NAVTELECOM

TELEMATICS SYSTEMS

Protocol of information exchange for Navtelecom equipment

6.0 Version

Moscow
2019

Content

History of changes

In the version 5.0:

- the description of the dynamic change of the data transmission mask by the server for FLEX was deleted;
- description of transfer of only changed parameters for the FLEX was deleted;
- description of FLEX 2.0 is added;
-
- description of work with auto-informer is added;
- description of the work with the display driver is added;
- description of the work with the tachograph is added;
- description of working with a digital camera is added;
- description of SMS device configuration is added.

In the version 5.1:

- error in the description of the "GPS / GLONASS navigation sensor status" field is fixed.

In the version 5.2:

- errors in the description of commands, requests and answers when operating with tachygraphers are fixed;
- errors in the description of commands, requests and responses when operating with cameras are fixed;
- structure of telemetric records FLEX format are updated (Appendix A.1).

In the version 5.3:

- errors in the description of the structure of telemetry records FLEX format (Appendix A.1), field №: 4, 5, 6, 53, 66, 68, 72, 122;
- types of variables in the description are given in the standard names;
- order of the fields in the table "Example of an additional telemetry package of the FLEX 2.0 protocol" is fixed;
- description of NTCB and NTCN command "Command to re-send telemetry from the black box" is added;
- description of NTCB and NTCN command "Command to re-send telemetry from SD-card" is added.

In the version 5.4:

- description of the structure of telemetry records FLEX format (Appendix A.1), field №: 4, 6;
- clarifications to the commands "Command to re-send telemetry from the black box", "Command to resend telemetry from SD-card" were made;
- SMS command to confirm black box synchronization with the server is added;
- "Standard SMS format M: 101" is added;
- arbitrary USSD NTCT and NTCB requests are added;
- sections 2.14, 4.2.12 "Working with a built-in accelerometer" are added;
- section 2.15 "Traffic accident fixation" is added;
- SMS command "SETTM" format is changed;
- description of the SMS request "POS" is added;
- formats of SMS commands and "SET" and "GET" requests are changed;
- request "*? ERFT" is added;
- section "5. Tone management" is added;

- "The format of the file that stores information about the accident" is added in the section "2.15 Accident fixation"
- "Appendix E. Table of error codes of digital FLS" is added;
- "Random buzzer indication command" is added in the section "2.1 System commands, requests, messages";
- "Format of a standard SMS-message M: 112" is added in the section "4.1. Formats of SMS-messages coming from the device";
- "Request for current data on the nearest stations of the cellular operator" is added in the section "4.2.2. Telemetry information";
- "Format of a standard SMS-message M: 113" is added in the section "4.1. Formats of SMS-messages coming from the device"
- "Request information about the firmware version of the GPS receiver" is added in the section "4.2.1. System requires and commands";
- "Appendix E. NTCB short list of messages" is added.

In the version 5.5:

- Field "Functional modules 2 status" is updated in the section "Appendix A.1. Structure of telemetry records FLEX format";
- new field type is added in the section "Structure of the dynamic part of the FLEX 2.0 additional package";
- "SIM card change command" is added in the section "2.1. System commands, requests, messages";
- "SIM card change command" is added in the section "4.2.1. System requests and commands";
- User command RS485 / RS232" is added in the section "2.12. Data exchange between external interfaces and server";
- "4.2.13. Data exchange between external interfaces" section is added;
- "Request for device status with sending to SMS" is added in the section "2.1. System commands, requests, messages";
- "Format of a standard SMS-message M: 114" is added in the section "4.1. Formats of SMS-messages coming from the device";
- "Device Status Request" is added in the section "4.2.1. System requests and commands".
- "Request a unique serial number of the SIM card" is added in the section "2.1. System commands, requests, messages".
- "Request a unique serial number of the SIM card" is added in the section "4.2.1. System requests and commands";
- New error code is added in the "Appendix E. Digital FLS error code table".

In the version 5.6:

- Request information about the firmware version of the GPS receiver" is added in the section "2.1. System commands, requests, messages";
- "Appendix E. NTCB short list of messages" is updated.

In the version 5.7:

- User command RS485 / RS232" renamed to "User command";
- for NTCB and NTCT commands "*! UC" interface "GPS" is added.

In the version 5.8:

- the special value 0xFFFFFFFF was removed from the description of the DDD file block request in the section "2.8.2. Formation and transfer of the DDD file to the server;
- he special value 0xFFFF was removed from the description of the command "Request snapshot data" and the maximum block size is specified in the section "2.11.2. Transferring images to the server";

- “Command to reset the current calibration” is added in the section “4.2.12. Work with a built-in accelerometer”;
- errors in the description of the parameter №77 of the FLEX structure is corrected
- he description of the parameter No. 108 of the FLEX structure is changed.

In the version 6.0:

- In section “1.2.1. The basic procedure for establishing a connection with the server using the FLEX protocol” was complemented by a table describing the package of the protocol version negotiation;
- In section “1.2.1. The basic procedure for establishing a connection with the server using the FLEX protocol” was complemented by the description of the current state package;
- In the section “1.2.2. List of versions and main FLEX messages” were complemented the following tables: “List of supported versions of the FLEX protocol” and “List of supported versions of the FLEX data structure”;
- In section “2.1. System commands, requests, messages” formats of the following command have been changed:
  “Free USSD request”,
  “Command of arbitrary sound indication by the buzzer”,
  “Command to change SIM-card”
  “Request the device status with SMS-transferring”;
  “Request a unique serial number of the SIM card”;
  “Request information about the firmware version of the GPS receiver”;
- Section “2.1. System Commands, Requests, Messages was complemented by description of the “Mic listening command” format;
- Section “2.2. Management of output lines” was complemented by description of the “Command for changing the state of the output line” format;
- Section “2.3. Output line control” was implemented by description of the following formats:
  "Command to block the input line",
  "Command unlock input line"
- In section “2.4. Telemetry commands, requests, messages” command formats have been changed:
  "Command to confirm the synchronization of the black box with the server",
  "Command to resend telemetry from the black box",
  “Command to resend telemetry from SD card”;
- In section “2.5. Management of device operation modes formats of the following commands were added:
  “Command to change the mode of the device”;
  “Request current mode of operation”;
- In section “2.6. Connection to RCS, RFU services” the following commands formats were added:
  “Command to connect to the configurator through the RCS service";
  "Command to connect to the RFU service to update the device firmware";
- In section “2.12. Data exchange between external interfaces and server” the format of the “User command” was changed;
- Description of the command “Request a program number of a CAN-LOG device” was added in the section “2.13. Work with CAN-LOG module”;
- “Accelerometer Calibration Command” was added in the section “2.14 Operation with a built-in accelerometer”;
- In the section “2.15. Fixation of traffic accident" description of the traffic accident file format in the “Format of the file that stores information about the accident” was changed;
- in the section "4.2.1 System requires and commands" changes were made in the following tables:
  "Requests",

"Commands";

- In section "4.2.2. Telemetry information" changes in the following tables were made:
  "Requests",
  "Command to resend telemetry from the black box",
  "Command to resend telemetry from SD card";
- In the section "4.2.3. Output lines" changed were made in the "Commands" table;
- In section "4.2.5. Services RCS, RFU" a table with a description of the new command format were added;
- The section "4.2.6. Subscribers" was deleted, numbering of section 4.2, starting with 4.2.6 (inclusive), has been amended accordingly;
- n section "4.2.6. Modes of operation of the device" changes were made in the following table:
  "Commands",
  "Requests";
- In the section "4.2.11. Working with a built-in accelerometer" changes were made in the "Commands" table;
- In the section "4.2.12 Data exchange between external interfaces" changes were made in the "Commands" table;
- Description of the format of new commands for configuration has been added to the section "4.3 SMS configuration";
- In Appendix "A.1. Structure of telemetric recordings of the FLEX format "for the FLEX1 and FLEX2 protocols the following fields have been changed: 4, 6, 38-44, 78-93, 100, 109.
- Appendix "A.4. Compressor configurations and conditions for Thermo King installations" was added;
- Appendix "C. A short list of NTCB messages" was added.

Symbols and abbreviations

Abbreviations	Transcription
AES128 (Advanced Transcription Standard)	symmetric block Transcription algorithm, block size is 128 bits
ASCII (American standard code for information interchange)	American character encoding table
CAN (Controller Area Network)	robust vehicle bus standard designed to allow microcontrollers and devices to communicate with each other in applications without a host computer
COM (Communications Port)	RS-232 interface standart
GPRS (General Package Radio Servic)	General Package Radio Service
Handshake	package consisting of a message with a string of an individual device identifier
GPS (Global Positioning System)	a global navigation satellite system that provides measurement of distance, time and defining a location in the world coordinate system WGS 84
GSM (Global Systemfor Mobilecommunications)	global digital standard for Mobile Communications
ID (Identifier)	identifier
IP (Internet Protocol)	routable protocol network layer stack TCP / IP
IMEI (International Mobile Equipment Identity)	International Mobile Equipment Identity
NTCB (Navtelecom Binary)	Navtelecom Binary exchange protocol
NTCT (Navtelecom Text)	Navtelecom Text exchange protocol
RCS (Remote Configuration Service)	Remote Configuration Service
RFU (Remote Firmware Update)	Remote Firmware Update
RS-232 (Recommended Standard 232)	standard describing an interface for serial data transmission that supports asynchronous communication
RS-485 (Recommended Standard 485)	standard for data transmission over two-wire half-duplex multipoint serial symmetric communication channel
SMS (Short Messaging Service)	technology that allows reception and transmission of short text messages over mobile radio telephone networks
TCP (Transmission Control Protocol)	network data transfer protocol
TCP/IP (Transmission Control Protocol / Internet Protocol)	set of network communication protocols
UDP (User Datagram Protocol)	user datagram protocol
USB (Universal Serial Bus)	universal serial wired bus
UTC (Coordinated Universal Time)	coordinated Universal Time
WGS-84 (World Geodetic System 1984)	World Geodetic System 1984
XOR	Boolean function, exclusive OR
ГЛОНАСС	Russian Federation Global Navigation Satellite System.
ПЗ-90.11	State geocentric reference system "Parametry Zemli 1990 goda"
ПК	Personal computer
ПО	software
ТС	vehicle

Symbols

In this document the following symbols are used:

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

For all the specified integer types, the byte order is little-endian.

The following types of records can be used to designate an array:
1) 16 * U8 - an array of 16 unsigned integer bytes;
2) char [7] is an array of 7 characters.

In order to indicate a string of arbitrary size without a terminator null, the following type of record is used: char [].

Introduction

In telematics devices ("SIGNAL" and "SMART" series) manufactured by Navtelecom LLC devices for all variants of the transmission of telematics information via communication channel, two protocols are used:

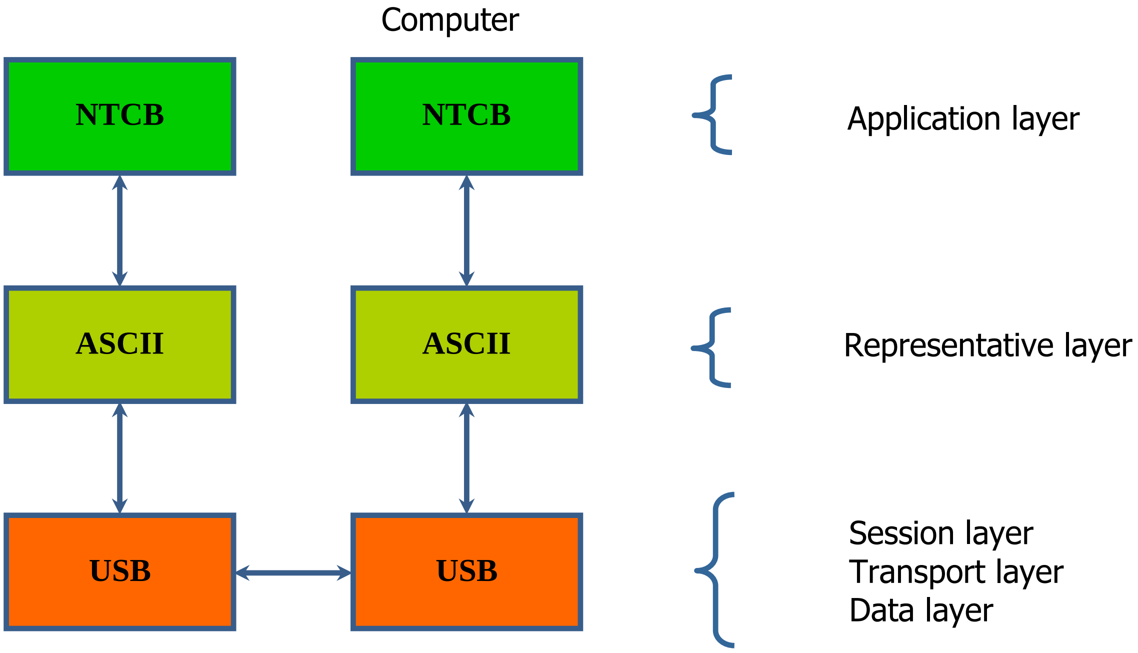
- binary protocol (NTCB) (Navtelecom Binary) with FLEX;
- text protocol (NTCT) (Navtelecom Text).

A text (symbolic) protocol is used for telemetry transmit via the SMS service of cellular operators. Packages of this protocol are limited by the length of one SMS-message (140 characters) and include all telemetric information about object of control.

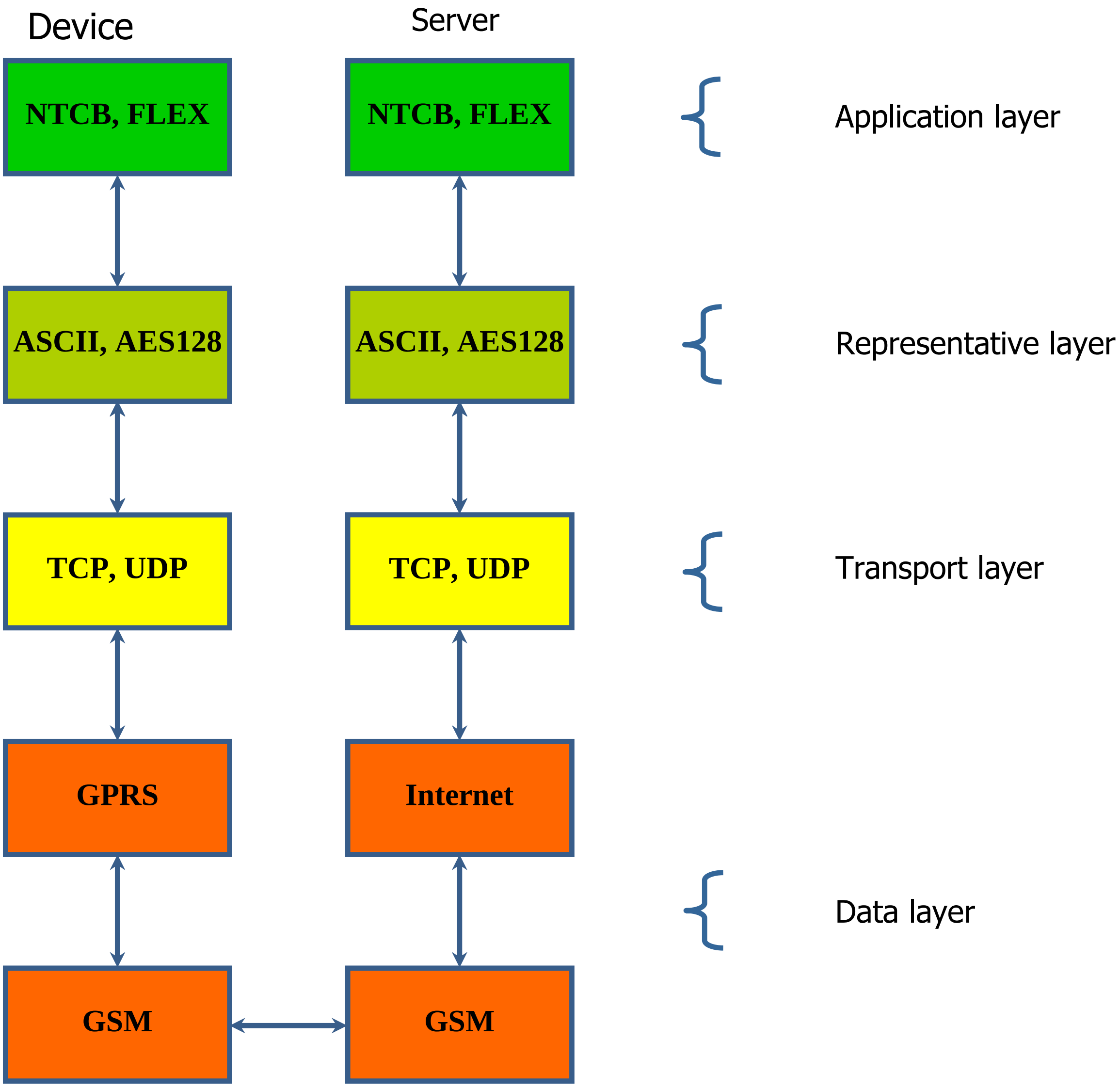
For transmitting full information about system, changing settings and internal software, a binary protocol for data exchange via USB, GPRS is used. NTCB protocol is divided into application and transport layers.

When exchanging over GPRS, the device supports various variants of the representative and transport levels of exchange. Representative levels: 1) ASCII - data without Transcription using the appropriate encoding table; 2) AES128 - data encrypted with a symmetric block Transcription algorithm. Transport levels: 1) TCP - exchange protocol, provides data transmission in networks and TCP / IP subnets; 2) UDP - exchange protocol for data transmission in IP networks without a connection. The work on UDP puncture does not differ from TCP in terms of the device's interaction with the server.

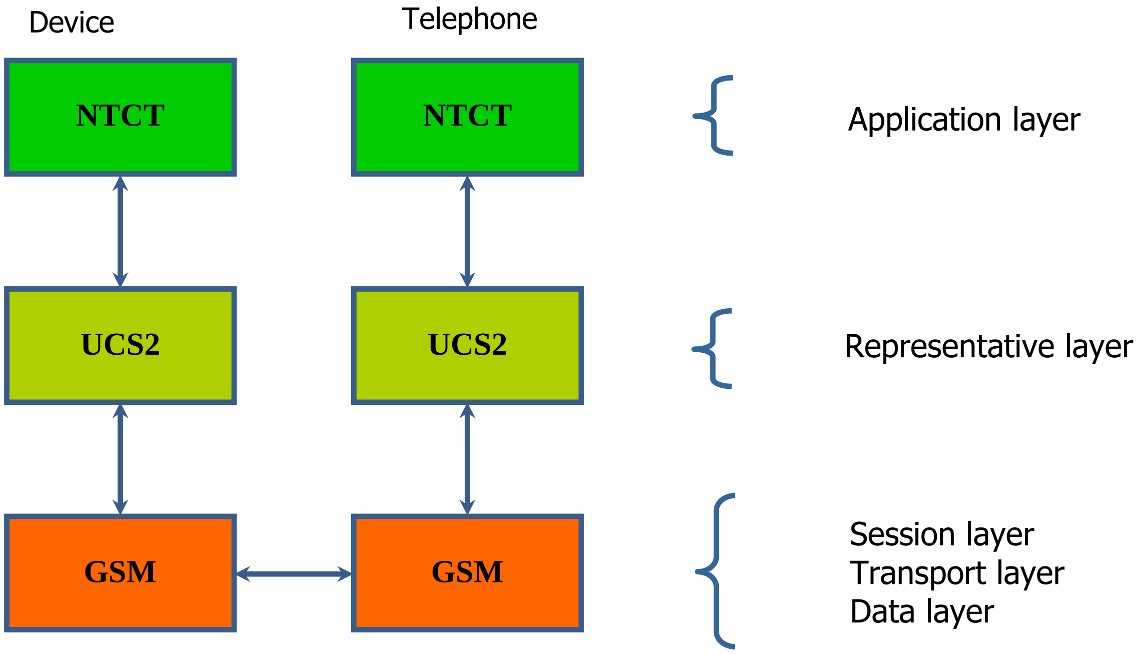
The scheme of interaction between the device and the computer



The scheme of interaction between the device and the computer



The scheme of interaction between the device and GSM phone (SMS)



1. NTCB binary protocol with FLEX extending

NTCB is a basic protocol with static telemetry packages. Under static telemetry packages means the structure of telemetry records: F1, F2, F3, F4, F5, F5.1, F5.2, F6. At this moment, the part of the protocol relating to static telemetry messages is not used, the FLEX protocol extension is used. FLEX protocol is a set of telemetry messages which allow to adjustment information transmission flexibly. FLEX also includes expansion of a set of commands and messages for operation with the device and periphery.

NTCB protocols allow to use packages with the length no more than 65551 bytes. Each message in the frame of this protocol consists of 2 parts: /// and data of the applicable layer. (telemetry information).

NTCB message structure

NTCB message field	<HEAD>(transport layer title)	<BODY> (transport layer data)
Field size in bites	16	from 0 to 65535

At the Internet-connection of the device with server via GPRS the packages can be transmitted both at the initiative of the server, and at the initiative of device. However, protocol NTCB is not full duplex, that is, simultaneous normal transfer of packages to both parties is impossible. When working with a telematics server, the device is the initiator of the transmission of telemetric packages. If a situation of simultaneous transmission of data from the device and a command from the server occurs, it is possible that the above exchange order will be violated. Initially, the device executes a command or request and sends confirmation to the server. After that, the device proceeds to waiting of confirmation of telemetric information reception by the server. In some cases, the device may skip an incoming package, if at the same time it sends another package (the incoming package may be strongly delayed or not received at all). In this case, it is necessary, in the absence of a response to the sent command, to provide for the command repeat. During active data exchange between the server and the device, it is necessary to send a command to the device instead of confirming to the telemetry package, then, waiting for the command to be executed, it is necessary to send an acknowledgment or wait for the telemetry package to be retransmitted.

If there is no confirmation of information receipt from the server, the device will repeat telematics package sending through pauses with a duration of 20 to 90 seconds depending on the signal quality of the cellular network. Also it is necessary to pay attention to the fact that the indicated time intervals depend not only on the signal quality, but also on the degree of availability of the GPRS channel.

In the information exchange via the USB interface, the exchange of packages begins exclusively at the initiative of the “host” (the master). For each package from the "host" device sends a confirmation or response to the request. Before sending a new package, the “host” must wait for confirmation or pause. If a command or request is not corresponding to the protocol requirements (wrong type, structure is failed, wrong checksum), confirmation of this command will not be sent. Similarly, when sending incorrect telemetry data from the device when connecting via GPRS, the server should not send back confirmation of these data receipt.

When the device is connected to the personal computer via USB, the device is defined as a virtual COM-port. Pause of waiting for an answer should not be less than a second. It is advisable to choose the value of this pause equal to 1 ... 5 seconds. If there is no answer, it is recommended to try again sending the package to the device.

1.1. NTCB protocol transport layer

In the NTCB protocol in the information package a 16-byte transport-layer title is used before the application-level data (commands, requests, acknowledgments, and responses).

.

Transport-layer title structure

Byte order	**0 1 2 3**	**4 5 6 7**	**8 9 10 11**	**12 13**	**14**	**15**
Title field	Preamble	Receiver identificator IDr	Sender identifiacator IDs	Number of data bytes n	Data Checksum CSd	Title Checksum CSp
Dimension and format of field data	char[4]	U32	U32	U16	U8	U8
Default package field values when sending it from device to 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 package field values when it is sent from server to 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 field (preamble, receiver identificator, sender identifiacator) is designed for exact definition of the device and the server when trying to establish connection. The values of these fields are set when the device is configuring. If these parameters are not specified when configuring the device, the device will use the default values of these fields.

The preamble consists of any four characters. By default, the first four characters of this package constitute the string "@NTC". When information is exchanged between the computer and the device via the USB interface, the preamble is always "@NTC" regardless of the device settings.

