its index
*?A	Query current state of the device in the main packet.
*?E	Query current state of the device in the additional packet.
*?ES	Query device status with SMS sending

*!SYNC	Command to confirm synchronization of internal memory with the server
*!REP_FL	Command to resend telemetry from non-volatile memory
*!REP_SD	Command to resend telemetry from SD card
Operating modes of the device	
*!GY	Command to enable security mode
*!GN	Command to disable security mode
*!M	Command to change device operating mode
*?M	Query current operating mode
RCS, RFU, DRC Services	
*!CNCT_RCS	Command to connect to the configurator via RCS
*!CNCT_RFU	Command to update device firmware via RFU
*!CNCT_DRC	Command to connect to DRC
Touch Memory keys	
*>TMKEY	NTCB packet structure of sending unregistered Touch Memory key
*?TM	Query code of the last Touch Memory key read by the device
*?ERFT	Query the last active radio tag
*!SETTM	Command to edit the list of registered Touch Memory keys
~O	Module ID: 0x7D Command code: 4 - command to edit the list of registered Touch Memory keys
Tachograph (Module code: 0x81)	
~Q	Request ID: 0 - query information on the current state of the tachograph; 2 - query information on card No. 1 3 - query information on card No. 2 4 - query vehicle registration information.
~O	Command code: 0 - authorization of the device in the tachograph; 1 - installation of a new authorization key; 2 - generation of the upload file; 3 - generation of the upload file with sending to email.
~G	Request ID: 0 – request of DDD file block.
Driver display (Module code: 0x82)	
~O	Command code: 0 - sending a message to the driver.
*!DV	Command to send a message to the driver display.
Autoinformer	
*!AINF	Command code: 0x01 - change of the current route; 0x02 - starts playback of the sound file; 0x03 - change of the current movement mode.
*!AINF:	Command to change the route
*!AINF!	Command to playback sound file
*!AINF#	Command to change speed mode
*&AINF	Autoinformer event alerts
*!AINF ROUTE	Command to change the route

*!AINF PLAY	Command to playback sound file
*!AINF MODE	Command to change current mode
*!AINF STOP	Command to change current stop
Camera (Module Code: 0x80)	
~O	Command code: 0 - automatic shooting control; 1 - take a photo; 2 - take a photo and then send it to email.
~Q	Request ID: 0 - obtaining information about camera; for photo information.
~N	Alert Code: 0 - notification of the creation of a new photo.
~G	Request ID: 0 - query overview photo data; 1 - query the main photo data.
*!DCAM	Text analogue of camera commands
Communication between external interfaces and the server (Module ID: 0x7F)	
*!U2S, *>U2S	Command to transfer data via USB device to the server.
*!S2U, *>S2U	Command to transfer from the server via USB.
~P	Message ID: 0x00 - data transfer in transparent mode from USB; 0x01 - data transfer in transparent mode from RS232; 0x02 - data transfer in transparent mode from RS485.
*!UC	User command for transfering data to interfaces
CAN-LOG	
*!CANLOG	Setting CAN-LOG device program number.
*?CANLOG	Query CAN-LOG device program number.
Working with built-in accelerometer	
*!ACL_C	Accelerometer calibration command
Accident detection	
*?KRAI	Query information about detected accident
*!KRAI	Command to reset overwrite protection
Working with temperature sensors 1-Wire	
*?TM TSV	Query all visible temperature sensors
*?TM TSU	Query all visible undefined temperature sensors
*!TM	Command for temperature sensors
Working with passenger flow counters	
*!ACOND	Command for "Autoconductor" counter
Working with refrigeration units (module code: 7A)	
~Q	Request ID: 0 – query information about IQFreeze.
Device configuration commands	
*!EDITS	Command to write settings to the device with saving and following reboot
*!EDIT	Command to write settings to the device without saving
*!READ	Command to read setting from device

*!SETFM	FLEX bit mask edit command
Encryption	
*>CODE	Command to switch to encryption mode
#	Container package for transferring encrypted data

Annex E. Table of digital FLS error codes

Error code	Appointment
SIGNAL/SMART devices errors	
65535 (or -1)	Invalid response from sensor or incorrect response prefix (general error code)
65534 (or -2)	Invalid command code in response
65533 (or -3)	Invalid sensor address in response
65532 (or -4)	CRC Error
65531 (or -5)	Sensor not initialized
65530 (or -6)	Line break, no response packet or it is incomplete
FLS Technoton errors	
65529 (or -7)	Calibration error (Technoton)
65528 (or -8)	Hardware Error (Technoton)
Ultrasonic FLS TC SENSOR" UZI-0.8 and UZI-2.5 errors	
65527 (or -9)	Cable break from sensor
65526 (or -10)	No signal
65525 (or -11)	Low battery power
65524 (or -12)	Low battery charge + cable break from sensor
65523 (or -13)	Low battery + no signal
LLS FLS errors	
65522 (or -14)	Incorrect value of fuel level (level value greater than 4095)