The identifiers of the "host" (server) and device are specified in the device settings. In the case of a package transfer from a host to a device, the receiver ID of the package corresponds to the device ID, and the sender ID corresponds to the host ID. When sending a reverse package (confirmation), the identifiers **are swapped**: the receiver ID of the package corresponds to the "host" ID, and the sender ID corresponds to the device ID. In order for connection by USB interface and for devices in which these data are not specified, defaults parameters are used: host ID (PC) - 1, device ID – 0.

Data bytes' number indicates the number of bytes of the package that follow the given 16-byte header. The number of bytes cannot exceed 65535.

Checksums used in the title are calculated over the entire length of the data specified in the previous field, using the "exclusive OR" algorithm (XOR) with the help of the following function:

```
unsigned char xor_sum
(
        unsigned char     *buffer,          /* (вх) pointer to the buffer with data          */
        unsigned int      length            /* (вх) bytes number for calculation             */
)
{
        unsigned char temp_sum = 0;

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

        return ( temp_sum );
}
```

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

Then, the checksum of the Csp data is calculated by the first 15 bytes: from the 1-st o the 15-th. This procedure is serves for control of the integrity of the transport layer data.

If the identifiers, the preamble, or the checksums calculated on both sides do not match, this package is considered as damaged and response messages from the receiving party will not be sent.

It is allowed to transfer an "empty" transport layer package in order to maintain a communication channel which consist of 16 bytes of the header without application data. The receiving party should not send the response to this package.

1.2. FLEX protocol. Operation with telematics servers via GPRS.

When operating in FLEX protocol, NTCB protocol preamble is fixed and contains value «@NTC» by default.

FLEX protocol message titles contain a preamble starting with a '~' character. There are no 16-byte titles in these messages. In order to inform the server that the data will be transferred in the FLEX protocol, an additional protocol version negotiation message is sent.

In order to maintain connection, a package consisting of one byte with a fixed value of 0x7F is used. Server confirmation for this package **is not required.**

Thus, FLEX messages can be distinguished from other NTCB messages by the first character:

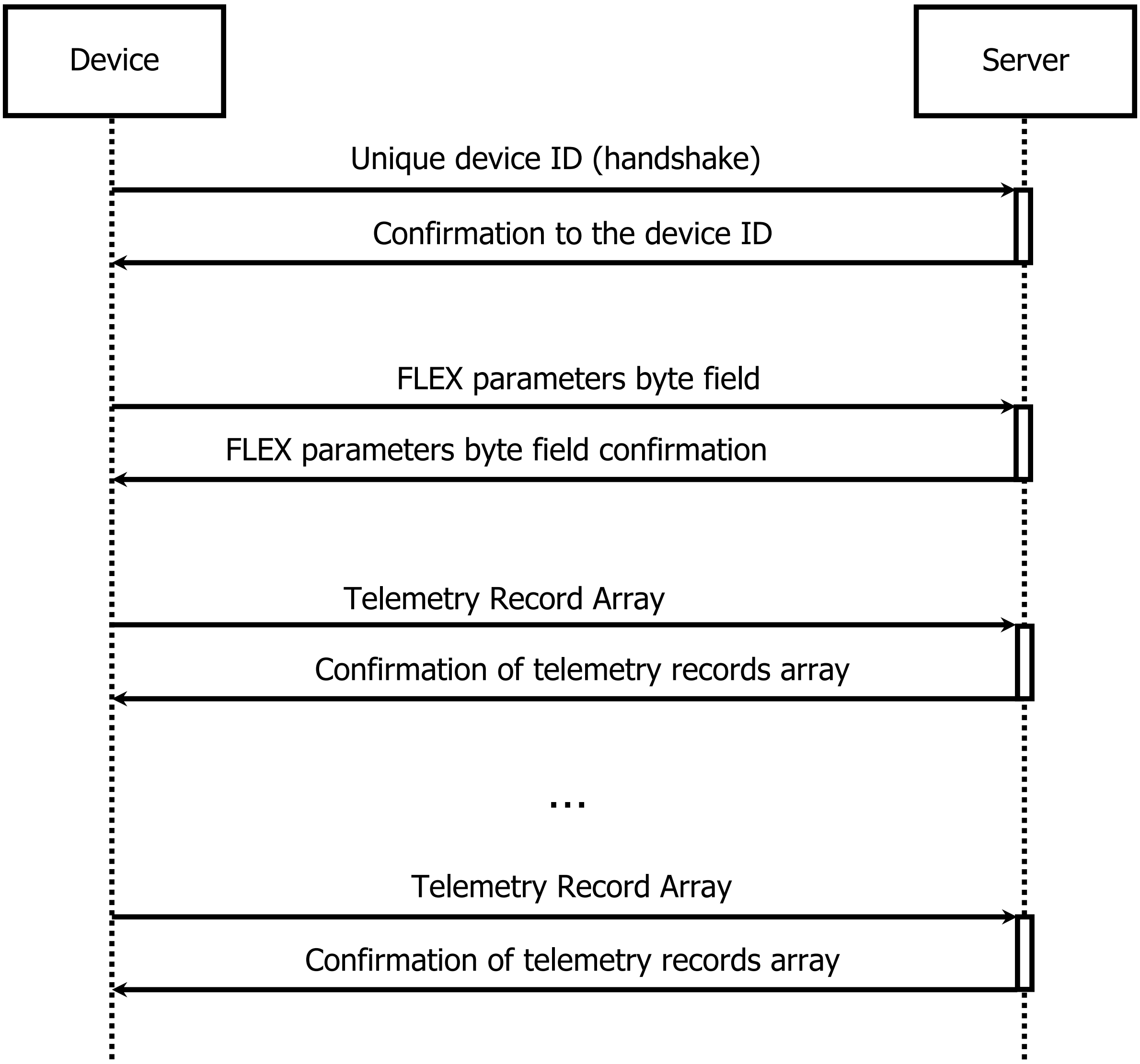
@	- NTCB message containing a 16-byte header;
~	- FLEX message;
DEL (0x7F)	- FLEX message for connection maintaining.

A universal telemetry custom data table is used in this protocol for telemetry information transmission. 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 flag value 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, the offset to the end of the last recorded field occurs.

1.2.1. Basic procedure for establishing connection with the server with the help of the FLEX protocol

When working via GPRS channel, the device will always be the initiator of establishing communication with the server.

The general case of the data exchange procedure with the server:


Device
Server
Unique device ID (handshake)
Confirmation to the device ID
FLEX parameters byte field
FLEX parameters byte field confirmation
Telemetry Record Array
Confirmation of telemetry records array
...
Telemetry Record Array
Confirmation of telemetry records array

Connection with telematics servers is performed according to the settings of the device. After connection was opened (socket), the device individually sends package to the server, which contains of messages with a link of unique ID- handshake. It is possible to extract the protocol settings from this package: preamble, object ID and server ID, as well as a unique device ID (GSM modem IMEI number).

Parameters obtained during the connection are compared with the settings available for this device and, based on the result, the following decision will be made: to allow further work or to terminate the connection. If the parameters match, the device sends a response to this handshake package, which informs the device that it is possible to begin telemetry data transmitting.

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

Message	<HEAD>*>S:<s>	
Response from the server	<HEAD>*<S	
Symbols	**Transcription**	**Data format**
<HEAD>	16-byte NTCB package title with @NTC preamble	U8[16]
*>S	0x2A 0x3E 0x53	char[3]
*<S	0x2A 0x3C 0x53	char[3]
<s>	Строка идентификатора *	char[15]

** This line includes the modem's IMEI, so in order to receive it, the modem must be turned on at least once. When replacing a modem, the unique identifier changes.*

Further the message for protocols version coordination is sent. This message determines the composition and amount of transmitted data, the FLEX protocol version, the version of the data structure.

Message	<HEAD>*>FLEX<protocol><protocol_version><struct_version><data_size><bitfield[data_size/8]>	
Response from the server	<HEAD>*<FLEX<protocol><protocol_version><struct_version>	
Symbols	**Transcription**	**Data format**
<HEAD>	16-byte NTCB package title with @NTC preamble	U8[16]
*>FLEX	0x2A 0x3E 0x46 0x4C 0x45 0x58	char[6]
*<FLEX	0x2A 0x3C 0x46 0x4C 0x45 0x58	char[6]
<protocol>	The symbol of the protocol in which the device is going to operate: 0xB0 – FLEX	U8
<protocol_version>	Protocol version is needed to identify the compatibility of the command set and the package format on the server and in the device. For the version 1.0 - 10 (0x0A) For the version 2.0 - 20 (0x14) For the version 3.0 - 30 (0x1E)	U8
<struct_version>	Version of the data structure is needed to identify the compatibility of the format of the transmitted data on the server and in the device. For the version 1.0 - 10 (0x0A) For the version 2.0 - 20 (0x14) For the version 3.0 - 30 (0x1E)	U8
<data_size>	Size of the subsequent configuration field <bitfield[data_size/8+(1)]> in bytes For version structure 1.0 - 69 For version structure 2.0 - 122 For version structure 3.0 - 255	U8
<bitfield[data_size/8+(1)]>	Bit array with information about the transmitted data structure fields. The transmitted field corresponds to the set bit; with the zero bit, the corresponding field is not transmitted. The value of the array is determined by the device configuration. The length in bytes is calculated as an integer number of bytes that can fit the number specified in the <data_size> field. For the structure of version 1.0 - 9 bytes For the structure of version 2.0 - 16 bytes For the structure of version 3.0 - 32 bytes	[U8] (массив байт)

Structure of bitfield command bit field

If the number of fields "n" is multiple of eight, filling bytes is carried out 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 fild	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	..	n-7................ n

If the number of fields "n" is not multiple of eight, the filling of the last "incomplete" byte begins with the high 7th bit, while 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 fild	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. At that, the number of bytes is equal to 21/8 + 1 = 3 (division is carried out to integer numbers). 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 fild	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	0	0	0

FLEX table fields for which the corresponding bits are not set are omitted. For example, for a field that has assumed the form 0x00 0xE0 0x00 0x00 0x00 0x00 0x00 0x00 0x00 for a FLEX table version 1.0, the transmitted telemetric information will look like this:

№	№ FLEX	Field record	Record element size	Data format	Received values
1	9	Time of the valid coordinates (before the event)	4	U32	Number of seconds since 1970
2	10	The last valid latitude	4	I32	Angle of latitude which was recorded when obtaining the last valid coordinates. In ten thousandths of a minute. For example, 55 ° 42.2389 'will be represented as 33422389
3	11	The last valid longitude	4	I32	Angle of longitude which was recorded when obtaining the last valid coordinates В десятитысячных долях минуты. For example, 37° 41,6063' 'will be represented as 22616063

In order the device starts to transfer data specified in the protocol version, a bitmask version, transferred from the device and in the response from the server should coincide. FLEX data formats have backward compatibility, i.e. data format version 2.0 contains version 1.0 and the added data fields are at the end of the data structure. Protocol version is updated when the format and composition of packages are changed. The device does not send new FLEX messages to incompatible server software. I.e. the device and the server always operate with compatible sets of commands and data.

Consider the example of exchange between the device with updated protocol version and the server which does not support this message. Suppose that the device supports version 2.0 of the data format, while the server supports only the version 1.0 of the data format.

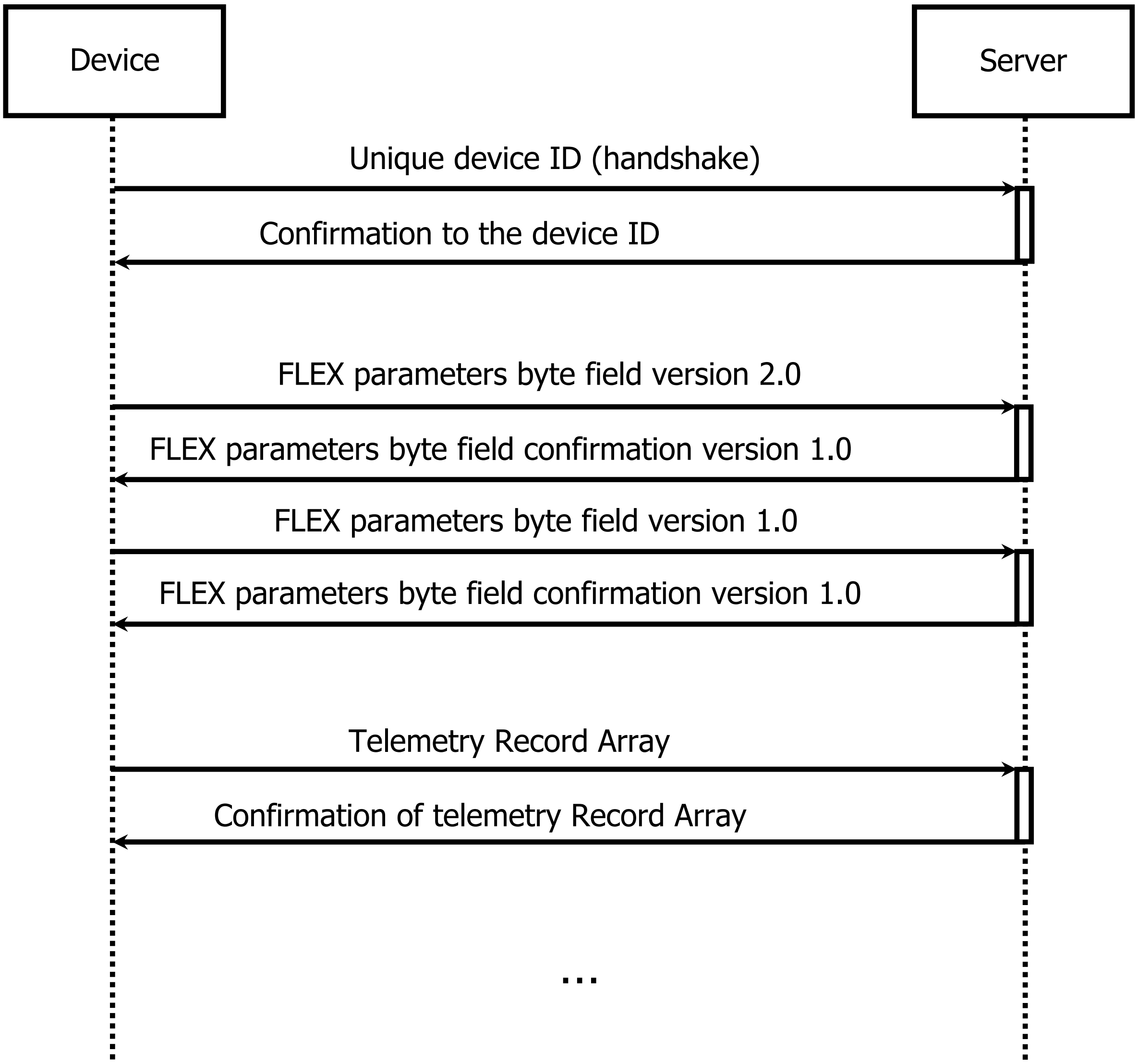
The process of confirmation is as follows.

- the device sends a reconciliation message with bitmask version 2.0;
- the server returns message in which bitmask 1.0 is specified.

If the server again returns a response that contains version 1.0 bitmask, then the data transfer begins. If the server returns a response that contains a divergent bitmask, then the device disconnects from this server. The protocol version confirmation looks the same way and can be done at the same time as the data version negotiation, the server just needs to specify the required protocol version and data structure in the response.

The device monitors amount of unsuccessful attempts of connection with the server. After three attempts, the server is blocked for the interval specified in the configuration.

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


Device
Server
Unique device ID (handshake)
Confirmation to the device ID
FLEX parameters byte field version 2.0
FLEX parameters byte field confirmation version 1.0
FLEX parameters byte field version 1.0
FLEX parameters byte field confirmation version 1.0
Telemetry Record Array
Confirmation of telemetry Record Array
...

After successful procedure of connection to the server, the device sends telematics data. While three types of telematics packages are used:

- A package of an array of telemetry records with events that occurred earlier and the current ones which were not transmitted to the server for any reason; they are contained in non-volatile memory (of the "black box" type);
- A package with extraordinary messages with the current event. Such packages are extraordinary and have a higher priority than the archive one. The telemetry transmitted by the device in an extraordinary message may not be duplicated in the archive message, therefore the parsing of these packages is required.
- The current status package does not have a corresponding event recorded in the archive. It is sent if the device has to transmit telemetry instead of "ping", or is added to the package of the array of telemetry records in case a large fragment of old data is transmitted. The current state has a zero entry index and an event code of 0xFF0.

Package structure of the telemetry array

The implementation of responses from the server to the structures listed in this section is mandatory in order the device work correctly.

Data accumulated in the black box always is transmitted in the package "Telemetry messages array". Package contains certain number of records from "black box". Package size does not exceed 1.3 KB (not counting the title). Packages with one entry are allowed.

Message	~A<size><x[0]-x[size-1]><crc8>	
Response from the server	~A<size><crc8>	
Symbols	**Transcription**	**Data format**
~A	0x7E 0x41	char[2]
<size>	Number of telemetry records, transmitted in as array	U8
<x[0]-x[size-1]>	An array of telemetry records with a FLEX structure. The number of transmitted parameters and the size in bytes corresponds to the value in the <bitfield [data_size / 8 + (1)]> FLEX package field. Records follow each other without any delimiters..	-
<crc8>	8-bit byte CRC8 characters ~ A and fields <size> and <x [0] -x [size-1]> See Appendix Appendix B. CRC8 Checksum Calculation Algorithm 8-bit byte CRC8 ~ A characters and fields <size> and <x [0] -x [size-1]> See Appendix Appendix B. CRC8 Checksum Algorithm.	U8

For the storage and transmission of some rarely changing data, additional packages and telemetry, entered in the protocol extension, starting with FLEX 2.0 are used. These packages replace the standard telemetry.

Message	~E<count><x[0]-x[count-1]><crc8>	
Response from the server	~E<count><crc8>	
Symbols	**Transcription**	**Data format**
<count>	Number of telemetry records, transmitted in as array	U8
<x[0]-x[size-1]>	Array of additional telemetry records. Records follow each other without any delimiters.	-
<crc8>	8-bit byte CRC8 ~ A characters and fields <size> and <x [0] -x [size-1]> See Appendix Appendix B. CRC8 Checksum Algorithm.	U8

Extraordinary message package structure

The implementation of responses from the server to the structures listed in this section is mandatory in order the device work correctly.

The device can transmit important events out of turn, i.e. not in order of increasing the index of messages in the black box. If event is last in the box and was committed to the server, it will not be transmitted in an archive message. Sending an extraordinary message has higher priority than all other packages. Until the reply to this message comes, the device will pause the transmission of the rest packages.

Message	~T<eventindex><x><crc8>	
Response from the server	~T<eventindex><crc8>	
Symbols	**Transcription**	**Data format**
~T	0x7E 0x54	char[2]
<x>	Telemetry recording with FLEX structure. The number of transmitted parameters and the size in bytes corresponds to the value in the <bitfield [data_size / 8 + (1)] field of the FLEX package	-
<eventindex>	Index of received telemetry record	U32
<crc8>	8-bit byte CRC8 ~ A characters and fields <size> and <x [0] -x [size-1]> See Appendix Appendix B. CRC8 Checksum Algorithm.	U8

By analogy with the telemetry message array, for extraordinary there are additional extraordinary packages entered in the protocol extension, starting with FLEX 2.0.

Message	~X<eventindex><x><crc8>	
Response from the server	~X<eventindex><crc8>	
Symbols	**Transcription**	**Data format**
<x>	Additional telemetry record.	-
<eventindex>	Index of received telemetry record.	U32
<crc8>	8-bit byte CRC8 ~ A characters and fields <size> and <x [0] -x [size-1]> See Appendix Appendix B. CRC8 Checksum Algorithm.	U8

Structure of the current state package

The implementation of responses from the server to the structures listed in this section is mandatory for the device correct operation.

Also the device can send data about the current state.
A package can either be added to the transferred arrays (~ A) as the last record, or it can be contained
in a separate package (~ C). The current state has a zero entry index and an event code of 0xFF00

Message	~C<x><crc8>	
Response from the server	~C<crc8>	
Symbols	**Transcription**	**Data format**
~C	0x7E 0x43	char[2]
<x>	Telemetry recording with FLEX structure. The number of transmitted parameters and the size in bytes corresponds to the value in the <bitfield[data_size/8+(1)]> field of the FLEX package	-
<crc8>	8-bit byte CRC8 ~ A characters and fields <size> and <x [0] -x [size-1]> See Appendix Appendix B. CRC8 Checksum Algorithm.	U8

Examples of basic FLEX packages are given in Appendix A.3. Examples of basic FLEX packages.

1.2.2. List of FLEX versions and major messages

List of supported FLEX protocol versions:

Version	Description
FLEX protocol version 1.0	The basic version includes messages described in the FLEX Message List for FLEX 1.0.
FLEX protocol version 2.0	Includes messages described in the FLEX Message List FLEX 2.0. Operation with the following devices is added: camera, tachograph, transparent port. It has features. If the camera is configured, then a special message will be sent to the server. It informs about the presence of images.
FLEX protocol version 3.0	The protocol version is increased because of the introduction of a new version of the data structure.

ATTENTION!
Each the next protocol version includes features of the previous one.
When connecting device, it is necessary to switch to the previous version of the protocol if the FLEX version is not supported.
It is allowed to use different versions of the protocol and data structure simultaneously. For example, the protocol version FLEX 1.0, the version of the data structure FLEX 2.0.

List of supported FLEX data structure versions:

Version	Description
FLEX data structure version 1.0	The basic data structure is presented in Appendix A.1. The structure of telemetry records FLEX format for FLEX 1.0.
FLEX data structure version 2.0	The basic data structure is presented in Appendix A.1. The structure of telemetry records FLEX format for FLEX 2.0. The structure of the additional telemetry record of the FLEX 2.0 extension is presented

	in Appendix A.2. The structure of the additional telemetry records extensions FLEX 2.0.
FLEX data structure version 3.0	The extension of the basic data structure is presented in Appendix A.1. The structure of telemetry records format FLEX for FLEX 3.0.

List of FLEX messages

Message type	**Name**
Telemetry message FLEX 1.0	
~A<size><x[0]-x[size-1]><crc8>	Transmission of accumulated telemetry messages with a structure like FLEX from a black box. Field flags are either taken from the device settings or from the server and are indicated in a separate command at the beginning of the connection.
~T<eventindex><x><crc8>	Transmission of extraordinary FLEX messages with structure like FLEX. Field flags are either taken from the device settings or from the server and are indicated in a separate command at the beginning of the connection.
~C<x><crc8>	Transmission of current telemetry messages with structure like FLEX. Field flags are either taken from the device settings or from the server and are indicated in a separate command at the beginning of the connection.
DEL (0x7F)	Message to support the connection of FLEX protocol.
Telemetry message FLEX 2.0	
~E<count><x[0]-x[count-1]><crc8>	Transmission of accumulated additional telemetry messages with a structure like FLEX 2.0.
~X<eventindex><x><crc8>	Transmission of additional extraordinary messages with structure like FLEX 2.0.
List of FLEX 2.0 service messages	
Q (query)	Data request (version, device state, etc).
I (information)	Response on request, if requested information is available.
U (unavailability)	Response on request, if requested information is unavailable.
O (order)	Order (turning on input lines, arming and etc.).
R (response)	Response on order if it was executed
F (failure)	Response on order if it was not executed
N (notification)	Notification.
G (get)	Data block request.
L (lack)	Negative response to a block request.
D (data)	Data block transmitted at request.
P (put)	Data block downloading.
S (saturation)	Negative response to data block downloading.
M (more)	Confirmation of data block downloading.

Detailed examples of FLEX service messages are located in sections which describe different specific functions.

2. NTCB and FLEX protocol applicable layer
Description of the request package structure, commands and informational massages.

Main types of messages:

1) requests – they are transferred from “host” to the device. Requests are not recorded in the black box of the device, except for the request of the current state of the device;

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

3) commands- they are transferred from the server (“host”) to the device.

Execution of the "host" command tracks when receiving a response to commands. Commands are mapped to events occurring in the device and, if executed, are recorded in a black box.

2.1. System commands, requests, messages

Device model and version request

Request	<HEAD>*?V	
Responce	<HEAD>*#V:<n>:<v1>.<v2>.<v3>:<d>.<m>.<y>:<loc> Exapmple: <HEAD>*#V:E-1110:01.00.53:07.02.08:RU	
Notion	It is possible to send by SMS channel	
Symbol	**Transcription**	**Data format**
<HEAD>	16-byte NTCB package title with preamble	16*U8
*?V	0x2A 0x3F 0x56	char[3]
*#V	0x2A 0x23 0x56	char[3]
<n>	Device model string (up to 6 symbols).	char[6]
<v1>.<v2>.<v3>	Softare version index (2 symbols))	char[2]
<d>.<m>.<y>	Day, month and year of this software version (2 symbols).	char[2]
<loc>	Software language version (2 symbols) (RU, DE, EN).	char[2]

Request for a unique device identifier

Request	<HEAD>*?S	
Responce	<HEAD>*#S:<s>	
Notion	It is possible to send by SMS channel	
Symbol	**Transcription**	**Data format**
<HEAD>	16-byte NTCB package title with preamble	16*U8
*?S	0x2A 0x3F 0x53	char[3]
*#S	0x2A 0x23 0x53	char[3]
<s>	Строка идентификатора.	char[]

Notion

This string includes modem IMEI, so in order to receive it, the modem must be turned on at least once. When replacing a modem, the unique identifier changes.

Message forwarded at handshake procedure with a string of an individual device identifier.

Message	<HEAD>*>S:<s>	
Response from the server	<HEAD>*<S	
Symbol	**Trancription**	**Data format**
<HEAD>	16-byte NTCB package title with preamble	16*U8
*>S	0x2A 0x3E 0x53	char[3]
*<S	0x2A 0x3C 0x53	char[3]
<s>	Identifier string.	char[]

<u>Notion</u>

This string includes modem IMEI, so in order to receive it, the modem must be turned on at least once. When replacing a modem, the unique identifier changes.

Command to reset the device

Message	<HEAD>*!DEV_RESET	
Responce	<HEAD>*@DEV_RESET	
Notion	It is possible to send by SMS channel	
Symbol	**Transcription**	**Data format**
<HEAD>	16-byte NTCB package title 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

Message	<HEAD>*?USSD<s><code>	
Responce	<HEAD>*#USSD<s><string>	
Notion	It is possible to send by SMS channel	
Symbol	**Transcription**	**Data format**
<HEAD>	16-byte NTCB package title with preamble	16*U8
*?USSD	0x2A 0x3F 0x55 0x53 0x53 0x44	char[6]
*#USSD	0x2A 0x23 0x55 0x53 0x53 0x44	char[6]
<s>	The parameter separator- space (0x20).	char
<code>	Line of request USSD code	char[]
<string>	Cellular response line	char[]

Buzzer arbitrary sound indication command

Command	<HEAD>*!BEEP<s><count>,<times>,<interval>,<freq>	
Response	<HEAD>*@BEEP<s><result>	
Notion	It is possible to send by SMS channel	
Symbol	**Transcription**	**Data format**
<HEAD>	16-byte NTCB package title with preamble	16*U8
*!BEEP	0x2A 0x21 0x42 0x45 0x45 0x50	char[6]
*@BEEP	0x2A 0x40 0x42 0x45 0x45 0x50	char[6]

<result>	Command processing result (ASCII):
	«OK» - command is executed;
	«FAIL» - command execution error;
		char[]

<s>	The parameter separator- space (0x20).	char
<count>	Number of pulses in each cycle. If value is> = 32, indication is continuous.	char[]
<times>	Number of cycles in sound indication	char[]
<interval>	Interval between each cycle in 1/128 fractions of a second.	char[]

<freq>	Sound indication frequency (Hz).	char[]

Command to change SIM-card

Command	<HEAD>*!CHNGSIM	
Response	<HEAD>*@CHNGSIM<s><x>-><y>	
Notion	It is possible to send by SMS channel	
Symbol	**Transcription**	**Data format**
<HEAD>	16-byte NTCB package title 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>	The 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

Request for device status via SMS

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

Command	<HEAD>*?ES<s><phone>	
Response	<HEAD>*#ES<s><phone>	
Notion	It is possible to send by SMS channel	
Symbol	**Transcriptrion**	**Data format**
<HEAD>	16-byte NTCB package title with preamble	16*U8
<s>	The parameter separator- space (0x20).	char
*?ES	0x2A 0x3F 0x45 0x53	char[4]
*#ES	0x2A 0x23 0x45 0x53	char[4]
<phone>	Phone number (starting from '+'), at which necessary to send SMS. .	char[]

Request for device status via SMS

Command	<HEAD>*?ICCID	
Response	<HEAD>*#ICCID<s><id>	
Notion	It is possible to send by SMS channel	
Symbol	**Transcription**	**Data format**
<HEAD>	16-byte NTCB package title with preamble	16*U8
<s>	The 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>	SIM-card serial number	char[]

Request for information about GPS receiver firmware version

Command	<HEAD>*?VGPS	
Response	<HEAD>*#VGPS<s><n>,<v1>.<v2>.<v3>,<d>.<m>.<y>,<gps_ver>	
Notion	It is possible to send by SMS channel	
Symbol	**Transcription**	**Data format**
<HEAD>	16-byte NTCB package title with preamble	16*U8
*?VGPS	0x2A 0x3F 0x56 0x47 0x50 0x53	char[6]
*#VGPS	0x2A 0x23 0x56 0x47 0x50 0x53	char[6]
<s>	The parameter separator- space (0x20).	char

<n>	The string of the device model (6 characters).	char[6]
<v1>.<v2>.<v3>	Software version indices of 2 characters.	char[2]
<d>.<m>.<y>	Day, month and year of this software version (2 characters).	char[2]
<gps_ver>	Line with model and version of the navigation receiver.	char[]

Mic Listening Command

Command	<HEAD>*!O<s><phnumber>	
Response	<HEAD>*#O<s><phnumber>	
Notion	It is possible to send by SMS channel	
Symbol	**Transcription**	**Data format**
<HEAD>	16-byte NTCB package title with preamble	16*U8
*!O	0x2A 0x21 0x4F	char[3]
*@O	0x2A 0x40 0x4F	char[3]
<s>	The parameter separator- space (0x20).	char
<phnumber>	<phnumber> is the number to be dialed. Telephone number (starting with '+'	char[]

2.2. Output line control

Command to change the state of the output line

Command	<HEAD>*!SETOUT<s><num><new_state>[,<num><new_state>] Example: <HEAD>*!SETOUT 1Y,2N	
Response	<HEAD>*@SETOUT<s><result>,<num><cur_state>[,<num><cur_state>] Example: <HEAD>*@SETOUT OK,1Y,2N **or** <HEAD>*@SETOUT FAIL,1N,2Y	
Notion	It is possible to send by SMS channel	
Symbol	**Transcription**	**Data format**
<HEAD>	16-byte NTCB package title 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>	The parameter separator- space (0x20).	char

<result>	Command processing result (ASCII):
	«OK» - command is executed;
	«FAIL» - command execution error;
		char[]

<num>	Number of the output line state of which it is necessary to change (ASCII). Numeratoin starts with 1.	char
<new_state>	The state of the output line, which must be installed (ASCII): 'Y' - enable; 'N' - turn off.	char
<cur_state>	The state of the output line after the execution of the command (ASCII): 'Y' - enabled; 'N'- is off.	char

Command to turn on Output line 1

Command	<HEAD>*!1Y	
Response	<HEAD>*@C<x>	
Symbol	**Transcription**	**Data format**
<HEAD>	16-byte NTCB package title with preamble	16*U8
*!1Y	0x2A 0x21 0x31 0x59	char[4]
*@C	0x2A 0x40 0x43	char[3]
<x>	Telemetry record with a structure that depends on the storage and exchange protocol.	-

Command to turn off Output line 1

Command	<HEAD>*!1N	
Response	<HEAD>*@C<x>	
Symbol	**Transcription**	**Data format**
<HEAD>	16-byte NTCB package title with preamble	16*U8
*!1N	0x2A 0x21 0x31 0x4E	char[4]
*@C	0x2A 0x40 0x43	char[3]
<x>	Telemetry record with a structure that depends on the storage and exchange protocol.	-

Command to turn on Output line 2

Command	<HEAD>*!2Y	
Response	<HEAD>*@C<x>	
Symbol	**Transcription**	**Data format**
<HEAD>	16-byte NTCB package title with preamble	16*U8
*!2Y	0x2A 0x21 0x32 0x59	char[4]
*@C	0x2A 0x40 0x43	char[3]
<x>	Telemetry record with a structure that depends on the storage and exchange protocol.	-

Command to turn off Output line 2

Command	<HEAD>*!2N	
Response	<HEAD>*@C<x>	
Symbol	**Transcription**	**Data format**
<HEAD>	16-byte NTCB package title with preamble	16*U8
*!2N	0x2A 0x21 0x32 0x4E	char[4]
*@C	0x2A 0x40 0x43	char[3]
<x>	Telemetry record with a structure that depends on the storage and exchange protocol.	-

Command to turn on Output line 3

Command	<HEAD>*!3Y	
Response	<HEAD>*@C<x>	
Symbol	**Transription**	**Data format**
<HEAD>	16-byte NTCB package title with preamble	16*U8
*!3Y	0x2A 0x21 0x33 0x59	char[4]
*@C	0x2A 0x40 0x43	char[3]
<x>	Telemetry record with a structure that depends on the storage and exchange protocol.	-

Command to turn off Output line 3

Command	<HEAD>*!3N	
Response	<HEAD>*@C<x>	
Symbol	**Transcription**	**Data format**
<HEAD>	16-byte NTCB package title with preamble	16*U8
*!3N	0x2A 0x21 0x33 0x4E	char[4]
*@C	0x2A 0x40 0x43	char[3]
<x>	Telemetry record with a structure that depends on the storage and exchange protocol.	-

Command to turn on Output line 4

Command	<HEAD>*!4Y	
Response	<HEAD>*@C<x>	
Symbol	**Transcription**	**Data format**
<HEAD>	16-byte NTCB package title with preamble	16*U8
*!4Y	0x2A 0x21 0x33 0x4E	char[4]
*@C	0x2A 0x40 0x43	char[3]
<x>	Telemetry record with a structure that depends on the storage and exchange protocol.	-

Command to turn off Output line 4

Command	<HEAD>*!4N	
Response	<HEAD>*@C<x>	
Symbol	**Transciprion**	**Data format**
<HEAD>	16-byte NTCB package title with preamble	16*U8
*!4N	0x2A 0x21 0x33 0x4E	char[4]
*@C	0x2A 0x40 0x43	char[3]
<x>	Telemetry record with a structure that depends on the storage and exchange protocol.	-

2.3. Input line control

Command to lock input line

Command	<HEAD>*!OFF<s><index>	
Response	<HEAD>*@C<x>	
Symbol	**Tanscription**	**Data format**
<HEAD>	16-byte NTCB package title with preamble	16*U8
*!OFF	0x2A 0x21 0x4F 0x46 0x46	char[5]
*@C	0x2A 0x40 0x43	char[3]
<s>	The parameter separator-space (0x20).	char
<index>	The index of the blocking input line (starting with 1). The parameter is written in symbolic form: '1'… '9' (0x31 - 0x39).	char
<x>	Telemetry record with a structure that depends on the storage and exchange protocol.	-

Command to unlock input line

Command	<HEAD>*!ON<s><index>	
Response	<HEAD>*@C<x>	
Symbol	**Transcription**	**Data format**
<HEAD>	16-byte NTCB package title with preamble	16*U8
*!ON	0x2A 0x21 0x4F 0x4E	char[4]
*@C	0x2A 0x40 0x43	char[3]
<s>	The parameter separator- space (0x20).	char
<index>	The index of the unblocking input line (starting with 1). The parameter is written in symbolic form: ‘1’… ‘9’ (0x31 - 0x39).	char
<x>	Telemetry record with a structure that depends on the storage and exchange protocol.	-

2.4. Telemetry commands, requests, messages

Array of telemetric message in FLEX format

Message	~A<size><x[0]-x[size-1]><crc8>	
Response from the server	~A<size><crc8>	
Symbol	**Transcription**	**Data format**
~A	0x7E 0x41	char[2]
<size>	Number of telemetry records, transmitted in the array	U8
<x[0]-x[size-1]>	An array of telemetry records with a FLEX structure. The number of transmitted parameters and the size in bytes correspond to the value in the <bitfield [data_size / 8 + (1)]> FLEX package field. Records follow each other without any delimiters.	-
<crc8>	8-bit byte CRC8 ~ A characters and fields <size> and <x [0] -x [size-1]> See Appendix Appendix B. CRC8 Checksum Algorithm.	U8

FLEX extraordinary message package structure

Message	~T<eventindex><x><crc8>	
Response from the server	~T<eventindex><crc8>	
Symbol	**Transciption**	**Data format**
~T	0x7E 0x54	char[2]
<x>	Telemetry record with FLEX structure. The number of transmitted parameters and the size in bytes corresponds to the value in the <bitfield [data_size / 8 + (1)]> field of the FLEX package.	-
<eventindex>	Index of received telemetry record.	U32
<crc8>	8-bit byte CRC8 ~ A characters and fields <size> and <x [0] -x [size-1]> See Appendix Appendix B. CRC8 Checksum Algorithm.	U8

FLEX format current state package structure

Message	~C<x><crc8>	
Response from the server	~C<crc8>	
Symbol	**Transcriptiton**	**Data format**
~C	0x7E 0x43	char[2]
<x>	Telemetry record with FLEX structure. The number of transmitted parameters and the size in bytes corresponds to the value in the <bitfield[data_size/8+(1)]> field of the FLEX package.	-
<crc8>	8-bit byte CRC8 ~ A characters and fields <size> and <x [0] -x [size-1]> See Appendix Appendix B. CRC8 Checksum Algorithm.	U8

Array of additional FLEX telemetry messages

Message	~E<count><x[0]-x[count-1]><crc8>	
Response from the server	~E<count><crc8>	
Symbol	**Transcription**	**Data format**
<count>	Number of telemetry records transmitted in array	U8
<x[0]-x[size-1]>	Array of additional telemetry records. Records follow each other without any delimiters.	-
<crc8>	8-bit byte CRC8 ~ A characters and fields <size> and <x [0] -x [size-1]> See Appendix Appendix B. CRC8 Checksum	U

	Algorithm.	

FLEX Extraordinary Message Package Structure

Message	~X<eventindex><x><crc8>	
Response from the server	~X<eventindex><crc8>	
Symbol	**Transcription**	**Data fromat**
<x>	Additional telemetry record	-
<eventindex>	Index of received telemetry record.	U32
<crc8>	8-bit byte CRC8 ~ A characters and fields <size> and <x [0] -x [size-1]> See Appendix Appendix B. CRC8 Checksum Algorithm.	U8

Request for telemetry record at the nearest moment before the specified date and time

Message	<HEAD>*?L<h><mn><s><d><m><y>	
Response from the server	<HEAD>*#L<h><mn><s><d><m><y><page><x>	
Symbol	**Transcription**	**Data format**
<HEAD>	16 bytes NTCB package title 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 telemetry record When h=m=s=d=m=y=0, a record record with a minimum number is searched.	U8 (каждого поля)
<page>	Number of non-volatile memory page, at which record is stored.	U32
<x>	Telemetry record with structure which depends on storage and exchange protocol.	-

Request for telemetry record at the nearest moment after the specified date

Request	<HEAD>*?R<h><m><s><d><m><y>	
Response	<HEAD>*#R<h><min><s><d><m><y><page><x>	
Symbol	**Transcription**	**Data format**
<HEAD>	16 bytes NTCB package title 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 telemetry record. When h=m=s=d=m=y=255, a record record with a maximum number is searched.	U8 (каждого поля)
<page>	Number of non-volatile memory page, at which record is stored.	U32
<x>	Telemetry record with structure which depends on storage and exchange protocol.	-

Request for telemetry record according to its index

Request	<HEAD>*?I<index>	
Response	<HEAD>*#I<x>	
Symbol	**Transcription**	**Data format**
<HEAD>	16 bytes NTCB package title with preamble	16*U8
*?I	0x2A 0x3F 0x49	char[3]
*#I	0x2A 0x23 0x49	char[3]
<index>	The absolute index of the record in the black box.	U32
<x>	Telemetry record with structure which depends on storage and exchange protocol.	-

Command to confirm synchronization of the black box with the server

Command	<HEAD>*!SYNC<s><index>	
Response	<HEAD>*@C<x>	
Notion	Sending by SMS channel is possible	
Symbol	**Transcription**	**Data format**
<HEAD>	16 bytes NTCB package title with preamble	16*U8
*!SYNC	0x2A 0x21 0x53 0x59 0x4E 0x43	char[6]
*@C	0x2A 0x40 0x43	char[3]
<s>	The parameter separator - space (0x20) or a colon (0x3A).	char
<index>	Index of server specified in device settings (starting from one). Supports binary recording - 0x01 or character (ASCII) - '1'. In devices without a GPRS channel, this command is executed, but does not carry any meaning.	U8 / char
<x>	Telemetry record with structure which depends on storage and exchange protocol.	-

Notion

This command is executed when data was taken from the device bypassing the GPRS operation algorithm (for example, via USB), and is sent in order to cancel the subsequent transfer according to the main operation algorithm.

Request for the current state of the device

Request	<HEAD>*?A	
Response	<HEAD>*#A<x>	
Notion	**Transcription**	**Data format**
<HEAD>	16 bytes NTCB package title with preamble	16*U8
*?A	0x2A 0x3F 0x41	char[3]
*#A	0x2A 0x23 0x41	char[3]
<x>	Telemetry record with structure which depends on the storage and exchange protocol.	-

Command to resend telemetry from the black box

Command (from the server):	<HEAD>*!REP_FL<s><srvindex>,<leftdate>[/<lefttime>],<rightdate>[/<rightdate>] For example: <HEAD>*!REP_FL 1,09.04.18/13:00:59,10.04.18/03:00:00 **or** <HEAD>*!REP_FL 1,09.04.18,10.04.18	
Response (from the server):	<HEAD>*@REP_FL<s><result>	
Notion	Sending via SMS channel is possible	
Symbol	**Transcription**	**Data format**
<HEAD>	16 bytes NTCB package title 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
/	Separator of fields of date and time (0x2F)	char
<result>	Command Processing Result (ASCII): "OK" - commands executed "FAIL" – command execution error.	char[]
<srvindex>	Server index for retry in text format: '0' - to all servers; '1' .. '3' - server index.	char
<leftdate>	The date of the left border of the requested telemetry interval in text format (in UTC): "DD.MM.YY".	char[8]
<lefttime>	***Optional parameter.*** The time of the left border of the requested telemetry interval in	char[8]

	text format (in UTC): "HH: MM: SS". The absence of a parameter is equal to the value "00:00:00".	
<rightdate>	Date of the right border of the requested telemetry interval in text format (in UTC): "DD.MM.YY".	char[8]
<righttime>	***Optional parameter.*** The time of the right border of the requested telemetry interval in text format (in UTC): "HH: MM: SS". The absence of a parameter is equal to the value «23:59:59».	char[8]

Command to resend telemetry from the SD card

Command (from the server):	<HEAD>*!REP_SD<s><srvindex>,<leftdate>[/<lefttime>],<rightdate>[/<rightdate>] For example: <HEAD>*!REP_SD 1,09.04.18/13:00:59,10.04.18/03:00:00 **or** <HEAD>*!REP_SD 1,09.04.18,10.04.18	
Response (from the device):	<HEAD>*@REP_FL<s><result>	
Notion	Sending via SMS channel is possible	
Symbol	**Transcription**	**Data format**
<HEAD>	16 bytes NTCB package title 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
/	Separator of fields of date and time (0x2F)	char
<result>	Command Processing Result (ASCII): "OK" - commands executed "FAIL" – command execution error.	char[]
<srvindex>	Server index for retry in text format: '0' - to all servers; '1' .. '3' - server index.	char
<leftdate>	The date of the left border of the requested telemetry interval in text format (in UTC): "DD.MM.YY".	char[8]
<lefttime>	***Optional parameter.*** The time of the left border of the requested telemetry interval in text format (in UTC): "HH: MM: SS". The absence of a parameter is equal to the value "00:00:00".	char[8]
<rightdate>	Date of the right border of the requested telemetry interval in text format (in UTC): "DD.MM.YY".	char[8]
<righttime>	***Optional parameter.*** The time of the right border of the requested telemetry interval in text format (in UTC): "HH: MM: SS". The absence of a parameter is equal to the value «23:59:59».	char[8]

2.5. Device operation modes control

Command for arming

Command	<HEAD>*!GY	
Response	<HEAD>*@C<x>	
Symbol	**Transcription**	**Data format**
<HEAD>	16 bytes NTCB package title with preamble	16*U8
*!GY	0x2A 0x21 0x47 0x59	char[4]
*@C	0x2A 0x40 0x43	char[3]
<x>	Telemetry record with structure which depends on storage and exchange protocol.	-

Command for disarming

Command	<HEAD>*!GN	
Response	<HEAD>*@C<x>	
Symbol	**Transcription**	**Data format**
<HEAD>	16 bytes NTCB package title with preamble	16*U8
*!GN	0x2A 0x21 0x47 0x4E	char[4]
*@C	0x2A 0x40 0x43	char[3]
<x>	Telemetry record with structure which depends on storage and exchange protocol.	-

Command to change device operation mode

Command	<HEAD>*!M<s><x>	
Responce	<HEAD>*@M<s><x>,<e>,<i>	
Notion	Sending via SMS channel is possible	
Symbol	**Transcription**	**Data format**
<HEAD>	16 bytes NTCB package title with preamble	16*U8
*!M	0x2A 0x21 0x4D	char[3]
*@M	0x2A 0x40 0x4D	char[3]
<s>	Parameter separator- space (0x20).	char
<x>	Device operation mode in which it is necessary to switch: 'G' – security; 'O' – observation. In the answer <x> is the current operating mode.	char
<e>	Reason of not switching to the security mode: '1' - the protection mode in the device configuration is off; '2' - the timeout for the prohibition on changing the regime has not expired; '3' - the mode is on: do not go into security mode with the ignition on; '4' - the device is already in this mode; '5' - the mode is on: do not switched to armed mode if one of the security sensors has been triggered.	char
<i>	Aliases of activated sensors through space (0x20). For devices S-2551, S-2333: "IN1", "IN2", "IN3", "IN4", "AIN1", "AIN2", "VOLT" – supply voltage sensor, "ENG" – engine operation mode. For devices S-243X: "IN1", "IN2", "IN3", "IN4", "IN5", "IN6", "VOLT" – supply voltage sensor, "ENG" – engine operation sensor.	char[]

Request for the current operation mode

Command	<HEAD>*?M	
Response	<HEAD>*#M<s><x>	
Notion	Sending via SMS channel is possible	
Symbols	**Transcription**	**Data format**
<HEAD>	16 bytes NTCB package title with preamble	16*U8
*?M	0x2A 0x3F 0x4D	char[3]
*#M	0x2A 0x23 0x4D	char[3]
<s>	Parameter separator- space (0x20).	char
<x>	Device operation mode in which it is necessary to switch: 'G' – security; 'O' – observation	char

2.6. Connection to RCS, RFU services

Device connects to the RCS and RFU servers with the help of the appropriate commands via SMS (5.2.5 RCS, RFU services) or GPRS channels.

RCS (remote control service) is designed to provide information interaction between the NTC Configurator program and the device via GPRS in order to change its configuration, update the program, and also to read telemetry from the device. RCS service is essentially a router server that provides information exchange between a device and a program operating with it on a PC.
Switching occurs by the unique ID (session identifier) of this router, which is received by the program that opens the connection to this service. The identifier is transmitted to the device.

Command to connect with configurator through RCS service

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>	
Notion	Sending via SMS channel is possible	
Symbol	**Transcription**	**Data format**
<HEAD>	16 bytes NTCB package title with preamble	U8[16]
*!CNCT_RCS	0x2A 0x21 0x43 0x4E 0x43 0x54 0x5F 0x52 0x43 0x53	char[10]
*@C	0x2A 0x40 0x43	char[3]
<s>	Parameters separator- space (0x20).	char
<ip>	A string containing the IP address of the RCS service server. For example: 89.208.152.55.	char[] min- 7 symbols, max- 15 symbols
<port>	A string containing the port on which the RCS service is started on the server. For example: 8100.	char[] max- 5 symbols
<commID>	Session ID for connection with the configurator. For example: 43644176.	char[] max- 8 symbols
<apn>	***Optional parameter.*** Access point name of cellular operator. If the field is empty in the settings of the mobile operator, the field must be omitted. For example: internet.mts.ru.	char[] max- 30 symbols
<login>	***Optional parameter.*** Login of cellular operator. оператора сотовой связи. If the field is empty in the settings of the mobile operator, the field must be omitted. For example: mts.	char[] max- 20 symbols
<password>	***Optional parameter.*** Password of cellular operator. If the field is empty in the settings of the mobile operator, the field must be omitted. For example: mts.	char[] max- 20 symbols
<x>	Telemetry record with structure which depends on the storage and exchange protocol.	U8[]

In the above message, it is obligatory to specify the IP and PORT of the RCS server, as well as the session ID. If the APN, LOGIN and PASSWORD settings are not passed, the device will use the values of these parameters from its own settings.

RFU service is designed for automotive firmware change in the device. Device connects with the RFU server by command: via SMS channel (5.2.5 RCS, RFU services) or GPRS.
The principle of operation of the device with the RFU server is the following: first, the device is remotely connected, then the device downloads a new version of the program from the server. Then, it reboots, then work is done on the new version of the program.

Command for connection with RFU service for device firmware updating

Command	<HEAD>*!CNCT_RFU<s><ip>,<port>,<firmware>,<apn>,<login>,<password> **or (for backward compatibility)** <HEAD>*!CNCT_RFU:<ip>:<port>:<firmware>:<apn>:<login>:<password>	
Responce	<HEAD>*@C<x>	
Notion	Sending via SMS channel is possible	
Symbols	**Transcription**	**Data format**
<HEAD>	16 bytes NTCB package title with preamble	U8[16]
*!CNCT_RFU	0x2A 0x21 0x43 0x4E 0x43 0x54 0x5F 0x52 0x46 0x55	char[10]
*@C	0x2A 0x40 0x43	char[3]
<s>	Parameters separator- space (0x20).	char
<ip>	A string containing the IP address of the RCS service server. For example: 89.208.152.55.	char[] min- 7 symbols, max- 15 symbols
<port>	A string containing the port on which the RCS service is started on the server. For example: 9100.	char[] max-5 symbols
<firmware>	Request version of software. For example: 02.01.00. It the last wersion is required, key world "LAST" is required.	char[] max-8 symbols
<apn>	***Optional parameter.*** Access point name of cellular operator. If the field is empty in the settings of the mobile operator, the field must be omitted. For example: internet.mts.ru.	char[] max-30 symbols
<login>	***Optional parameter.*** Login of cellular operator. оператора сотовой связи. If the field is empty in the settings of the mobile operator, the field must be omitted. For example: mts.	char[] max-20 symbols
<password>	***Optional parameter.*** Password of cellular operator. If the field is empty in the settings of the mobile operator, the field must be omitted. For example: mts.	char[] max-20 symbols
<x>	Telemetry record with structure which depends on the storage and exchange protocol.	U8[]

In the above message, it is obligatory to specify the IP and PORT of the RCS server, as well as the session ID. If the APN, LOGIN and PASSWORD settings are not passed, the device will use the values of these parameters from its own settings

After receiving RCS and RFU commands, device generates and transmits response to the telemetry server, then it disconnects with telemetry server and establishes connection with appropriate RCS and RFU server. If the device is connected to the RFU service, only request packages are used. In the case of a connection to the RCS service, only the connection establishment package is transmitted. Further, the device works the same way as on the USB interface.

2.7. Operation with Touch Memory key

In addition to other telemetry data, data about the attached Touch Memory keys can come to the server. When such a key is applied to the contact pads of the Touch Memory controller, its number and current reading time are recorded in the non-volatile memory of the system, and then this message is transmitted to the server. This package has a higher priority than sending an array of telemetric records or the current state, but lower priority than an extraordinary message.

The structure of the NTCB package for sending an unregistered Touch Memory key code

<table>
<tr><td><b>Message</b></td><td colspan="2">&lt;HEAD>*>TMKEY<datetime>:<code&gt;</td></tr>
<tr><td><b>Response from the server</b></td><td colspan="2">&lt;HEAD>*&lt;TMKEY</td></tr>
<tr><td><b>Symblols</b></td><td><b>Transcription</b></td><td><b>Data format</b></td></tr>
<tr><td>&lt;HEAD&gt;</td><td>16 bytes NTCB package title with preamble</td><td>16*U8</td></tr>
<tr><td>*&gt;TMKEY</td><td>0x2A 0x3E 0x54 0x4D 0x4B 0x45 0x59</td><td>char[7]</td></tr>
<tr><td>*&lt;TMKEY</td><td>0x2A 0x3C 0x54 0x4D 0x4B 0x45 0x59</td><td>char[7]</td></tr>
<tr><td>&lt;datetime&gt;</td><td>Event time (record formation) on-board device. Time and date of event fixation:
<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 (from 2000 year)</td></tr>
</table></td><td>U8 U8 U8 U8 U8 U8</td></tr>
<tr><td>&lt;code&gt;</td><td>The ID of the applied TM device in its entirety without device type and without checksum.</td><td>U64</td></tr>
</table>

When the protocol FLEX 2.0 is used, key sending is possible in a type of additional telemetry records. See par. Appendix A.2. The structure of additional telemetry recordings of the FLEX 2.0 extension.
Keys registered in the device are transmitted in telemetry packages in the form of an event indicating the slot number in the configuration where the corresponding key is recorded.
See events 0x1900 - 0x2040 in the file “Telematics event code table”.

Request for the last Touch Memory key code read by the device

Request	<HEAD>*?TM	
Response	<HEAD>*#TM<key>	
Symbols	**Transcription**	**Data format**
<HEAD>	16 bytes NTCB package title with preamble	16*U8
*?TM	0x2A 0x3F 0x54 0x4D	Char[4]
*#TM	0x2A 0x23 0x54 0x4D	Char[4]
<key>	TouchMemory Key	U64

Request for the last active radio tag

Request	<HEAD>*?ERFT	
Response	<HEAD>*#ERFT<ID><pwr>	
Symbols	**Transcription**	**Data format**
<HEAD>	16 bytes NTCB package title 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	RFID signal power in dBm	S8

Command for editing Touch Memory keys registered in the device

Page with Touch Memory parameters has to be downloaded into the device in advance.
It is possible to edit only one key at a time, after each command a mandatory reboot of the device occurs.

Command (from the server)	~O<module><id><msg_length><message><crc8>	
Response to a command in case if it is not executed	~F<module><command><result><crc8>	
Response to a command if it is executed	~R<module><command><crc8>	
Symbols	**Transcription**	**Data format**
~O	0x7E 0x4F	2*U8
~F	0x7E 0x46	2*U8
~R	0x7E 0x52	2*U8
<module>	Module code: 0x7D – settings editor.	U8
<command>	Command code: 4 – TM key editing command.	U8
<msg_length>	The length of the message. Up to 139 characters inclusive.	U8 (little-endian)

<message>	Text message encoded in CP1251. No terminal zero at the end of the message is required.
	Content:<num><sp> <address><sp><nick_name><sp><mode>
	Transcription:
	<sp> - space;
	<num> - number of key in the configuration (2 symbols always, 1-64, for example "01");
	<address> - key address (16 symbols);
	<nick_name> - key alias (10 symbols);
	<mode> - key operation mode (5 symbols):
	«NOACT» – no action;
	«GUARD» – changes operation mode;
	«IMMOB» – «immobilizer».
		<msg_length>*U8

<result>	Command execution result Code:
	0x20 – error: parameters are nor set;
	0x21 – error in parameter№1;
	0x22 – error in parameter №2;
	0x23 – error in parameter №3;
	0x24 – error in parameter №4;
	0x25 – error in parameter №5;
	0x26 – error in parameter №6;
	0x27 – error in parameter №7;
	0x28 – error in parameter №8;
		U8

<crc8>	Checksum according to LEX protocol. See Appendix B. CRC8 checksum calculation algorithm.	U8

2.8. Operation with tachograph

2.8.1. Commands and Queries control

Obtaining information which is not recorded in the telemetry or DDD files uploading is carried out using commands and queries by the FLEX 2.0 protocol. Information about number of installed driver's card is transmitted to the server in additional telemetry packages, see par. Appendix A.2. Structure of additional telemetry records of the FLEX 2.0 extension.

Informational requests:

1. Request for information about the current state of the tachograph;
2. Request for information about card №1;
3. Request for information about card №2;
4. Request for registration information.

Management commands:

1. Authorization of the device in the tachograph;
2. Installing a new device authorization key in the tachograph

Request for information about the current state of the tachograph

<table>
<tr><td><b>Request</b></td><td colspan="2">~Q&lt;module><query><crc8&gt;</td></tr>
<tr><td><b>Positive response to the request</b></td><td colspan="2">~I&lt;module><query><time><state><cards_state><drivercard1><drivercard2><mileageTRIP><voltage><drv1AT><drv1DT><drv1CT><drv1RT><drv2AT><drv2DT><drv2CT><drv2RT><crc8&gt;</td></tr>
<tr><td><b>Negative response to the request</b></td><td colspan="2">~U&lt;module><query><result><crc8&gt;</td></tr>
<tr><td><b>Symbols</b></td><td><b>Transcription</b></td><td><b>Data format</b></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>&lt;module&gt;</td><td>Module code:
0x81 – tachograph</td><td>U8</td></tr>
<tr><td>&lt;query&gt;</td><td>Request code:
0x00 – request for information about the current state of the tachograph</td><td>U8</td></tr>
<tr><td>&lt;time&gt;</td><td>Time according to tachograph in UNIX format</td><td>U32 (little-endian)</td></tr>
<tr><td>&lt;state&gt;</td><td>Tachograph and cryptographic information protection means (CIPM) operation mode:
<table>
<tr><td>Bites</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>CIPM operation mode</td><td>Depends on tachograph</td></tr>
</table></td><td>U8</td></tr>
<tr><td>&lt;cards_state&gt;</td><td>Card state:
<table>
<tr><td>Bites</td><td>Description</td><td>Values</td></tr>
<tr><td>0…3</td><td>Card state №1</td><td>0 =no card, 1=not authorized, 2=authorized, 3=failed to remove</td></tr>
<tr><td>4…7</td><td>Card state №2</td><td>0 = no card, 1= not authorized, 2=authorized, 3=failed to remove</td></tr>
</table></td><td>U8</td></tr>
<tr><td>&lt;drivercard1&gt;</td><td>Card type №1 and driver activity №1:</td><td>U8</td></tr>
</table>

Bites	Description	Values
0…3	Driver activity	0=rest, 1=availability, 2=work, 3=driving
4…7	Card type	0 = card is missing; 1 = driver; 2 = master; 3 = controller; 4 = enterprise

<drivercard2>	Card type №2 and driver activity№2 (see Card type №1 and driver activity №1)	U8
<mileageTRIP>	TRIP distance in 0,005 km	U32 (little-endian)
<voltage>	On-board voltage 0.2 V	U8
<drv1AT>	The time spent by driver No. 1 in the current mode (minutes)	U16 (little-endian)
<drv1DT>	Total time of work of driver №1 per day (in minutes)	U16 (little-endian)
<drv1CT>	Continuous time of driver No. 1 work(minutes)	U16 (little-endian)
<drv1RT>	The time of the total breaks of the driver No. 1 (minutes)	U16 (little-endian)
<drv2AT>	The time spent by driver No. 2 in the current mode (minutes)	U16 (little-endian)
<drv2DT>	Total time of work of driver №2 per day (in minutes)	U16 (little-endian)
<drv2CT>	Continuous time of driver No. 2 work(minutes)	U16 (little-endian)
<drv2RT>	The time of the total breaks of the driver No. 2 (minutes)	U16 (little-endian)
<result>	Command execution result code	U8
<crc8>	Checksum according to FLEX protocol. See Appendix B. CRC8 checksum calculation algorithm.	U8

Request for information about cards №1 и №2

<table>
<tr><td><b>Request</b></td><td colspan="2">~Q&lt;module><query><crc8&gt;</td></tr>
<tr><td><b>Positive response to the request</b></td><td colspan="2">~I&lt;module><query><type_state><reserved1><issuing><number><reserved2><cp1><text1><cp2><text2><cp3><text3><cp4><text4><crc8&gt;</td></tr>
<tr><td><b>Negative response to the request</b></td><td colspan="2">~U&lt;module><query><result><crc8&gt;</td></tr>
<tr><td><b>Symbols</b></td><td><b>Transcription</b></td><td><b>Data format</b></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>&lt;module&gt;</td><td>Module code:
0x81 – tachograph</td><td>U8</td></tr>
<tr><td>&lt;query&gt;</td><td>Request code:
0x01 – request for information about cards №1;
0x02 – request for information about cards №2.</td><td>U8</td></tr>
<tr><td>&lt;type_state&gt;</td><td>State and type of the card:
<table>
<tr><td>Bites</td><td>Description</td><td>Values</td></tr>
<tr><td>0…3</td><td>Card state</td><td>0 = no card, 1=not authorized, 2=authorized, 3=failed to remove</td></tr>
<tr><td>4…7</td><td>Card type</td><td>0 = card is missing; 1 = driver; 2 = master; 3 = controller; 4 = enterprise</td></tr>
</table></td><td>U8</td></tr>
<tr><td>&lt;reserved1&gt;</td><td>Authentication type (reserve= 0)</td><td>U8</td></tr>
<tr><td>&lt;issuing&gt;</td><td>Country code (Russia– 0x2B)</td><td>U8</td></tr>
<tr><td>&lt;number&gt;</td><td>Card number (for example, “RUD1000002718000”)</td><td>16*U8</td></tr>
<tr><td>&lt;reserved2&gt;</td><td>Card expiry date (reserve = 0)</td><td>U32 (little-endian)</td></tr>
<tr><td>&lt;cp1&gt;</td><td>The code page number of ISO 8859 in which line No. 1 is transmitted</td><td>U8</td></tr>
<tr><td>&lt;cp2&gt;</td><td>The code page number of ISO 8859 in which line No. 2 is transmitted</td><td>U8</td></tr>
<tr><td>&lt;cp3&gt;</td><td>The code page number of ISO 8859 in which line No. 3 is transmitted</td><td>U8</td></tr>
<tr><td>&lt;cp4&gt;</td><td>The code page number of ISO 8859 in which line No. 4 is transmitted</td><td>U8</td></tr>
</table>

<text1>	String №1	35*U8
<text2>	String №2	35*U8
<text3>	String №3	35*U8
<text4>	String №4	35*U8
<result>	Command execution result code	U8
<crc8>	Checksum. See Appendix B. CRC8 checksum calculation algorithm.	U8

Request for registration information

Request	~Q<module><query><crc8>	
Positive response to the request	~I<module><query><version><vin><nation><vrn_cp><vrn><reserved1><speed_limit><next_calib><activation><expiry><serial><reg_no><crc8>	
Negative response to the request	~U<module><query><result><crc8>	
Symbols	**Transcription**	**Data format**
~Q	0x7E 0x51	2*U8
~I	0x7E 0x49	2*U8
~U	0x7E 0x55	2*U8
<module>	Module code: 0x81 – tachograph	U8
<query>	Request code: 0x03 – request for vehicle and tachograph registration information.	U8
<version>	Tachograph version in text view	32*U8
<vin>	Vehicle ID (VIN).	17*U8
<nation>	Country code in which vehicle is registered (Russia – 0x2B)	U8
<vrn_cp>	ISO 8859 code page, in which the vehicle registration number is presented	U8
<vrn>	Vehicle registration number.	13*U8
<reserved1>	Reserve = 0	2*U8
<speed_limit>	Vehicle speed limit (km/h)	U8
<next_calib>	Time of the next tachograph calibration (Unix-time)	U32 (little-endian)
<activation>	CIPM time activization (Unix-time)	U32 (little-endian)
<expiry>	Time of end of activization (Unix-time)	U32 (little-endian)
<serial>	CIPM serial number	16*U8
<reg_no>	CIMP registration number	16*U8
<result>	Command execution result code	U8
<crc8>	Checksum. See Appendix B. CRC8 checksum calculation algorithm.	U8

Device authorization in the tachograph

Command	~O<module><command><login><psswrd><crc8>	
Response to the command if it is executed	~R<module><command><crc8>	
Response to the command if it is not executed	~F<module><command><result><crc8>	
Symbol	**Transcription**	**Data format**
~O	0x7E 0x4F	2*U8
~R	0x7E 0x52	2*U8
~F	0x7E 0x46	2*U8
<module>	Module code: 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 Appendix B. CRC8 checksum calculation algorithm.	U8

Installing a new device authorization key in the tachograph

Command	~O<module><command><old_login><old_psswrd><new_login><new_psswrd><crc8>	
Response to the command if it is executed	~R<module><command><crc8>	
Response to the command if it is not executed	~F<module><command><result><crc8>	
Symbol	**Transcription**	**Data format**
~O	0x7E 0x4F	2*U8
~R	0x7E 0x52	2*U8
~F	0x7E 0x46	2*U8
<module>	Module code: 0x81 – tachograph	U8
<command>	Command code: 0x01 – installation of a new authorization key:	U8
<old_login>	Previous user ID.	3*U8
<old_psswrd>	Previous 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 Appendix B. CRC8 checksum calculation algorithm.	U8

2.8.2. DDD file generation and transferring to the server

The device supports the generation of an uploaded file (DDD file) containing information about the activities of drivers and vehicle operation from the tachograph. In order to start generating a DDD file, a corresponding command is sent to the device, which indicates the type of file, the slot number in which the card is installed, and additional parameters, if necessary.

Command to start generating DDD file

<table>
<tr><th>Command</th><td colspan="2">~O&lt;module><command><type><param><crc8&gt;</td></tr>
<tr><th>Response to the command if it is executed</th><td colspan="2">~R&lt;module><command><size><crc8> (см. примечание)
~R<module><command><size><fn_len><fn><crc8&gt;</td></tr>
<tr><th>Response to the command if it is not executed</th><td colspan="2">~F&lt;module><command><result><crc8&gt;</td></tr>
<tr><th>Symbol</th><th>Transcription</th><th>Data format</th></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>&lt;module&gt;</td><td>Module code:
0x81 – tachograph</td><td>U8</td></tr>
<tr><td>&lt;command&gt;</td><td>Command code:
0x02 – DDD file generation;
0x03 – DDD file generation with subsequent sending to email.</td><td>U8</td></tr>
<tr><td>&lt;type&gt;</td><td>Type of generated file:
0x00 -tachograph (review);
0x01 - tachograph (activity on a specified date)
0x02 - tachograph (events and malfunctions)
0x03 - tachograph (speed mode details)
0x04 - tachograph (technical details)
0x05 - tachograph (unloading data from the card)</td><td>U8</td></tr>
<tr><td>&lt;param&gt;</td><td>Parameter depends on the type of generated file:
- tachograph (activity at the specified date): data in UNIX format.

- tachograph (unloading data from the card):
<table>
<tr><td>Bites</td><td>Description</td><td>Values</td></tr>
<tr><td>0</td><td>Slot number</td><td>0x01 – slot №1,
0x02 – slot №2</td></tr>
<tr><td>1…3</td><td>Mask for elementary files</td><td>reserv = 0xFFFFFF</td></tr>
</table>
<b>Notion:</b> file with codes Files with codes 0x00 ... 0x04 are service files. Therefore, on some tachographs and for the driver’s card, their unloading is impossible.</td><td>U32 (little-endian)</td></tr>
<tr><td>&lt;fn_len&gt;</td><td>The standard name length for a DDD file is up to 74 bytes inclusive</td><td>U8</td></tr>
<tr><td>&lt;fn&gt;</td><td>The standard name for the DDD file (without ‘\ 0’ at the end)</td><td>&lt;fn_len&gt;*U8</td></tr>
<tr><td>&lt;result&gt;</td><td>Command execution result code</td><td>U8</td></tr>
<tr><td>&lt;size&gt;</td><td>Size of generated upload file</td><td>U16 (little-endian)</td></tr>
<tr><td>&lt;crc8&gt;</td><td>Checksum. See Appendix B. CRC8 checksum calculation algorithm.</td><td>U8</td></tr>
</table>

Notion: devices with previous firmware versions (before 06.10.00) can send in response to a command to start generating a DDD file a reduced response that does not contain a standard name for the generated DDD file (~ R <module> <command> <size> <crc8>)

Generation of DDD file takes on average no more than 5 minutes. The size of the generated file does not exceed 64 KB. The generated file is stored in the internal non-volatile memory of the device and is considered relevant for 1 hour. This means that if the device re-receives the command to start generating the DDD file within 1 hour from the last generation of the DDD file, then it will not generate it again, but will use the file previously stored in non-volatile memory.

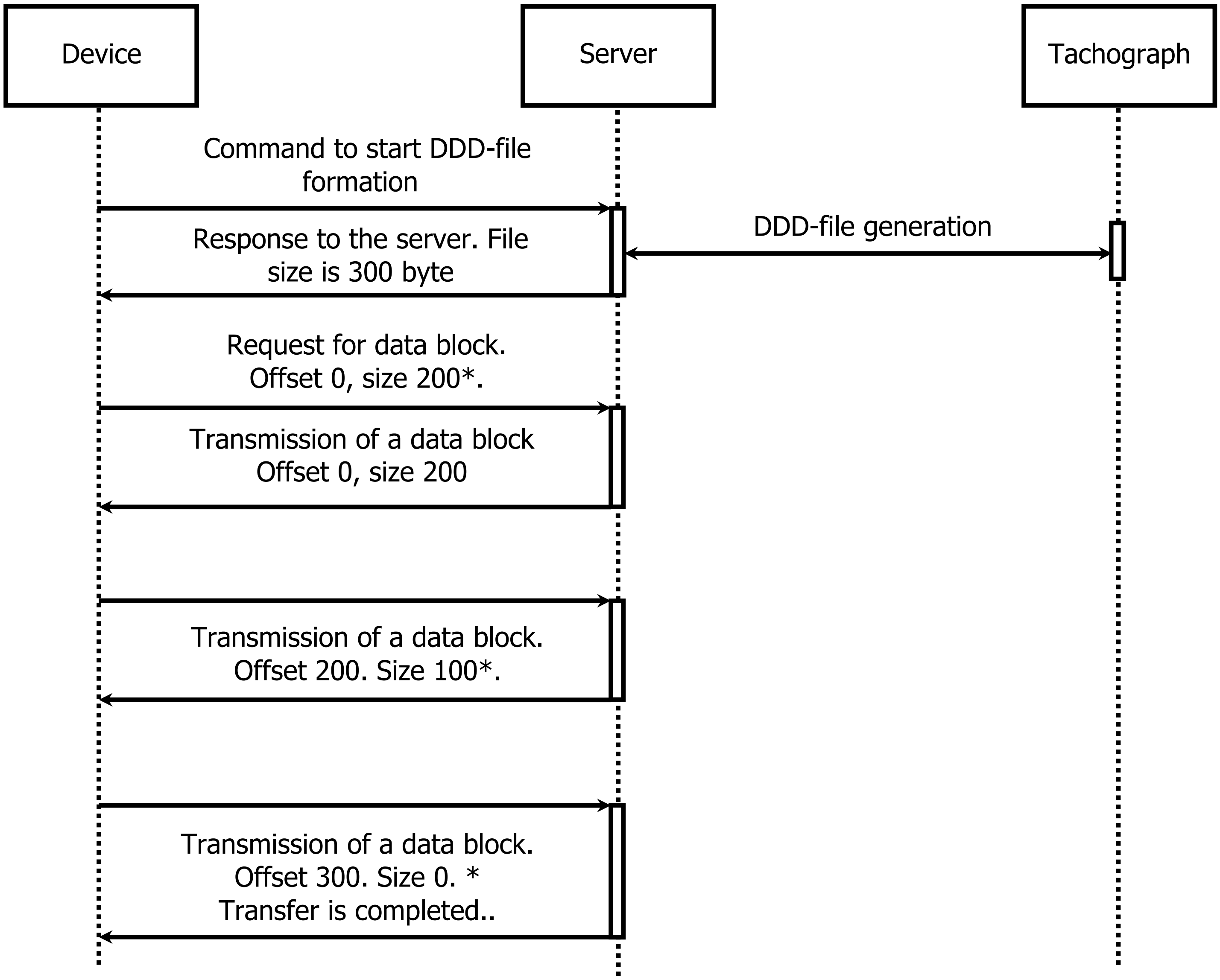
The time period for transferring a file to the server is unlimited. However, due to the fact that the same address space is used to store the DDD file and firmware, after reflashing the DDD device, the file is lost.

DDD file transfer is carried out by blocks of arbitrary length at the initiative of the server. The server alternately requests data blocks using the corresponding request, controls their integrity and transfer order. In each request, the server must specify the number of bytes to read from the DDD file and the offset from its beginning, from which the reading begins. In the response the device sends data block with number of bytes which were read from the file. The maximum block size that can be transferred by the device at a time is 960 bytes. If the data block was not received by the server, a repeated request of the block is possible with the same offset and number of bytes. A file transfer is considered complete when the device sends a file block in which the read number of bytes will be equal to 0.

Request for DDD-file block

Command	~G<module><get_idx><offset><size_need><crc8>	
Response to the command if it is executed	~D<module><get_idx><offset><size_read><data><crc8>	
Response to the command if it is not executed	~L<module><get_idx><result><crc8>	
Symbol	**Transcription**	**Data format**
~G	0x7E 0x47	2*U8
~D	0x7E 0x44	2*U8
~L	0x7E 0x4C	2*U8
<module>	0x81 – тахограф	U8
<get_idx>	Request ID: 0x00 – request for DDD-file block	U8
<offset>	Offset from the beginning of the DDD file in bytes.	U32 (little-endian)
<size_need>	Number of bites which have to be read.	U16 (little-endian)
<size_read>	Number of read data in bytes. The size of the read block may differ from the size of the requested block, but only downward.	U16 (little-endian)
<data>	Data block	<size_read>*U8
<result>	Command execution result code	U8
<crc8>	Checksum. See Appendix B. CRC8 checksum calculation algorithm.	U8

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


Device
Server
Tachograph
Command to start DDD-file formation
Response to the server. File size is 300 byte
DDD-file generation
Request for data block. Offset 0, size 200*.
Transmission of a data block Offset 0, size 200
Transmission of a data block. Offset 200. Size 100*.
Transmission of a data block. Offset 300. Size 0. * Transfer is completed..

2.8.3. Codes of the commands and requests execution results

Code	Description
0x10	command can not be performed, because the previous command has not completed yet
0x20	unknown command/request
0x21	invalid parameters are specified in the command (for the command to change the authorization key)
0x30	device is not configured on the operation with tachograph
0x31	no connection with tachograph
0x32	uploaded file is not generated
0x33	failed to authorized in the tachograph
0x34	saving / reading data from non-volatile memory error
0x35	DDD generation: invalid parameters are specified in the request (data, card number)
0x36	DDD generation: tachograph can not generate file in the current mode of operation
0x37	DDD generation: unsupported file type
0x38	DDD generation: no data for file generation
0x39	DDD generation: receiving file from tachograph error
0x3A	DDD generation: generation is interrupted (when flashing the device)
0x3B	DDD generation: file has incorrect structure (failed to form name of the file)
0x3C	failed to send file to e-mail
0x70	confirmation of receipt of the command (for the command to create the upload file)

2.9. Operation with the driver’s display

The command to send a message to the driver's display using the FLEX protocol via GPRS

Command (from the server):	~O<module><id><index><confirm><msg_length><message><crc8>	
Positive response (from the device):	~R<module><id><crc8>	
Negative response (from the device):	~F<module><id><result><crc8>	
Symbol	**Transcription**	**Data format**
~O	0x7E 0x4F	2*U8
~F	0x7E 0x46	2*U8
~R	0x7E 0x52	2*U8
<module>	Module code: 0x82 – driver’s display DV-01	U8
<id>	Command code: 0 – sending 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 determining whether confirmation of receipt of a message is required: ! (0x21) – confirmation is required, Rest symbols- confirmation is not required.	U8
<msg_length>	Message length. Up to 139 symbols inclusive.	U8 (little-endian)
<message>	A text message to the driver encoded in CP1251. A terminal zero at the end of the message is not required.	<msg_length>*U8
<result>	Code of command execution result: 0x01 –command is executed, but the message was truncated due to exceeding of the maximum length 0x10 – the device has not sent the previous message to the display yet; 0x20 – message length is 0; 0x30 – the device is not configured to work with the display; 0x31 – no connection with the display.	U8
<crc8>	Checksum. See Appendix B. CRC8 checksum calculation algorithm.	U8

Command to send a message to the driver's display using the NTCB protocol via GPRS

Command (from the server):	<NTC HEAD>*!DV<index><confirm><message>	
Response (from the server):	<NTC HEAD>*@DV<result>	
Symbols	**Transcription**	**Data format**
<NTC HEAD>	16 bytes NTCB package title with preamble	16*U8
*!DV	0x2A 0x21 0x44 0x56	4*U8
*@DV	0x2A 0x40 0x44 0x56	4*U8
<index>	Reserved place for the index assigned to the message by the server (not used at this stage and equal to 0xFFFFFFFF)	U32
<confirm>	Symbol determining whether confirmation of receipt of a message is required: ! (0x21) – confirmation is required, Rest symbols- confirmation is not required.	U8
<message>	A text message to the driver encoded in CP1251 and a length up to 139 symbols inclusive. A terminal zero at the end of the message is not required.	N*U8, N – message length (is calculated according to transport layer)
<result>	Code of command execution result: 0x01 – command is executed, but the message was truncated due to exceeding of the maximum length; 0x10 – the device has not sent the previous message to the display yet; 0x20 – message length is 0; 0x30 – the device is not configured to work with the display; 0x31 – no connection with the display.	U8

2.10. Operation with auto-informer

Operate with the autoinformer is carried out according to the NTCB protocol. Notification of events in geofences is transmitted only via USB when the corresponding setting is turned on. Alerts can be used for interracton with the mobile application.

Command to control Auto-informer

This command is used both when working on USB, and when working on GPRS.

Command	<HEAD>*!AINF<code><data>	
Response	<HEAD>*@AINF<code><cop>	
Symbols	**Transcription**	**Data format**
<HEAD>	16 bytes NTCB package title 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 – start sound file playback; 0x03 – change of the current motion mode.	U8
<data>	Data which depend on a command: – change of the current mode: route identifier; – start sound file playback: sound file identifierидентификатор звукового; – change of the current motion mode: motion mode identifier.	U16
<cop>	Parameter indicating the success of the command execution. It takes the following values: 0x00 – command was executed successfully; 0x01 –command was executed, but failed to update settings (for commands to change route or motion mode);	U8

	0x10 –command cannot be executed because previous command has not completed yet; 0x20 – unknown command (for command codes bigger than 0x030x03); 0x30 – route cannot be downloaded (invalid format); 0x31 – failed to download the route (route is not in the list); 0x32 – failed to start sound file playback; 0x33 – sound file not found; 0x34 – failed to specify motion mode (mode is not find in the list); 0x35 – failed to download geo-fence list.	

There are text analogues for this command:

Command to change routes

Command	<HEAD>*!AINF:<number><char>	
Response	<HEAD>*@AINF:<result>	
Symbols	**Transcription**	**Data format**
<HEAD>	16 bytes NTCB package title 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>	Route letter	cp1251
<result>	Literal code of the command execution result: "S0" (0x53 0x30) – command was executed successfully; "S1" (0x53 0x31) – command was executed, but failed to update settings (for commands to change route or motion mode); "B0" (0x42 0x30) – command cannot be executed because previous command has not completed yet; "E0" (0x45 0x30) – route cannot be downloaded (invalid format) "E1" (0x45 0x30) – failed to download the route (route is not in the list); "E5" (0x45 0x30) – failed to download list of geo-fences.	char[2]

Command to play sound file

Command	<HEAD>*!AINF!<soundid>	
Response	<HEAD>*@AINF!<result>	
Symbols	**Transcription**	**Data format**
<HEAD>	16 bytes NTCB package title 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 identifier in text view	cp1251
<result>	Literal code of the command execution result: "S0" (0x53 0x30) – command was executed successfully; "B0" (0x42 0x30) – command cannot be executed because previous command has not completed yet; "E2" (0x45 0x30) –failed to play sound file; "E3" (0x45 0x30) – sound file not found.	char[2]

Command to change speed motion

Command	<HEAD>*!AINF#<spdmodeid>	
Respose	<HEAD>*@AINF#<result>	
Symbols	**Transcription**	**Data format**
<HEAD>	16 bytes NTCB package title 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 motion identifier in text view	cp1251
<result>	Literal code of the command execution result: "S0" (0x53 0x30) – command was executed successfully; "S1" (0x53 0x31) – command was executed, but failed to update settings (for commands to change route or motion mode); "B0" (0x42 0x30) – command cannot be executed because previous command has not completed yet; "E4" (0x45 0x30) – failed to specify motion mode (mode is not find in the list).	char[2]

Notification about Auto-informer events

Message	<HEAD>*&AINF<title><id><data>	
Response	Not required	
Symbols	**Transcription**	**Data format**
<HEAD>	16 bytes NTCB package title with preamble	16*U8
*&AINF	0x2A 0x26 0x41 0x49 0x4E 0x46	char[6]
<title>	Type of notification: 0 – entrance into the geo-fence; 1 – exit from the geo-fence; 2 – start sound file playback.	U8
<id>	Current route identifier	U16
<data>	Depends from the type of notification: – entrance into the geo-fence: geo-fence identifier; – exit from the geo-fence: geo-fence identifier; – sound of sound file playback: sound file identifier.	U16

2.11. Operation with camera

2.11.1. Control Commands and requests

In order to control camera operation and receive information about it, the following features are designed:

1) Command "Auto shooting control"- performs specified number of shootings with specified pause between them;
2) Command "Make shooting" - checks the presence of images for a specified period of time;
3) Request "Receive information about camera" - checks the presence of images for a specified period of time;

Command "Auto shooting control"

Command:	~O<module><cmd_id><param><crc8>	
Positive answer:	~R<module><cmd_id><crc8>	
Negative answer:	~F<module><cmd_id><result><crc8>	
Symbols	**Transcription**	**Data format**
~O	0x7E 0x4F	2*U8
~F	0x7E 0x46	2*U8
~R	0x7E 0x52	2*U8
<module>	Module code:	U8

	0x80 – digital camera	
<cmd_id>	Command code: 0x00 – auto shooting control.	U8
<param>	Auto shooting control: 0x00 – turn off auto shooting, 0x01 – turn on auto shooting.	U8
<result>	Code of the command execution result 2.11.3. Codes of commands and requests execution results.	U8
<crc8>	Checksum. See Appendix B. CRC8 checksum calculation algorithm.	U8

Command "Make shooting"

Command:	~O<module><cmd_id><count><delay><crc8>	
Positive answer:	~R<module><cmd_id><crc8>	
Negative answer:	~F<module><cmd_id><result><crc8>	
Symbols	**Transcription**	**Data format**
~O	0x7E 0x4F	2*U8
~F	0x7E 0x46	2*U8
~R	0x7E 0x52	2*U8
<module>	Module code: 0x80 – digital camera	U8
<cmd_id>	Command code: 0x01 – image is made; 0x02 – to take am image and then send it by email.	U8
<count>	Number of images: 1…65535	U16 (little-endian)
<delay>	Pause between images (seconds): 1…65535	U16 (little-endian)
<result>	Code of the command execution result 2.11.3. Codes of commands and requests execution results.	U8
<crc8>	Checksum. See Appendix B. CRC8 checksum calculation algorithm.	U8

Text analogue of commands to operate with camera

Command	<HEAD>*!<cmd>	
Response	<HEAD>*@<answer>	
Symbols	**Transcription**	**Data format**
<HEAD>	16 bytes NTCB package title with preamble	16*U8
*!	0x2A 0x21	char[2]
*@	0x2A 0x40	char[2]
<cmd>	Command in test view. All SMS commands 4.2.10 are supported. Digital camera.	char[]
<answer>	The answer is in text view. See the table "Responses to Camera Control Commands" (NTCT Text Protocol).	char[]

Request «Receive information about camera»

<table>
<tr><td><b>Request</b></td><td colspan="2">~Q&lt;module><query><crc8&gt;</td></tr>
<tr><td><b>Positive answer to request</b></td><td colspan="2">~I&lt;module><query><flags><version><dir><crc8&gt;</td></tr>
<tr><td><b>Negative answer to request</b></td><td colspan="2">~U&lt;module><query><result><crc8&gt;</td></tr>
<tr><td><b>Symbols</b></td><td><b>Transcription</b></td><td><b>Data format</b></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>&lt;module&gt;</td><td>Module code:
0x80 – digital camera</td><td>U8</td></tr>
<tr><td>&lt;query&gt;</td><td>Request code:
0x00 – receive information about camera.</td><td>U8</td></tr>
<tr><td>&lt;flags&gt;</td><td><table><tr><td><b>Bites</b></td><td><b>Description</b></td><td><b>Value</b></td></tr><tr><td>0</td><td>Auto shooting</td><td>0 – off,
1 – on</td></tr><tr><td>1-7</td><td>Reserv</td><td>0</td></tr></table></td><td>U8</td></tr>
<tr><td>&lt;version&gt;</td><td>Camera version (16 symbols + ‘\0’):
Example, “VC0706 1.00”.</td><td>17*U8</td></tr>
<tr><td>&lt;dir&gt;</td><td>Catalogue in which photos are stored (8 symbols + ‘\0’):
Example, “PHOTOS”.</td><td>9*U8</td></tr>
<tr><td>&lt;result&gt;</td><td>Code of the command completion result 2.11.3. Codes of commands and requests execution results.</td><td>U8</td></tr>
<tr><td>&lt;crc8&gt;</td><td>Checksum. See Appendix B. CRC8 checksum calculation algorithm.</td><td>U8</td></tr>
</table>

2.11.2. Photo transmission to the server

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

To promptly notify the server about the appearance of a new photo, the device sends a notification “Notification about new photo” to the server each time a new photo is created.

Notification about new photo

Notification	~N<module><id><time><crc8>	
Response from the server is not required		
Symbols	**Transcription**	**Data format**
~N	0x7E 0x4E	2*U8
<module>	Module code: 0x80 – digital camera.	U8
<id>	Notification code: 0x00 – notification about creation of a new photo.	U8
<time>	Data and time of creation of the last photo: Unix-time (number of seconds from 00:00:00 of 1 January 1970) in unsigned format.	U32 (little-endian)
<crc8>	Checksum. See Appendix B. CRC8 checksum calculation algorithm	U8

Information about the size of the main and overview pictures, about the availability of pictures for a certain period of time can be obtained using the query “Request information about the picture”. The request indicates the time relative to which the search is performed and the search mode. The response to the request contains the time of creating the image, which is the closest to the search time and satisfies the search conditions.

Request “Request for information about photo”

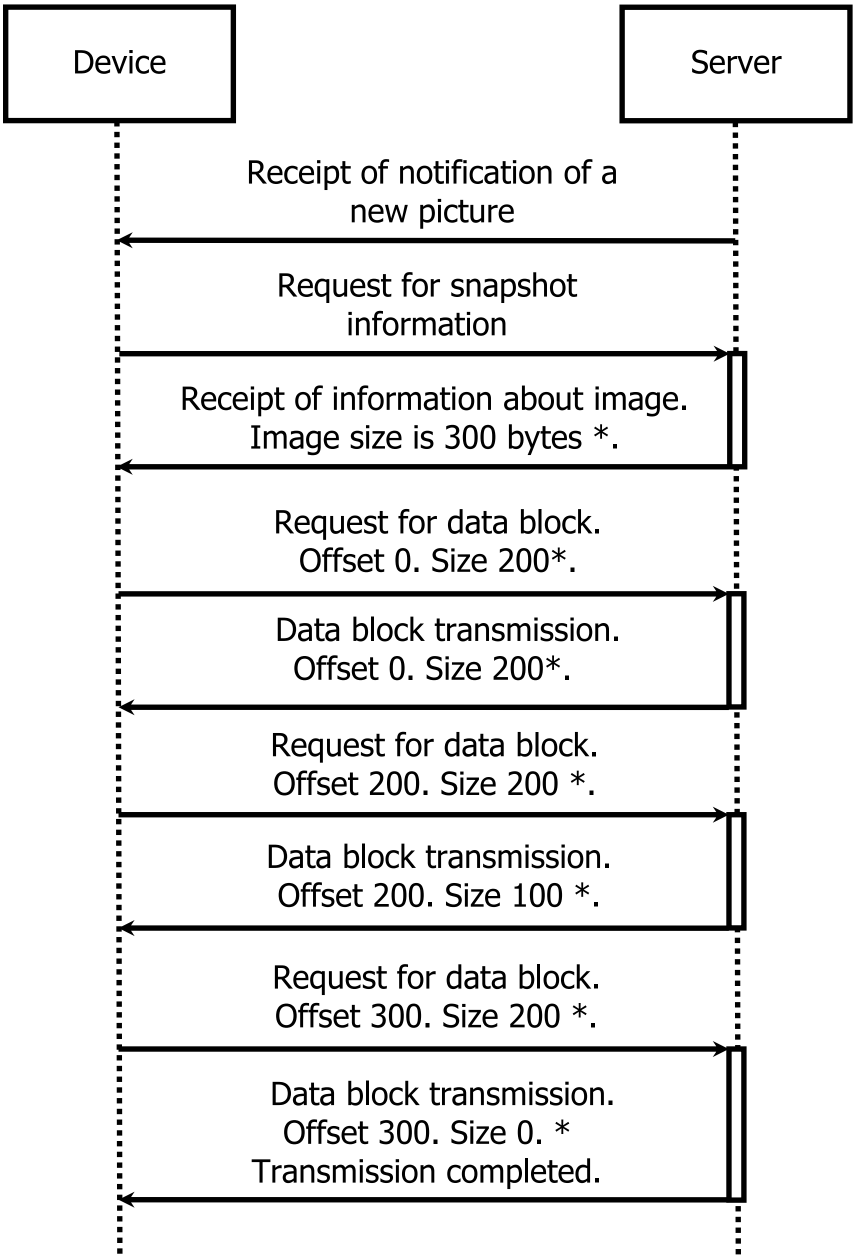
<table>
<tr><th>Request</th><td colspan="2">~Q<module><query><flags><search_time><crc8></td></tr>
<tr><th>Positive response to the request</th><td colspan="2">~I<module><query><time><size_fair><size_rough><crc8></td></tr>
<tr><th>Negative response to the server</th><td colspan="2">~U<module><query><result><crc8></td></tr>
<tr><th>Symbols</th><th>Transcription</th><th>Data format</th></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 code:
0x80 – digital camera</td><td>U8</td></tr>
<tr><td><query></td><td>Request code:
0x01 – request for information about photo.</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 time of creation coincides with the time <search_time></td><td>0 – no,
1 – yes</td></tr>
<tr><td>1</td><td>Search for a photo which was taken before <search_time> (not inclusive)</td><td>0 – no,
1 – yes</td></tr>
<tr><td>2</td><td>Search for a photo which was taken after <search_time> (not inclusive)</td><td>0 – no,
1 –yes</td></tr>
<tr><td>3-7</td><td>Reserv</td><td></td></tr>
</table>
* Bit values can be combined to provide additional search conditions. For example, a combination of bit 0 and bit 1 forms the search condition “Search for an image taken up to the time <search_time> inclusive”.</td><td>U8</td></tr>
<tr><td><search_time></td><td>Date and time of a search: Unix-time (number of seconds from 00:00:00 1 January 1970) in unsigned format.</td><td>U32 (little-endian)</td></tr>
<tr><td><time></td><td>Date and time of creation of found image. (If image is not found, answer ~U with corresponding code is sent.)</td><td>U32 (little-endian)</td></tr>
<tr><td><size_fair></td><td>Size of the main image (byte).</td><td>U16 (little-endian)</td></tr>
<tr><td><size_rough></td><td>Size of overview image (byte).</td><td>U16 (little-endian)</td></tr>
<tr><td><result></td><td>Code of the command completion result 2.11.3. Codes of commands and requests execution results.</td><td>U8</td></tr>
<tr><td><crc8></td><td>Checksum. See Appendix B. CRC8 checksum calculation algorithm</td><td>U8</td></tr>
</table>

Images are transferred to the server in blocks using the “Request image data” command. The command indicates the type of image (main or overview), the time the image was taken, the offset in bytes from the beginning of the image, and the size of the block to be transmitted, so that random access to the image data is possible. It is required when resuming broken dowmload of image and re-requesting blocks of image data. The maximum block size that can be transferred by the device in one time is 960 bytes

Command “Request for image data”

Command	~G<module><get_id><utc_time><offset><size_need><crc8>	
Positive response to the command	~D<module><get_id><utc_time><offset><size_read><data><crc8>	
Negative response to the server (in case of mistake)	~L<module><get_id><result><crc8>	
Symbols	**Transcription**	**Data format**
~G	0x7E 0x47	2*U8
~D	0x7E 0x44	2*U8
~L	0x7E 0x4C	2*U8
<module>	Module code: 0x80 – digial camera	U8
<get_id>	Request code: 0x00 – request for snapshot data; 0x01 – request for snapshot data.	U8
<utc_time>	Time of creation of required image: Unix-time (number of seconds from 00:00:00 1 January 1970) in unsigned format.	U32 (little-endian)
<offset>	Offset in bytes from the beginning of the image file.	U16 (little-endian)
<size_need>	Size of required data block in bytes.	U16 (little-endian)
<size_read>	Size of the read block of image data in bytes. Size of the read block may differ from the size of the requested block, but only downward.	U16 (little-endian)
<data>	Image Data Block.	<size_read>*U8
<result>	Code of the command completion result 2.11.3. Codes of commands and requests execution results.	U8
<crc8>	Check sum.	U8

In general, the procedure for transferring pictures to the server can be represented by the following diagram:


Device
Server
Receipt of notification of a new picture
Request for snapshot information
Receipt of information about image. Image size is 300 bytes *.
Request for data block. Offset 0. Size 200*.
Data block transmission. Offset 0. Size 200*.
Request for data block. Offset 200. Size 200 *.
Data block transmission. Offset 200. Size 100 *.
Request for data block. Offset 300. Size 200 *.
Data block transmission. Offset 300. Size 0. * Transmission completed.

* Offsets and sizes are given for example.

The completion of the image transfer is determined based on the size of the image, the offset from the beginning of the image and the length of the block. With a sequential request for data, the download is considered as completed if a response is received from the device with a block length of 0.

2.11.3. Codes of the results of completion commands and queries

Code	Description
0x10	command cannot be completed, because the previous command was not completed
0x20	unknown command
0x30	device is not configured the operation with camera
0x31	no connection with camera
0x32	auto shooting is off
0x33	operation with SD-card error
0x34	image is not found
0x35	error while receiving images from camera
0x36	failed to send image by e-mail

2.12. Data exchange between external interfaces and server

USB data transfer command from device to server

Command	<HEAD>*!U2S<data>	
Response	<HEAD>*@U2S<cop>	
Symbols	**Transcription**	**Data format**
<HEAD>	16 bytes NTCB package title 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 from 1 to 1003 bytes long. The amount of data is calculated by the titile of the transport layer.	U8 * size_of_data
<cop>	A parameter indicating the success of the command completion. It takes the following values: 0x31 – command completed, data sent; 0x32 – command cannot be completed; the server is not configured; 0x33 – command cannot be completed; server is server is not available; 0x34 – command cannot completed, server команда не может быть выполнена, returned an error, or does not respond to the message.	U8

Message	<HEAD>*>U2S<data> (от устройства)	
Response	<HEAD>*<U2S<cop>	
Symbols	**Transcription**	**Data format**
<HEAD>	16 bytes NTCB package title 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 from 1 to 1003 bytes long. The amount of data is calculated by the title of the transport layer.	U8 * size_of_data
<cop>	A parameter indicating the success of the command completion. It takes the following values: 0x31 – command completed, data received; 0x34 — reception error.	U8

Command to transfer data from the server via USB

Command	<HEAD>*!S2U<data>	
Response	<HEAD>*@S2U<cop>	
Symbols	**Transcription**	**Data format**
<HEAD>	16 bytes NTCB package title 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 from 1 to 1003 bytes long. The amount of data is calculated by the title of the transport layer.	U8 * size_of_data
<cop>	A parameter indicating the success of the command completion. It takes the following values: 0x31 – command completed, data sent; 0x33 - command cannot be completed, USB is not connected; 0x34 – command cannot be executed, host returned mistake or does not response.	U8
<HEAD>	16 bytes NTCB package title with preamble	16*U8

Message	<HEAD>*>S2U<data> (from device)	
Response	<HEAD>*<S2U<cop>	
Symbols	**Transcription**	**Data format**
<HEAD>	16 bytes NTCB package title 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 from 1 to 1003 bytes long. The amount of data is calculated by the title of the transport layer..	U8 * size_of_data

<cop>	A parameter indicating the success of the command completion. It takes the following values: 0x31 – command completed; 0x34 — reception error.	U8

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

Message (from device)	~P<module><put_id><data_length><data><crc8>	
Positive response (from server):	~M<module><put_id><crc8>	
Negative response (from server):	~S<module><put_id><result><crc8>	
Symbols	**Transcription**	**Data format**
~P	0x7E 0x50	2*U8
~S	0x7E 0x53	2*U8
~M	0x7E 0x4D	2*U8
<module>	Module code: 0x7F – ретранслятор RS232/RS485/USB	U8
<put_id>	Message code: 0x00 – data transfer in the transparent mode from USB. 0x01 – data transfer in the transparent mode from RS232. 0x02 – data transfer in the transparent mode from RS485. **Notion:** Device with previous firmware versions (up to 7.00.00) supports only 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>	A parameter indicating the success of the command completion. It takes the following values: 0x31 – successful command completion; 0x34 – command cannot be completed, reception error.	U8
<crc8>	Checksum. See Appendix B. CRC8 checksum calculation algorithm	U8

Data transfer command from the server to the USB, RS232, RS485 of the device

Message (from device)	~P<module><put_id><data_length><data><crc8>	
Positive response (from server):	~M<module><put_id><crc8>	
Negative response (from server):	~S<module><put_id><result><crc8>	
Symbols	**Transcription**	**Data format**
~P	0x7E 0x50	2*U8
~S	0x7E 0x53	2*U8
~M	0x7E 0x4D	2*U8
<module>	Module code: 0x7F – Radio repeater RS232/RS485/USB	U8
<put_id>	Message code: 0x00 – data transfer in the transparent mode from USB. 0x01 – data transfer in the transparent mode from RS232. 0x02 – data transfer in the transparent mode from RS485. **Notion:** Device with previous firmware versions (up to 7.00.00) supports only 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>	A parameter indicating the success of the command completion. It takes the following values: 0x30 or 0x31 – successful completion of the command; 0x33 – command cannot be completed, USB, RS232, RS485 not connected; 0x34 - command cannot be completed, reception error; 0x35 - command cannot be completed, the previous command is being processed.	U8
<crc8>	Checksum. See Appendix B. CRC8 checksum calculation algorithm	U8

User command

Message	<HEAD>*!UC<s><i>,<msg>	
Response from the	<HEAD>*@UC<s><i>,<msg>	
Notion	Sending via SMS channel is possible	
Symbols	**Transcription**	**Format**
<HEAD>	16 bytes NTCB package title 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», – «RS232», – «GPS».	char[]
<msg>	Package as a HEX string (ASCII). For example: 3101066C	char[]

2.13. Operation with CAN-LOG module

Setting the program number of the CAN-LOG device

Message	<HEAD>*!CANLOG<ver>	
Response from the server	<HEAD>*@CANLOG<cop>	
Notion	Sending via SMS channel is possible	
Symbols	**Transcription**	**Data format**
<HEAD>	16 bytes NTCB package title 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>	Version of needed CAN-LOG program in the text view For example, «123»	char[]
<cop>	Result of execution of program change operation: 0x31 – program is installed; 0x32 – incorrect program designation; 0x33 – module is not responding; 0x34 – device is not configured to operate with CAN-LOG.	U8

Request CAN-LOG device program number

Message	<HEAD>*?CANLOG	
Response from the server	<HEAD>*#CANLOG<s><result>[,<ver>]	
Notion	Sending via SMS channel is possible	
Symbols	**Transcription**	**Data format**
<HEAD>	16 bytes NTCB package title 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>	Version of needed CAN-LOG program in the text view For example, «123»	char[]
<result>	Result of execution of program change operation: 0x31 – executed; 0x33 – module is not responding; 0x34 – device is not configured to operate with CAN-LOG.	U8

2.14. Operation with built-in accelerometer

Accelerometer Calibration Command

Message	<HEAD>*!ACL_C:<command><param>	
Response from the server	<HEAD>*@ACL_C:<command><result>	
Symbols	**Transcription**	**Data format**
<HEAD>	16 bytes NTCB package title 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) – accelerometer calibration (calibration verification).	U8
<param>	Additional parameter, explaining the operation of the command. 0x30 – accelerometer calibration start; 0x31 – accelerometer calibration verification start; 0x32 – cancel previously started calibration; 0x33 – cancel accelerometer calibration according to GNSS; 0x34 – delete the current calibration.	U8
<result>	Command execution result. 0x30 – unknown command; 0x31 – command executed successfully; 0x32 – error: vehicle was moving or vibrating strongly; 0x33 – dynamic calibration is started; 0x34 – error: vehicle did not move; 0x35 – calibration completed, check the accuracy of direction detection; 0x36 – error: insufficient number or/intensity of accelerations/ braking; 0x37 – dynamic verification started; 0x39 – calibration performed successfully; 0x3A – error: failed to measure gravity; 0x3B – calibration according GNSS started.	U8

2.15. Traffic accident fixation

Request for information about fixed accident

Message	<HEAD>*?KRAI	
Response from the server	<HEAD>*#KRAI<time><size><flags><filename>	
Symbols	**Transcription**	**Data format**
<HEAD>	16 bytes NTCB package title with preamble	16*U8
*?KRAI	0x2a 0x3f 0x4b 0x52 0x41 0x49	U8[6]
*#KRAI	0x2a 0x23 0x4b 0x52 0x41 0x49	U8[6]
<time>	UTC of accident fixation in unix time format. 0 – accident is not fixed.	U32 (little-endian)
<size>	Size of file with information about accident. 0 – file is not generated.	U32 (little-endian)
<flags>	Protection mode against overwrite: 0 – protection mode is off; Other values - protection mode is on.	U8
<filename>	The name (including extension) with which file Hs to be saved. Contains terminal zero at the end.	char[] – the size is calculated by the header of the transport layer

Command to reset protection against overwrite

Message	<NTC HEAD>*!KRAI<key>	
Response from the server	<NTC HEAD>*@KRAI<result>	
Symbols	**Transcription**	**Data format**
<HEAD>	16 bytes NTCB package title with preamble	16*U8
*!KRAI	0x2a 0x21 0x4b 0x52 0x41 0x49	6*U8
*@KRAI	0x2a 0x40 0x4b 0x52 0x41 0x49	6*U8
<key>	Key to reset information about accident. Key is equal to value of block "Checksum" from the file.	U32 (little-endian)
<result>	Command execution result: 0x31 – command is executed successfully; 0x32 – operation with external memory error; 0x33 – invalid key for reset is specified; 0x35 – no accident data, no reset required.	U8

Format of file which stores information about accident

<table>
<tr><th>Block file</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. Traffic accident fixation time</td><td></td><td></td></tr>
<tr><td>4.1 Traffic accident fixation time</td><td>uint32_t</td><td>UTC time of accident fixation in unix time format</td></tr>
<tr><td>4.2. High precision timer readings</td><td>uint16_t</td><td>High precision timer readings in the moment of accident fixation (0…65535)</td></tr>
<tr><td>5. Navigation information</td><td></td><td></td></tr>
<tr><td>5.1. GPS / GLONASS navigation sensor status</td><td>uint8_t</td><td>Bit field:
<table>
<tr><th>Digit</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 – invalid navigation;
1 – valid navigation.</td></tr>
<tr><td>2..7</td><td>Number of navigation satellite 0-31</td></tr>
</table></td></tr>
<tr><td>5.2. Time</td><td>uint32_t</td><td>Unix time of the last valid navigation</td></tr>
<tr><td>5.3. Latitude</td><td>uint32_t</td><td>Latitude angle, fixed upon receipt of 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, fixed upon receipt of the last valid coordinates. In ten thousandths of a minute.</td></tr>
<tr><td>5.5. Height</td><td>int32_t</td><td>Altitude relative to sea level, recorded upon receipt of the last valid coordinates. In decimetres.</td></tr>
<tr><td>5.6. Speed</td><td>float</td><td>Speed, recorded when receiving the last valid coordinates. In km / h</td></tr>
<tr><td>5.7. Rate</td><td>uint16_t</td><td>The course recorded when receiving the last valid coordinates. 0 ° ... 360</td></tr>
<tr><td>6. Accelerometer state</td><td></td><td></td></tr>
<tr><td>6.1. Calibration current mode</td><td>uint8_t</td><td>0xAA – accelerometer is calibrated at offset;
0XBB – accelerometer is calibrated along the axles;
Rest values – accelerometer is not calibrated.</td></tr>
<tr><td>6.2. Accelerometer basic</td><td></td><td></td></tr>
<tr><td>6.2.1. Ось X</td><td>int16_t[3]</td><td>Along X, Y, Z axles. Scale factor is 16 384;</td></tr>
<tr><td>6.2.2. Ось Y</td><td>int16_t[3]</td><td>Along X, Y, Z axles. Scale factor is 16 384;</td></tr>
<tr><td>6.2.3. Ось Z</td><td>int16_t[3]</td><td>Along X, Y, Z axles. Scale factor is 16 384;</td></tr>
<tr><td>6.2.4. Offset</td><td>int16_t[3]</td><td>Along X, Y, Z axles. Scale factor is 16 384;</td></tr>
<tr><td>6.3. Total number of fixed acceleration points</td><td>uint16_t</td><td></td></tr>
<tr><td>6.4. Accident fixation point in the data array from the accelerometer</td><td>uint16_t</td><td></td></tr>
<tr><td>6.5. Acceleration along axles</td><td>struct
{
int16_t x;
int16_t y;
int16_t z;
} []</td><td>Array of data from the accelerometer along the X, Y, Z axes in m / s2 with a scale factor of 128.</td></tr>
<tr><td>7. Data about speed</td><td></td><td></td></tr>
<tr><td>7.1. Total number of fixed speed points</td><td>uint16_t</td><td></td></tr>
<tr><td>7.2. Accident fixation point in the speed data array</td><td>uint16_t</td><td></td></tr>
<tr><td>7.3. Data about speed (validity of the point and speed value)</td><td>struct
{
uint8_t valid;
uint8_t speed;
} []</td><td>Speed data array:
valid - validity coordinate point (1 - valid, 0 - invalid);
speed - km/h at a point.</td></tr>
</table>

8. Check sum	uint32_t	CRC32 checksum with poly 0x77073096, initial value 0xFFFFFFFF and without resault inverting. Calculation is performed for blocks 4…7.

3. AES128 Encryption

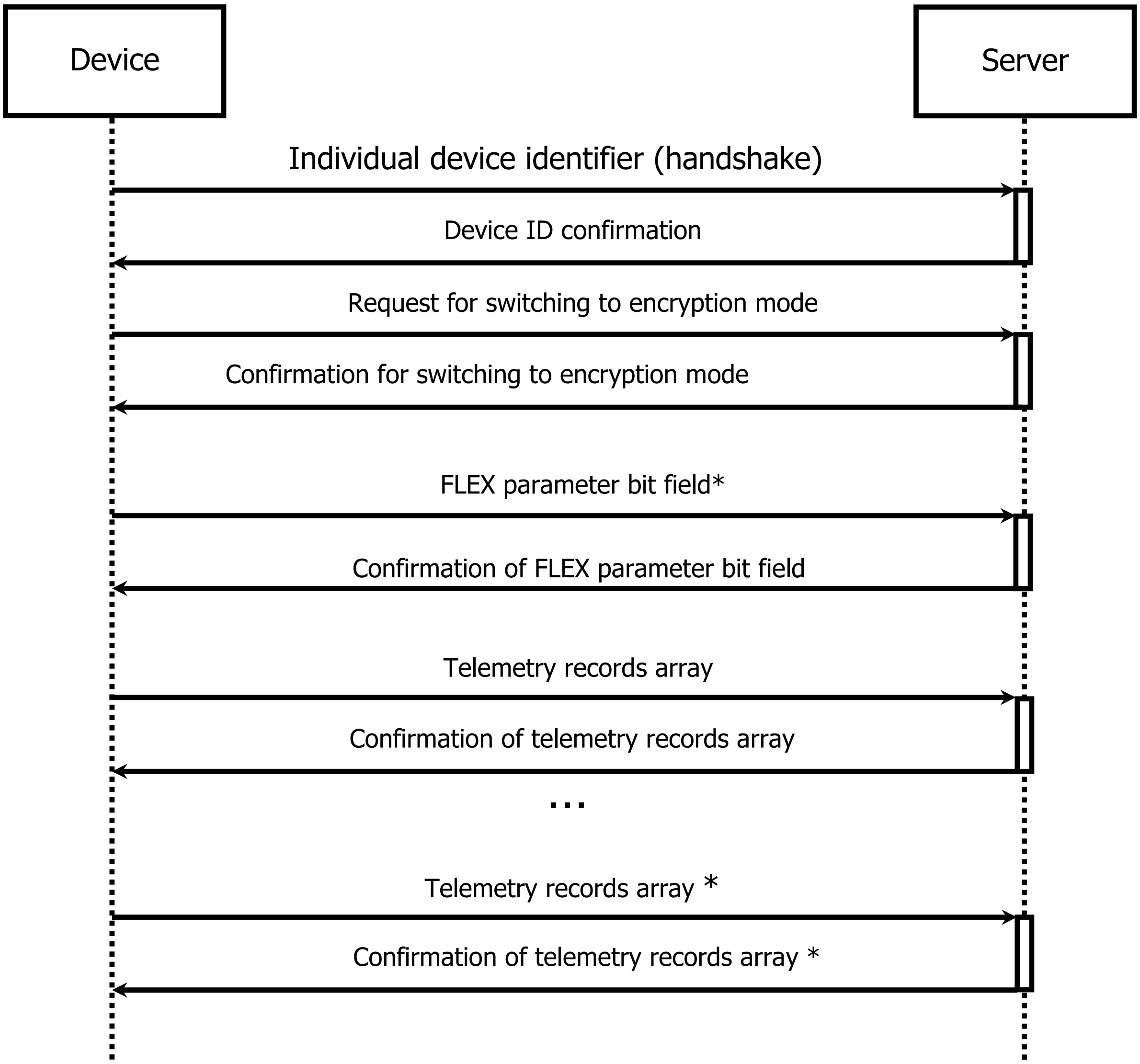
The device uses the Advanced Transcription Standard (AES) algorithm in the mode of coupling blocks of ciphertext (Cipher Block Chaining, CBC) with a key of 128 bits in size. The PKCS7 algorithm described in RFC 5652 is used as an algorithm for adding data to a multiple of the size of the Transcription block: the addition is performed by bytes, each of which is equal to the number of bytes that need to be added to the open data.

Basic definitions

Definition	Value
Device key	A cryptic clue, which is used for data encryption, which are transmitted both from the device and from the server. This key is generated with help of the server key from the IMEI device, supplemented with a '*' at the end and encoded in ASCII.
Initialization vector	Vector used for CBC algorithm initialization. This vector is generated by the device itself when connected to the server and every 12 hours if the device remains connected. An initialization vector is generated based on 4 random numbers received from a random number generator (RNG).

Procedure for connecting the device to the server with turning on Encription

After opening the connection, the device sends an unencrypted handshake package. In response, the server either sends the corresponding unencrypted response if the device is registered in the database or disconnects.
After handshake procedure was completed successfully, the device sends to the server an unencrypted command to switch to the Transcription mode, in which the required Transcription mode and the initialization vector of the blocking algorithm for blocking data blocks are indicated.
When the command to switch to Transcription mode was received, the server must confirm support of the specified Transcription mode by sending an unencrypted response. If the server does not confirm Transcription, it is necessary to disconnect from the device.
If the server confirms support of the required Transcription, the device starts to transmit data only in the encrypted form in special package-container. But the procedure of connection remains the same.


Device
Server
Individual device identifier (handshake)
Device ID confirmation
Request for switching to encryption mode
Confirmation for switching to encryption mode
FLEX parameter bit field*
Confirmation of FLEX parameter bit field
Telemetry records array
Confirmation of telemetry records array
...
Telemetry records array *
Confirmation of telemetry records array *

* are transferred in the encrypted form in special package-container

Command to switch to encription mode

Command	<HEAD>*>CODE<mode><vector>	
Response	<HEAD>*<CODE<mode>	
Symbols	**Transcription**	**Data format**
<HEAD>	16 bytes NTCB package title 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: Transcription mode: 0x00 – no Transcription; 0x01 – AES128 CBC. In response: Value from the command if server supports required Transcription, 0- if the server does not support required Transcription.	U8
<vector>	Vector of initialization with Transcription machine. Is generated by the device.	N*U8 N – vector length, is determined by the titile of transport layer.

Package-container for encrypted data transmission

Container	#<count><blocks><crc16>	
Symbols	**Transcription**	**Data format**
#	0x23	U8
<count>	Number of blocks transferred minus 1.	U8
<blocks>	Encrypted data blocks.	<blocks_count> × N × U8 N – Size of Transcription block. For AES128 it is equal to 16.
<crc16>	Check sum. Polynomial 0x1021, initial value 0. See Appendix C. CRC16 Checksum Algorithm	U16 (little-endian)

4. NTCT Text Protocol

NTCT text messages are used for transmission information via SMS. The format of a standard SMS message depends on the telemetry recording structure used in the device.

4.1. Formats of SMS-messages, coming from the device

For records F2, F5, F5.1, F5.2, F6, FLEX, the message format is M: 111

№ of line	Content of the line	Transcription
1	M:111	Message type
2	_ _ _ _ _ _ _ _ _ _ _	Type of occurred event, see file "Telemetry events code table".
3	ЧЧ.ММ.СС	UTC Event time
4	ДД/ММ/ГГ	UTC Event date
5	G:X	Operation mode where X: 0 «observation»; 1 «security»; 2 «service mode».

6	I:XXXXXXXX	The status of the inputs at the time the event was recorded in the black box. From I1 to I8 from left to right
		X – not used;
		Y – activation;
		N – no activation;
		L – blocked by command.

7	O:XXXX	The status of the inputs at the time the event was recorded in the black box. From O1 to O4 from left to right. X – is not used, Y – ON , N – OFF.
8	AK:XX.X XX.X	The voltage at the inputs of the main and backup power in volts (with tenths)
9	T:ZXX	Temperature in Celsius Z - sign '+' or '-', XX - value
10	AN:XX.X XX.X	The field is reserved for compatibility with the E-1111 device. Only zero values will be present in it.
11	HH.MM.SS	Time to get the latest valid UTC coordinates
12	DD/MM/YY	Date of receipt of the last valid coordinates by UTC
13	NXXX XX.XXXX	Latitude in degrees, minutes and fractions of minutes. N - north latitude; S - south 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	Black box write index (in hexadecimal notation)

Format of standard SMS-message M:100

№ of line	Content of the line	Notion
1	M:100	Message type
2	<vendor>	Manufacturing firm
3	X-XXXX	Device model string (6 symbols)
4	Software version:	“Firmware” version
5	XX.XX.XX	Version number
6	XX.XX.XX	Version data
7	XX	Localization (RU – Russian version, DE – german version)

Standard SMS-message format M: 101

№ of line	Content of the line	Notion
1	M:101	Message type
2	<text>	Text of the message generated by the mobile operator in response to a USSD request

Standard SMS-message format M:104

№ of line	Content of the line	Notion
1	M:104	Message type

2	<MODE>	Current operation mode of the device:
		NOGUARD – observation mode;
		GUARD – security mode;
		GUARD2 – additional security mode 1;
		GUARD3 – additional security mode 2;

3	<NAME>	Character-numerical name:
		IX – inputs, where X = 1…8;
		OX – outputs, where X = 1…4;
		UG – main power source;
		UR – reserve power source;
		T – temperature;
		A1 – voltage at analog input I7 / A1;
		A2 – voltage at analog input I8 / A2.

4	<STATE>	State: LOCKED – blocked 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 a short circuit on the line; <numerical value> - for voltages and temperature; NOT DEFINED – if parameter is unknown or entered incorrectly.

Standard SMS-message format M:105

№ of line	Content of the line	Notification
1	M:105	Message type

2	<>	Phone list type;
		PVD – list of voice alert subscribers;
		PST - list of standard SMS alerts subscribers;
		PU - list of user SMS alert subscribers.

3	<phone alias 1>:<flag>	Name of the first subscriber and notification state

4	< phone alias 2>:<flag>	Name of the second subscriber and notification state
5	<псевдоним телефона 3>:<флаг>	Name of the third subscriber and notification state
6	<псевдоним телефона 4>:<флаг>	Name of the fourth subscriber and notification state
7	<псевдоним телефона 5>:<флаг>	Name of the fifth subscriber and notification state

Standard SMS-message format M:106

№ of line	Content of the line	Notification
1	M:106	Message type
2	Reset device	Response to the RESET command Device will be rebooted

Standard SMS-message format M:107

№ of line	Content of the line	Notification
1	M:107	Message type
2	Firmware OK Firmware <string> error <error code> <ip>:<port> not responding Start connect to <ip>:<port>	Confirmation that the firmware was downloaded successfully through the RFU service and it will be rewritten. Error downloading firmware. <string> - srting with firmware version; <error code> - code detection error. specified server and port does not respond to connection requests. command to update the firmware was received, an attempt to establish a connection with the specified server is performing. <ip> - RFU IP address specified in the command; <port> - RFU IP port specified in the command.

Standard SMS-message format M:112

№ of line	Content of the line	Transcryption
1	M:112	Message type
2	HH.MM.SS	UTC Event time
3	DD/MM/YY	UTC Event date
4	CX,MCC:Y,MNC:Y, CID:Y, LAC:Y,R:Z	X – station serial number Y – station identification parameters Z – dBM signal weakening to the station
5	CX,MCC:Y,MNC:Y, CID:Y, LAC:Y,R:Z	X – station serial number Y – station identification parameters Z – dBM signal weakening to the station
6	CX,MCC:Y,MNC:Y, CID:Y, LAC:Y,R:Z	X – station serial number Y – station identification parameters Z – dBM signal weakening to the station

Standard SMS-message format M:114

№ of line	Content of the line	Notification
1	M:114	Message type
2	X-XXXX	Device model string (6 symbols)
3	XX.XX.XX	Version number
4	IMEI	Identification number of satellite navigation equipment.
5	HH.MM.SS	UTC Event time
6	DD/MM/YY	UTC Event date
7	…	Googlemap hyperlink

SMS-message format in compliance with State standart P 56361-2015

№ of line	Content of the line	Notification
1	IMEI	Identification number of satellite navigation equipment.
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 – north latitude; S – south latitude.
4	EXXX XX.XXXX	Longitude in degrees, minutes, and fractions of minutes. E - east longitude; W - west longitude.
5	G:XXXX	Height above sea level in decimeters.
6	XXX	Speed in km/h.
7	XXX	Course in degrees (from 0 to 359).
8	HH.MM.SS	Time of receiving the last valid UTC coordinates
9	DD/MM/YY	Date of receiving the last valid UTC coordinates

10	I:XXXXXXXX	State of the input lines at the time the event was recorded in the black box. From I1 to I8 from left to right.
		X – is not used;
		Y – activation;
		N – no activation;
		L – blocked by command.

4.2. SMS-requests and commands

Via the SMS channel it is possible to request information from the black box of the system, issue commands and standard requests.

4.2.1. System commands and requests

Requests

№	Request text	Request description	Response message
1	*?V	Model and version request	*#V:<n>:<v1>.<v2>.<v3>:<d>.<m>.<y>:<loc> <n> - Device model string (6 symbols). <v1>.<v2>.<v3> - software version indices (2 symbols). <d>.<m>.<y> - Accordingly, the day, month and year of this software version 2 characters. <loc> - Firmware version language (2 symbols) (RU, DE, EN).

2	B	SIM account balance request	M:101
3	*?USSD<s><code>	Arbitrary USSD request from the device with content <code>. <s> - Parameter separator– space (0x20).	*#USSD<s><string> <string> - mobile operator response line. <s> - Parameter separator– space (0x20).
4	*?VGPS	Request for information about GPS receiver firmware version	*#VGPS<s><n>,<v1>.<v2>.<v3>,<d>.<m>.<y>,<gps_ver> <s> - Parameter separator– space (0x20). <n> - Device model string (6 symbols). <v1>.<v2>.<v3> - software version indices (2 symbols). <d>.<m>.<y> - Accordingly, the day, month and year of this software version 2 characters. <gps_ver> - string with model and version of the navigation receiver.
5	*?ICCID	Request for - unique serial number of the SIM-card	*#ICCID<s><id> <id> - unique serial number of the SIM-card. <s> - Parameter separator– space (0x20).
6	*?ES	Request for device status	M:114

Commands

№	Text of the command	Command description	Response message
1	*!O<s><phnumber>	A command for microphone listening with a call to a phone number <phnumber>. <s> - parameter separator– space (0x20).	*#O<s><phnumber> <phnumber> - number on which dialing will be made. <s> - parameter separator– space (0x20).
2	RESET	Command to reboot the device	M:106
3	*!DEV_RESET	Command to reboot the device	*@DEV_RESET
4	*!SYNC<s><x>	Command to confirm black box synchronization with the device: <s> - Parameter separator– space (0x20) or ‘:’ (0x3A); <x> - index of the server specified in the device settings (start from 1).	M:111
5	*!CHNGSIM	Command to change SIM-card	*@CHNGSIM<s><x>-><y> where <x>, <y>: ‘1’ – SIM 1 (External), ‘2’ – SIM 2 (Internal). <s> - parameter separator– space (0x20).

4.2.2. Telemetric information

Requests

№	Request text	Request description	Response message
1	*?A	Request for the current state	M:111

2	A<x>	Request of the current state of the sensor<x>.
		<x> - alphanumeric value of the sensor in the system:
		I1-I8 – inputs;
		O1-O4 – outputs;
		UG, UR – supply voltage;
		T1-T4 – temperature.
			M:104

3	*?POS	Request for the current location in type of a hyperlink	M:114
4	*?LBS	Request for current data about the nearest stations of the mobile operator	M:112

Commands

Command to resend telemetry from the black box

№	Text of the command	Command description	Response message
1	*!REP_FL<s><srvindex>, <leftdate>[/<lefttime>], <rightdate>[/<rightdate>]	Command to resend telemetry from the black box. **<s>** - Parameter separator– space (0x20). **/** - Date and time field separator(0x2F) **<result>** - Command processing result (ASCII): «OK» - command completed; «FAIL» - command completion error. **<srvindex>** - Server index for retry in text format: '0' – on all servers; '1'..'3' – server index. **<leftdate>** - date of the left border of the requested telemetry interval in text format (in UTC): «DD.MM.YY». **<lefttime>** -***Optional parameter.*** Time of the left border of the requested telemetry interval in text format (in UTC): «DD:MM:SS». Lack of parameter equals to «00:00:00». **<rightdate>** - date of the right border of the requested telemetry interval 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): «DD:MM:SS». Lack of parameter equals to «23:59:59». For example: <HEAD>*!REP_FL 1,09.04.18/13:00:59,10.04.18/03:00:00 **or** <HEAD>*!REP_FL 1,09.04.18,10.04.18	REP_FL<s>OK – command completed successfully. REP_FL<s>FAIL – command error

Command to resend telemetry from SD-card

№	Text of the command	Command description	Response message
1	*!REP_SD<s><srvindex>, <leftdate>[/<lefttime>], <rightdate>[/<rightdate>]	Command to resend telemetry from the black box. **<s>** - Parameter separator– space (0x20). **/** - Date and time field separator (0x2F) **<result>** - Command processing result (ASCII): «OK» - command completed; «FAIL» - command completion error. **<srvindex>** - Server index for retry in text format: '0' – on all servers; '1'..'3' – server index. **<leftdate>** - date of the left border of the requested telemetry interval in text format (in UTC): «DD.MM.YY». **<lefttime>** - ***Optional parameter.*** Time of the left border of the requested telemetry interval in text format (in UTC):: «HH:MM:SS». Lack of parameter equals to «00:00:00». **<rightdate>** - date of the right border of the requested telemetry interval 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): «DD:MM:SS». Lack of parameter equals to «23:59:59». For example: <HEAD>*!REP_SD 1,09.04.18/13:00:59,10.04.18/03:00:00 **or** <HEAD>*!REP_SD 1,09.04.18,10.04.18	REP_SD<s>OK – command completed successfully. REP_SD<s>FAIL – command error

4.2.3. Output lines

Commands

№	Text of the command	Command description	Response message
1	1Y	Turn on the first output O1	M:111
2	1N	Turn off the first output O1	M:111
3	2Y	Turn off the second output O2	M:111
4	2N	Turn on the second output O2	M:111
5	3Y	Turn on the third output O3	M:111
6	3N	Turn off the third output O3	M:111
7	4Y	Turn on the fourth output O4	M:111
8	4N	Turn off the fourth output O4	M:111
9	*!SETOUT<s> <num><new_state> [,<num><new_state>]	Command to change the state of the output line: **<s>** - Parameter separator– space (0x20). **<result>** - Command processing result (ASCII): «OK» - command completed; «FAIL» - command completion error; **<num>** - number of the output line state of which is necessary to change (ASCII). Numeration starts with 1. **<new_state>** - State of the output line which has to be set (ASCII): 'Y' - ON; 'N' - OFF. **<cur_state>** - State of the output line after command completion (ASCII): 'Y' - ON; 'N' - OFF. For example: *!SETOUT 1Y,2N	*@SETOUT<s> <result>, <num><cur_state> [,<num><cur_state>] For example: *@SETOUT OK,1Y,2N or *@SETOUT FAIL,1N,2Y

4.2.4. Input lines

Commands

№	Text of the command	Command description	Response message
1	LOCK IX	Command to lock input line X	M:105
2	UNLOCK IX	Command to unlock input line	M:105

4.2.5. RCS, RFU services

Commands

In order to connect the device with RCS, RFU servers, it is necessary to send on it appropriate command via SMS.

Previous command formats:
(temporarily supported for backward compatibility)

№	Text of the command	Command description	Response message

1	UPFRM
	<IP>
	<PORT>
	<FIRMWARE>
	<APN>
		Command for connection to RFU service:
		<IP> - RFU server IP address; For example: 89.208.152.55;
		<PORT> - RFU server IP port; For example: 9100;
		<FIRMWARE> - firmware version number (LAST) for the newest;
		<APN> - mobile operator apn;
			M:107

	<LOGIN> <PASSWORD>	<LOGIN> - mobile operator login; <PASSWORD> - mobile operator password. Parameter separator is space or line break.	

2	NTC_CONNECT
	<IP>
	<PORT>
	<COMID>
	<APN>
	<LOGIN>
	<PASSWORD>
		Command for connection to RCS service:
		<IP> - RCS server IP address; For example: 89.208.152.55;
		<PORT> - RCS server IP port; For example: 8100;
		<COMID> - RCS Session ID; For example: 43644176;
		<APN> - mobile operator apn;
		<LOGIN> - mobile operator login;
		<PASSWORD> - mobile operator password.
		Parameter separator is space or line break
			M:107

New command formats:

№	Text of the command	Command description	Response message

1	*!CNCT_RFU<s><ip>,<port>,<firmware>,<apn>,<login>,<password>	Command for connection to RFU service:
		<s> - Parameter separator - space (0x20)
		<ip> - RFU server IP address; For example: 89.208.152.55;
		<port> - RCS server IP port; For example: 9100;
		<firmware> firmware version number (LAST) for the newest;
		Optional parameter:
		<apn> - mobile operator apn;
		<login> - mobile operator login;
		<password> - mobile operator password.
			M:107

2	*!CNCT_RCS<s><ip>,<port>,<commID>,<apn>,<login>,<password>	<s> - Parameter separator - space (0x20)
		<ip> - RCS server IP address; For example: 89.208.152.55;
		<port> - RCS server IP port; For example: 8100;
		<commID> - RCS Session ID; For example: 43644176;
		Optional parameter:
		<apn> - mobile operator apn;
		<login> - mobile operator login;
		<password> - mobile operator password;
			M:107

In the settings it is necessary specify the IP and PORT of the RCS server, as well as the session ID. If APN, LOGIN and PASSWORD settings were not specified, the device will use appropriate parameters from its own settings. Also, any of these fields can be omitted if they are not in the GSM settings of the mobile operator. If there is a password field, but apn and login are missing, in the SMS command in the corresponding lines it is necessary to enter a newline character or a space instead of the missing fields, i.e. the string must be empty.

4.2.6. Device operation mode

Commands

№	Text of the command	Command description	Response message
1	GY или G0	Switch to the «Security» mode	M:111
2	GN или G1	Switch to the «Observation» mode	M:111
3	*!M<s><x>	Command to switch security modes. <s> – parameter separator- space (0x20). <x> - device operation mode in which necessary to switch: ‘G’ – security; ‘O’ – observation;	*@M:<s><x>,<e>,<i> <x> - current operation mode: ‘G’ – security; ‘O’ – observation. <s> – parameter separator - space (0x20). <e> - reason not to switch to security mode: ‘1’ – security mode is disabled in the

			device configuration; '2' – timeout of prohibition to change mode is not expired; '3' – mode is on: not to switch to security mode when ignition is on; '4' – device is already in this mode; '5' – mode is on: not to switch to security if one of security sensors was activated. "S:" – optional part of the message, present only when the security is not entered by mistake No. 5 <i> - aliases for activated sensors separated by spaces. For devices S-2551, S-2333: "IN1", "IN2", "IN3", "IN4", "AIN1", "AIN2", "VOLT" – supply voltage sensor, "ENG" – engine operation sensor. For devices S-243X: "IN1", "IN2", "IN3", "IN4", "IN5", "IN6", "VOLT" – supply voltage sensor, "ENG" – engine operation sensor.

Requests

№	Request text	Request description	Response message
1	*?M	Request the current security mode.	*#M:<x> <s> – Parameter separator - space (0x20). <x> - device operation mode: 'G' – security; 'O' – observation.

4.2.7. Tachograph

Commands

№	Request text	Request description	Response message
1	TACH <space> EMAIL <space> <card_num>	Generate DDD file (data uploading from the card) with subsequent sending on e-mail. <card_num> - card number for uploading Accepted values: 1, 2.	TACH EMAIL: OK – command completed successfully. TACH EMAIL: BUSY – command cannot be completed in that moment. TACH EMAIL: NO TACHOGRAPH – device is not configured to work with a tachograph. TACH EMAIL: NOT CONNECTED – no connection with tachograph. TACH EMAIL: AUTH ERROR – failed to authorize in the tachograph. TACH EMAIL: DISK ERROR - error saving / reading data from non-volatile memory. TACH EMAIL: DDD WRONG PARAMS – DDD generation: invalid parameters are specified in the request (data, card number). TACH EMAIL: DDD WRONG STATE - DDD

			generation: tachograph cannot generate file in the current mode. TACH EMAIL: DDD NOT SUPPORTED - DDD generation: unsupported file type. TACH EMAIL: DDD NO DATA - DDD generation: no data for file generation. TACH EMAIL: DDD LOAD ERROR - DDD generation: receiving file from tachograph error. TACH EMAIL: DDD BUILD ABORTED - DDD generation: generation is interrupted (when flashing the device). TACH EMAIL: DDD BROKEN - DDD generation: file has incorrect structure (failed to generate file name). TACH EMAIL: EMAIL SEND ERROR – failed to send file to e-mail.

Requests

№	Request text	Request description	Response message
1	TACH <пробел> INFO	Receiving information about the state of tachograph cards.	TACH INFO: NO TACHOGRAPH – device is not configured on operation with tachograph TACH INFO: NOT CONNECTED – no connection with tachograph TACH INFO: <serial> 1 - <card 1 number> 2 - <card 2 number> <serial> - model and serial number <card 2 number> - card number (if it is) or "NONE", if there is no card For example: TACH: KASBI 1 - B000019483001001 2 — NONE

4.2.8. Auto-informer

Commands

№	Text of the command	Command description	Response message
1	AINF <пробел> ROUTE <пробел> <n><l>	Change of the current mode: <n> – route number; <l> - route symbol	AINF ROUTE: OK – command complete successfully. AINF ROUTE: BUSY – command cannot be completed, because previous command has not completed yet. AINF ROUTE: INVALID ROUTE – route was formed incorrectly. AINF ROUTE: NO ROUTE - route not found.

			AINF ROUTE: NO ZONES – failed to download list of geo-fences.
2	AINF <пробел> PLAY <пробел> <n>	Start sound file playback: <n> – sound file numerical identifier;	AINF PLAY: OK — command completed successfully. AINF PLAY: BUSY - command cannot be completed, because previous command has not completed yet. AINF PLAY: INVALID SOUND — failed to start sound file playback. AINF PLAY: NO SOUND – sound file not found.
3	AINF <пробел> SM <пробел> <n>	Change of the current speed mode: <n> – motion mode numerical identifier;	AINF SM: OK — command completed successfully. AINF SM: BUSY - command cannot be completed, because previous command has not completed yet. AINF SM: NO SM – speed mode not found in the list of modes.

4.2.9. Digital camera

Commands

№	Text of the command	Command description	Response message
1	DCAM <пробел> SS <пробел> <n> <пробел> <m>	To take a series of shots: <n> - number of images; <m> - pause between images (seconds). If only <n> is specified, then <m> is equal to 1 sec. If no parameter is specified, then <n> = <m> = 1.	See the table "Responses to camera control commands".
2	DCAM <пробел> EMAIL <пробел> <date>	Take an image with subsequent sending on email or to request the closest snapshot relative to <date> <date> - date of image creation, if it is not in the command, a new image will be generated. Format: HH.MM.SS <space> DD/MM/YY (in UTC). "15.30.45 17/09/15"	See the table "Responses to camera control commands".
3	DCAM<space>ON	Turn on auto shooting.	See the table "Responses to camera control commands".
4	DCAM<space>OFF	Turn off auto shooting.	See the table "Responses to camera control commands".

Requests

№	Request text	Request description	Response message
5	DCAM <пробел> INFO	Receiving information about camera.	See the table "Responses to camera control commands".

Responses to camera control commands

Response	Description
<cmd>: OK	Command completed successfully
<cmd>: BUSY	Command cannot be completed at this moment.
<cmd>: NO CAMERA	device is not configured to work with the camera
<cmd>: NOT CONNECTED	No connection with camera
<cmd>: TURNED OFF	Auto shooting disabled
<cmd>: DISK ERROR	Operation with SD-card error
<cmd>: NO PHOTO	Image not found
<cmd>: SS ERROR	Receiving image from camera error
<cmd>: EMAIL SEND ERROR	Failed to send image to e-mail

DCAM INFO: <state> <version> DIR <dir>	Response to the command to receive information about the camera:
	<state> - auto shooting mode: "ON" – turn on; "OFF" – turn off;
	<version> - camera version: "VC0706 1.00";
	<dir> - cataloque where photos are saved: "PHOTOS".

DCAM PHOTO: <date>	Response to the request the closest snapshot relative to <date> date> - date of image creation, if it is not in the command, a new image will be generated. Format: HH.MM.SS <space> (in UTC). "15.30.45 17/09/15"

Notion: <cmd> - transmitted command without parameters

4.2.10. Drivers display

Commands

№	Text of the command	Command description	Response message
1	DV<message>	Sending a message that does not require confirmation. <message> - text message to the driver up to 229 characters inclusive;	DV: OK – message accepted. DV: BUSY – message is not accepted, because the previous message was not sent to the display. DV: NO DV – the device is not configured to work with the display. DV: INVALID MSG – message length = 0. DV: NOT CONNECTED – no connection with the display
2	DV!<message>	Sending a message that requires confirmation <message> - text message to the driver up to 229 characters inclusive;	

Notification that driver reads messages which require confirmation

№	Notification	Command description
1	DV:<date>	<date> - time and date when the device received a message for the driver in format HH:MM:SS DD/MM/YY

4.2.11. Operation with built-in accelerometer

Requests

№	Request text	Request description	Response message
1	CALIB <space> ACL <space> ?	Request for information about the current state of the accelerometer calibration	CALIB ACL: <msg> Request for information about the current state of the accelerometer calibration <msg> - explanatory message: • Accelerometer calibration is not supported. • Accelerometer is not calibrated; • Accelerometer is calibrated at offset; • Accelerometer is calibrated at axles.

Commands

№	Text of the command	Command description	Response message
1	CALIB <space> ACL <space> S	Command to start accelerometer axles calibration	CALIB ACL: <msg> Response to command to start accelerometer axles calibration. <msg> - explanatory message: • Accelerometer calibration is not supported. • Static calibration started. • Dynamic calibration started. • Calibration completed. • Error: failed to measure gravity. • Error: vehicle was moving (vibrated strongly). • Error: vehicle did not move. • Error: insufficient number of sharp acceleration / deceleration.
2	CALIB <space> ACL <space> C	Command to verify accelerometer axles calibration	CALIB ACL: <msg> Response to command to start accelerometer axles calibration. <msg> - explanatory message: • Accelerometer calibration is not supported. • Static calibration started. • Error: failed to measure gravity. • Error: vehicle did not move. • Dynamic calibration started. • Calibration completed correctly. • Necessary to repeat calibration
3	CALIB <space> ACL <space> R	Command to cancel previously started calibration	CALIB ACL: <msg> Response to command to cancel previously started calibration. <msg> - explanatory message: • Accelerometer calibration is not supported. • Calibration interrupted.
4	CALIB <пробел> ACL <пробел> G	Command to start accelerometer axles calibration according to GNSS data	CALIB ACL: <msg> Response to command to start accelerometer axles calibration according to GNSS data. <msg> - explanatory message: • Accelerometer calibration is not supported. • Static calibration according to GNSS data started.

			• Dynamic calibration according to GNSS data started. • Calibration completed. • Error: failed to measure gravity. • Error: vehicle was moving (vibrated strongly).
5	CALIB <space> ACL <space> E	Command to reset current calibration	CALIB ACL: <msg> Response to command to reset current calibration. <msg> - explanatory message: • Calibration reset.

4.2.12. Data exchange between external interfaces

Commands

№	Text of the command	Command description	Response message
1	*!UC<s><i>,<msg>	Command to send data to the external interface, where: <i> - иinterface: 1. "RS485", 2. "RS232", 3. "GPS". <msg> - HEX string package. <s> - parameter separator- space (0x20).	*@UC<s><i>,<msg> where: <i> - interface: 1. "RS485", 2. "RS232", 3. "GPS". <msg> - HEX string package. <s> - parameter separator- space (0x20).

4.3. SMS-configuration

4.3.1. Description and principles of the commands

SMS-configuration allows to change the device configuration remotely without a computer. SMS structure depends on device configuration structure. The parameters are accessed according to the scheme: page-> tag (parameter, parameter, parameter). Parameters are arranged in strict order in accordance with the configuration structure. If the SMS command is incomplete, then the filled fields will be replaced, and empty (two commas in a row) will be skipped. If the symbol '!' is entered instead of the parameter in SMS, then the parameter in the device configuration is reset. Parameters are specified completely to the last one that needs to be changed. Each correct SMS is accompanied by a reboot with the sending of a response message.

№	Text of the command	Command description	Response message
1	*!EDITS<s><p>:<t> (<a1>,<a2>,…<aX>)	Command to edit settings with saving. Команда редактирования настроек с сохранением. Initiate a reboot of the device. <s> - parameter separator- space. <p> - name of the page on which tag is located. It has to ne ended with the following symbol **':'**. <t> - tag name which has to be edited. It has to ne ended with the following symbol **'('**. <aX> - separator of parameters which are recorded in the device. They are located in order which are determined by the configuration structure.	*@EDITS<s>OK
2	*!READ<s><p>:<t>	Request for the current parameters values. <s> - parameter separator- space. <p> - name of the page on which tag is located. It has to ne ended with the following symbol **':'**. <t> - tag name which has to be read.	*@READ<s><p>:<t>(<a1>,<a2>,…<aX>) <p> - name of the page on which tag is located. <t> - name of read tag. <aX> -value of the parameters which are read from the device. They are located in order which are determined by configuration structure.

SMS format supports accessing several pages and tags in one SMS, for this you need to write the name of the next page separated by a comma. See SMS examples.

SMS examples:

```
*!EDITS TRANS:OBJECT(object1),SRV1(FLEX,,,89.208.152.54,5100),AP1(internet.mts.ru,mts,mts)
*!READ TRANS:OBJECT,SRV1,AP1
```

4.3.2. Configuration structure

Parameter	Value	Type	Reset
Page: **TRANS**	Parameters which determine data transfer order.		
Tag: **OBJECT**	Parameter object.		
name	Object name (up to 64 symbols). By default "NONAME".	Char[24]	Yes
Тэг: **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" – with authorization. By default: "FLEX".	Char[8]	Yes Yes
objID	Object identifier. By default: 0.	U32	Yes
dcID	Dispatch center identifier (personal account number). By default: 1.	U32	Yes
addr	IP-address или domain server name. For example: "89.208.152.54". By default: "0.0.0.0" (This value indicates that the server is not in use.).	Char[35]	Yes
port	Port. For example: "5100". By default: "0".	Char[7]	Yes
transProto	Transport layer protocol: – "TCP", – "UDP". By deafult: "TCP".	Char[3]	Yes
Tag: **SRV2**	The 2-nd server parameter.		
See tag: SRV1			
Tag: **SRV3**	The 3-rd server parameter.		
See tag: SRV1			
Tag: **AP1**	Access point settings for SIM1.		
name	Name of the access point of the mobile operator. For example: "internet.mts.ru" By default: "". An empty value in the parameters name, login, pass is a sign that the "Automatic" operating mode has been selected for the access point.	Char[30]	Yes
login	Mobile Operator Login. For example: "mts" By default: "". An empty value in the parameters name, login, pass is a sign that the "Automatic" operating mode has been selected for the access point.	Char[20]	Yes
pass	Mobile Operator password. For example: "mts" By default: "". An empty value in the parameters name, login, pass is a sign that the "Automatic" operating mode has been selected for the access point.	Char[20]	Yes
Тэг: **AP2**	Access point settings for SIM2.		
See tag: AP1			

4.3.3. Previous command formats for SMS configuration

This section contains old previous formats that are temporarily supported for backward compatibility.

Commands

№	Text of the command	Command description	Response message
1	SET.<x>.<y> <IP> <port> <IDo> <IDc> <apn> <login> <password> <protocol>	Records to the device the basic parameters which are necessary for working with the server: <x> - number of the server from 1 to 3, if not specified, the settings are edited for the first server; <y> - SIM card number from 1 to 2, if not specified the settings are edited for the first SIM-card; <IP> - IP-address or domain name of the server. For example: 90.156.232.36; <port> - port. For example: 4000; <IDo> - object identification number; <IDc> - personal account number; <apn> - name of the access point of the mobile operator. For example: internet.mts.ru; <login> - mobile operator login. For example: mts; <password> - mobile operator password. For example: mts; < protocol > - exchange with the server protocol. Accepted values: “F6”; “F5.2”; “FLEX”; “EGTS” – without authorization; “EGTS_A” – EGTS with authorization. The separator between the parameters is a space or line break.	Contains the current settings of the device: CUR.<x>.<y> - message type; <IP> - IP-address or domain name of the server; <port> - server port; <IDo> - object identification number; <IDc> - number of personal account; <apn> name of the access point of the mobile operator; <login> - mobile operator login; <password> - mobile operator password; <protocol> - exchange with the server protocol; <IMEI> - unique object number. Received after rebooting. Where: <x> - server number from 1 to 3, if not stated, the settings are edited for the first server; <y> - SIM-card number from 1 to 2, if not stated, the settings are edited for the first card;
2	SET1.<x>.<y> <IP> <port> <IDo> <IDc>	Records to the device the basic parameters which are necessary for working with the server: <x> - number of the server from 1 to 3, if not specified, the settings are edited for the first server; <y> - SIM card number from 1 to 2, if not specified, the settings are edited for the first SIM-card; <IP> - IP-address or domain name of the server. For example: 90.156.232.36; <port> - port. For example: 4000; <IDo> - object identification number; <IDc> - personal account number. The separator between the parameters is a space or line break.	Contains the current settings of the device: CUR1.<x>.<y> - message type; <IP> - IP-address and domain name of the server; <port> - server port; <IDo> - object identification number; <IDc> - number of personal account; <IMEI> - unique object number. Received after rebooting. Where: <x> - server number from 1 to 3, if not stated, the settings are edited for the first server; <y> - SIM-card number from 1 to 2, if not stated, the settings are edited for the first card;
3	SET2.<x>.<y> <apn> <login> <password> <protocol>	Records to the device the basic parameters which are necessary for Internet access (GPRS setting): <x> - number of the server from 1 to 3, номер сервера от 1 до 3, if not specified, the settings are edited for the first server; <y> - SIM card number from 1 to 2, if not	Contains the current settings of the device CUR2.<x>.<y> - message type; <apn> - name of the access point of the mobile operator; <login> - mobile operator login; <password> - number of personal account; <protocol> - exchange with the server protocol

		specified, the settings are edited for the first SIM-card;
		<apn> - name of the access point of the mobile operator. For example: internet.mts.ru;
		<login> - mobile operator login. For example: mts;
		<password> - mobile operator password. For example: mts;
		<protocol> - exchange with the server protocol.
		Accepted values:
		"F6";
		"F5.2";
		"FLEX";
		"EGTS" – without authorization;
		"EGTS_A" – EGTS with authorization.
		The separator between the parameters is a space or line break.
			Received after rebooting.
			
			Where:
			<x> - server number from 1 to 3, if not stated, the settings are edited for the first server;
			<y> - SIM-card number from 1 to 2, if not stated, the settings are edited for the first card;

4	SETTM<num> <address> <mode> <nick_name>	Edits TM key parameters (the page with keys should already be in the device): <num> - key number in the configuration (2 symbols, 01-16, for example "01"); <address> - key address (12 symbols); <mode> - key operation mode (5 symbols): «NOACT» – no action; «GUARD» – changes operation mode; «IMMOB» – immobilizer. <nick_name> - key alias (10 symbols, not provided for extended key storage format); The separator between the parameters is a space or line break..	If the command was completed successfully: SETTM OK. If an error is present in the command, a standard text response will be generated for the SET command.

Requests

№	Text of the command	Command description	Response message
1	GET.<x>.<y>	Receive current parameters from the device which are required for operation with the server and which are generated in a template.	Contains the current settings of the device: CUR.<x>.<y> - message type; <IP> - IP-address or domain name of the server; <port> - server port; <IDo> - object identification number; <IDc> - personal account number; <apn> - name of the access point of the mobile operator; <login> - mobile operator login; <password> - mobile operator password; <protocol> - exchange with the server protocol; <IMEI> - unique object number Where: <x> - server number from 1 to 3, if not stated, the settings are edited for the first server; <y> - SIM-card number from 1 to 2, if not stated, the settings are edited for the first card;

2	GET1.<x>.<y>	Receive current parameters from the device which are required for operation with the server and which are generated in a template.	Contains the current device settings:
			CUR1.<x>.<y> - message type;
			<IP> - IP-address and server domain name;
			<port> - server port;
			<IDo> - object identification number;

			<IDc> - personal account data; <IMEI> - unique object number. Where: <x> - server number from 1 to 3, if not stated, the settings are edited for the first server; <y> - SIM-card number from 1 to 2, if not stated, the settings are edited for the first card;
3	GET2.<x>.<y>	Receive current parameters from the device which are required for operation with the server and which are generated in a template.	CUR2.<x>.<y> - message type; <apn> name of the access point of the mobile operator; <login> - mobile operator login; <password> - mobile operator password; <protocol> - exchange with the server protocol; Where: <x> - server number from 1 to 3, if not stated, the settings are edited for the first server; <y> - SIM-card number from 1 to 2, if not stated, the settings are edited for the first card;

In case if an error occurred while reading configuration SMS-command, SMS- message containing information about mistake will be generated. Settings will not be edited.

SMS response message structure:
<Title>
<Error type>

<Title> field contains fixed string "Error!"
<Error type> field can take the following values:

"Parameters not specified".

"Parameter № <Parameter number>: "has invalid length".

"Parameter № <Parameter number>: "contains invalid character".

"Parameter value <Parameter number> "exceeds the permissible size".

<Parameter number> field- serial number of the parameter from the beginning of the SMS, not including the message header.

The list of excluded characters for parameters of a general type: <,>, control characters.

For numeric parameters, all characters except numbers are prohibited.

5. Tone control

When dialing, it is possible to control the device using tone commands. The sign of completion of a command is always the "*" key (asterisk). If after command completion it is necessary to break he connection, it is necessary to enter the "#" sign (pound sign). In this case, immediately after receiving the commands, the device will break the connection. In this case, the 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 – number of a SIM-card in the device;
P – Latin letter "P", meaning a pause in the set of tone signals;

1234* – password with a sign of completion input

1 –command to activate the first controlling line O1;

# – a sign of the end of command input and disconnection.

Thus, it is possible to turn on the control line simply by selecting from the phonebook of the subscriber's phone, Turn On O1, whose phone number is written in the form: + 79999999999P1234 * 1 #, and call him. After receiving the command, the connection will be disconnected.

№	**Command (key sequence)**	**Essence of the request or command**
1	1*	To turn on the first control line O1
2	10*	To turn off the first control line O1
3	2*	To turn on the second control line O2
4	20*	To turn off the second control line O2
5	3*	To turn on the third control line O3
6	30*	To turn off the third control line O3
7	4*	To turn on the fourth control line O4
8	40*	To turn off the fourth control line O4
9	54**X***	Sensor lock with serial number **X** = 1..5
10	55**X***	Sensor lock with serial number **X** = 1..5
11	549*	Internal impact, tilt, displacement sensors lock
12	559*	Internal impact, tilt, displacement sensors unlock

Appendix A. Telemetry Record Structures

Appendix A.1. FLEX format telemetry records structure

Table contains the current structure of the telemetry package with dividing into versions.

<table>
<tr><th>№</th><th>Field record</th><th>Record item size</th><th>Data format</th><th>Received values</th></tr>
<tr><td colspan="5"><b>FLEX 1.0</b></td></tr>
<tr><td>1</td><td>Write number in nonvolatile memory</td><td>4</td><td>U32</td><td>Starts from zero, increments
at each entry. Never decreases.</td></tr>
<tr><td>2</td><td>Event code corresponding to this record</td><td>2</td><td>U16</td><td>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 starting from 1970.</td></tr>
<tr><td>4</td><td>Device status</td><td>1</td><td>U8</td><td>Bit field:
<table>
<tr><td>Digit</td><td>Values</td></tr>
<tr><td>0</td><td>Test mode
1 — test mode
0 — operation mode</td></tr>
<tr><td>1</td><td>Alarm notificatoin
1 — ON
0 — OFF</td></tr>
<tr><td>2</td><td>Alarm
1 — ON
0 — OFF</td></tr>
<tr><td>3-4</td><td>Operation mode:
0 — observation
1 — security
2 — ad. security
3 — service</td></tr>
<tr><td>5</td><td>Evacuation:
0 — not fixed
1 — fixed</td></tr>
<tr><td>6</td><td>Energy conservation mode:
0 – no;
1 – yes.</td></tr>
<tr><td>7</td><td>Accelerometer calibration:
0 — not calibrated
1 — calibrated</td></tr>
</table></td></tr>
<tr><td>5</td><td>Functional module status 1</td><td>1</td><td>U8</td><td>Bit field:
<table>
<tr><td>Digit</td><td>Values</td></tr>
<tr><td>0</td><td>0 - GSM off
1 - GSM on</td></tr>
<tr><td>1</td><td>0 - USB off
1 - USB on</td></tr>
<tr><td>2</td><td>0 – additional high-precision navigation receiver disabled
1 - additional high-precision navigation receiver connected</td></tr>
<tr><td>3</td><td>0 – the clock is not synchronized by GPS
1- the clock is synchronized by GPS</td></tr>
<tr><td>4</td><td>0 – the first SIM-card operates</td></tr>
</table></td></tr>
</table>

<table>
<tr><td></td><td></td><td></td><td></td><td><table>
<tr><td></td><td>1 – the second SIM-card operates</td></tr>
<tr><td>5</td><td>0 - no cellular network registration
1- cellular network registration</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>Functional module status 2</td><td>1</td><td>U8</td><td>Bit field:<table>
<tr><td>Digit</td><td>Value</td></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 GSM jamming
1 — GSM jamming detected</td></tr>
<tr><td>3</td><td>GNSS coordinates averaging:
0 – no;
1 – yes.</td></tr>
<tr><td>4</td><td>Accelerometer status:
0 — no errors;
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>Reserv</td></tr>
</table></td></tr>
<tr><td>7</td><td>GSM level</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 celluar network signal.</td></tr>
<tr><td>8</td><td>GPS/GLONASS navigation sensor state</td><td>1</td><td>U8</td><td>Bit field:<table>
<tr><td>Digit</td><td>Value</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 0-32</td></tr>
</table></td></tr>
<tr><td>9</td><td>Time of the valid coordinates (before the previous event)</td><td>4</td><td>U32</td><td>Number of seconds starting from 1970 г.</td></tr>
<tr><td>10</td><td>Last valid latitude</td><td>4</td><td>I32</td><td>Latitude angle recorded when receiving the last valid coordinates. In ten thousandths of a minute. For example, 55 ° 42.2389 'will be represented as 33422389</td></tr>
<tr><td>11</td><td>Last valid longitude</td><td>4</td><td>I32</td><td>Longitude angle recorded when receiving the last valid coordinates.
In ten thousandths of a minute.
For example, 37 ° 41.6063 'would be represented as 22616063</td></tr>
<tr><td>12</td><td>Last valid height</td><td>4</td><td>I32</td><td>Height relative to sea level, recorded upon receipt of the last valid coordinates:
In decimetres
For example, 205 meters will be represented as 2050 decimeters</td></tr>
<tr><td>13</td><td>Speed</td><td>4</td><td>Float</td><td>Speed recorded when receiving the last valid coordinates.</td></tr>
</table>

<table>
<tr><td></td><td></td><td></td><td></td><td>Km/h</td></tr>
<tr><td>14</td><td>Rate</td><td>2</td><td>U16</td><td>Rate recorded when receiving the last valid coordinates.
0° … 360°</td></tr>
<tr><td>15</td><td>Current mileage</td><td>4</td><td>Float</td><td>Mileage recorded at the time of the event which is calculated at the time of receipt of valid navigation data
In km..</td></tr>
<tr><td>16</td><td>Last section of the road</td><td>4</td><td>Float</td><td>Mileage calculated between this event and the previous one. (between two points on the track)
In km.</td></tr>
<tr><td>17</td><td>Total number of seconds on the last section of the road</td><td>2</td><td>U16</td><td>Total number of calculated points by navigation coordinates receiver at a rate of once per second.</td></tr>
<tr><td>18</td><td>Total number of seconds on the last section of the road according to which the mileage was detected (valid navigation)</td><td>2</td><td>U16</td><td>Total number of calculated points by navigation coordinates receiver at a rate of once per second with valid navigation data</td></tr>
<tr><td>19</td><td>Voltage at the main power source</td><td>2</td><td>U16</td><td>in millivolts
0-65535 mV</td></tr>
<tr><td>20</td><td>Redundant Power Supply Voltage</td><td>2</td><td>U16</td><td>in millivolts
0-65535 mV</td></tr>
<tr><td>21</td><td>Voltage at analog input 1 (Ain1)</td><td>2</td><td>U16</td><td>in millivolts
0-65535 mV</td></tr>
<tr><td>22</td><td>Voltage at analog input 2 (Ain2)</td><td>2</td><td>U16</td><td>in millivolts
0-65535 mV</td></tr>
<tr><td>23</td><td>Voltage at analog input 3 (Ain3)</td><td>2</td><td>U16</td><td>in millivolts
0-65535 mV</td></tr>
<tr><td>24</td><td>Voltage at analog input 4 (Ain4)</td><td>2</td><td>U16</td><td>in millivolts
0-65535 mV</td></tr>
<tr><td>25</td><td>Voltage at analog input 5 (Ain5)</td><td>2</td><td>U16</td><td>in millivolts
0-65535 mV</td></tr>
<tr><td>26</td><td>Voltage at analog input 6 (Ain6)</td><td>2</td><td>U16</td><td>in millivolts
0-65535 mV</td></tr>
<tr><td>27</td><td>Voltage at analog input 7 (Ain7)</td><td>2</td><td>U16</td><td>in millivolts
0-65535 mV</td></tr>
<tr><td>28</td><td>Voltage at analog input 8 (Ain8)</td><td>2</td><td>U16</td><td>in millivolts
0-65535 mV</td></tr>
<tr><td>29</td><td>Current values of the discrete sensors 1</td><td>1</td><td>U8</td><td>Bit fields:
<table>
<tr><th>Digit</th><th>Values</th></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 a normal state
1 —sensor activated</td></tr>
<tr><td>30</td><td>Current values of the discrete sensors 2</td><td>1</td><td>U8</td><td>Bit field:
<table>
<tr><th>Digit</th><th>Values</th></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 a normal state</td></tr>
</table>

<table>
<tr><td></td><td></td><td></td><td></td><td>1 — sensor activated</td></tr>
<tr><td>31</td><td>Current state of output 1</td><td>1</td><td>U8</td><td>Bit field:
<table>
<tr><td>Digit</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 is off
1- output is on</td></tr>
<tr><td>32</td><td>Current state of output 2</td><td>1</td><td>U8</td><td>Bit field:
<table>
<tr><td>Digit</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 is off
1- output is on</td></tr>
<tr><td>33</td><td>Pulse counter readings 1</td><td>4</td><td>U32</td><td>Number of pulses counted at the time of the event $0\text{-}2^{32}$</td></tr>
<tr><td>34</td><td>Pulse counter readings 2</td><td>4</td><td>U32</td><td>Number of pulses counted at the time of the event $0\text{-}2^{32}$</td></tr>
<tr><td>35</td><td>Frequency at analog-frequency fuel level sensor 1</td><td>2</td><td>U16</td><td>Frequency value 0-20000 Гц</td></tr>
<tr><td>36</td><td>Frequency at analog-frequency fuel level sensor 2</td><td>2</td><td>U16</td><td>Frequency value 0-20000 Гц</td></tr>
<tr><td>37</td><td>Engine hours calculated during the operation of the engine operation sensor</td><td>4</td><td>U32</td><td>$0\text{-}2^{32}$ sec</td></tr>
<tr><td>38</td><td>Fuel level measured by fuel level sensor 1 RS-485</td><td>2</td><td>U16</td><td>Value relative to the level of digital sensor
0 – 65499 - level,
65500 – 65535 – error code (see Appendix E)

<b>Notion:</b>
1. If a calibration for the sensor is loaded in the device, then 1 level unit corresponds to 0.1 l of fuel.
2. Depending on the setting, the entire range 0 - 65535 can be used for level transmission.</td></tr>
<tr><td>39</td><td>Fuel level measured by fuel level sensor 2 RS-485</td><td>2</td><td>U16</td><td>Value relative to the level of digital sensor
0 – 65499 - level,
65500 – 65535 – error code (see Appendix E)

<b>Notion:</b>
1. If a calibration for the sensor is loaded in the device, then 1 level unit corresponds to 0.1 l of fuel.
2. Depending on the setting, the entire range 0 - 65535 can be used for level transmission.</td></tr>
<tr><td>40</td><td>Fuel level measured by fuel level sensor 3 RS-485</td><td>2</td><td>U16</td><td>Value relative to the level of digital sensor
0 – 65499 - level,
65500 – 65535 - level (see Appendix E)

<b>Notion:</b>
1. If a calibration for the sensor is loaded in the device, then 1 level unit corresponds to 0.1 l of fuel.
2. Depending on the setting, the entire range 0 - 65535 can be used for level transmission.</td></tr>
</table>

<table>
<tr><td>41</td><td>Fuel level measured by fuel level sensor 4 RS-485</td><td>2</td><td>U16</td><td>Value relative to the level of digital sensor
0 – 65499 - level,
65500 – 65535 – code error (see Appendix E)

<b>Notion:</b>
1. If a calibration for the sensor is loaded in the device, then 1 level unit corresponds to 0.1 l of fuel.
2. Depending on the setting, the entire range 0 - 65535 can be used for level transmission.</td></tr>
<tr><td>42</td><td>Fuel level measured by fuel level sensor 5 RS-485</td><td>2</td><td>U16</td><td>Value relative to the level of digital sensor
0 – 65499 - level,
65500 – 65535 – error code (see Appendix E)

<b>Notion:</b>
1. If a calibration for the sensor is loaded in the device, then 1 level unit corresponds to 0.1 l of fuel.
2. Depending on the setting, the entire range 0 - 65535 can be used for level transmission.</td></tr>
<tr><td>43</td><td>Fuel level measured by fuel level sensor 6 RS-485</td><td>2</td><td>U16</td><td>Value relative to the level of digital sensor
0 – 65499 - level,
65500 – 65535 – error code (see Appendix E)

<b>Notion:</b>
1. If a calibration for the sensor is loaded in the device, then 1 level unit corresponds to 0.1 l of fuel.
2. Depending on the setting, the entire range 0 - 65535 can be used for level transmission.</td></tr>
<tr><td>44</td><td>Fuel level measured by fuel level sensor RS-232</td><td>2</td><td>U16</td><td>Value relative to the level of digital sensor
0 – 65499 - level,
65500 – 65535 – error code (see Appendix E)

<b>Notion:</b>
1. If a calibration for the sensor is loaded in the device, then 1 level unit corresponds to 0.1 l of fuel.
2. Depending on the setting, the entire range 0 - 65535 can be used for level transmission.</td></tr>
<tr><td>45</td><td>Temperature from digital sensor 1 (in Celsius)</td><td>1</td><td>I8</td><td>-55°C … +125°C
(-128 °C — sensor is not connected)</td></tr>
<tr><td>46</td><td>Temperature from digital sensor 2 (in Celsius)</td><td>1</td><td>I8</td><td>-55°C … +125°C
(-128 °C — sensor is not connected)</td></tr>
<tr><td>47</td><td>Temperature from digital sensor 3 (in Celsius)</td><td>1</td><td>I8</td><td>-55°C … +125°C
(-128 °C — sensor is not connected)</td></tr>
<tr><td>48</td><td>Temperature from digital sensor 4 (in Celsius)</td><td>1</td><td>I8</td><td>-55°C … +125°C
(-128 °C — sensor is not connected)</td></tr>
<tr><td>49</td><td>Temperature from digital sensor 5 (in Celsius)</td><td>1</td><td>I8</td><td>-55°C … +125°C
(-128 °C — sensor is not connected)</td></tr>
<tr><td>50</td><td>Temperature from digital sensor 6 (in Celsius)</td><td>1</td><td>I8</td><td>-55°C … +125°C
(-128 °C — sensor is not connected)</td></tr>
<tr><td>51</td><td>Temperature from digital sensor 7 (in Celsius)</td><td>1</td><td>I8</td><td>-55°C … +125°C
(-128 °C — sensor is not connected)</td></tr>
<tr><td>52</td><td>Temperature from digital sensor 8 (in Celsius)</td><td>1</td><td>I8</td><td>-55°C … +125°C
(-128 °C — sensor is not connected)</td></tr>
<tr><td>53</td><td>CAN
Fuel level in the tank</td><td>2</td><td>U16</td><td><table>
<tr><td>Digit</td><td>Values</td></tr>
<tr><td>0-14</td><td>If digit 15 is equal to one
0-100 in % (accuracy to1%)

If digit 15 is equal to one
0-32766 in 0,1 litres (0-3276,6 liters)
32767 (0x7FFF) – Parameter not read</td></tr>
<tr><td>15</td><td>1 — percent of the volume
0 — volume in tenths of a liter</td></tr>
</table></td></tr>
</table>

<table>
<tr><td>54</td><td>CAN
Total fuel consumption</td><td>4</td><td>Float</td><td>Litres
0 — (3,4 • 10^{38}) l
If value is negative, then the parameter is not read</td></tr>
<tr><td>55</td><td>CAN
Engine revolutions</td><td>2</td><td>U16</td><td>Revolutions per minute,
0-65534
65535 (0xFFFF) – parameter is not read</td></tr>
<tr><td>56</td><td>CAN
Coolant Temperature (Engine)</td><td>1</td><td>I8</td><td>In degree Celsius
-127 до 127 °C
-128 °C (0x80) – parameter is not read</td></tr>
<tr><td>57</td><td>CAN
Total vehicle mileage</td><td>4</td><td>Float</td><td>Kilometers
0 — (3,4 • 10^{38}) km
If value is negative, then the parameter is not read</td></tr>
<tr><td>58</td><td>CAN
Load on axle 1</td><td>2</td><td>U16</td><td>In kilograms
0-65534 кг
65535 (0xFFFF) – parameter is not read</td></tr>
<tr><td>59</td><td>CAN
Load on axle 2</td><td>2</td><td>U16</td><td>In kilograms
0-65534 кг
65535 (0xFFFF) – parameter is not read</td></tr>
<tr><td>60</td><td>CAN
Load on axle 3</td><td>2</td><td>U16</td><td>In kilograms
0-65534 кг
65535 (0xFFFF) – parameter is not read</td></tr>
<tr><td>61</td><td>CAN
Load on axle 4</td><td>2</td><td>U16</td><td>In kilograms
0-65534 кг
65535 (0xFFFF) – parameter is not read</td></tr>
<tr><td>62</td><td>CAN
Load on axle 5</td><td>2</td><td>U16</td><td>In kilograms
0-65534 кг
65535 (0xFFFF) – parameter is not read</td></tr>
<tr><td>63</td><td>CAN
Gas pedal position</td><td>1</td><td>U8</td><td>In percentage
0-100%
255 (0xFF) – parameter is not read</td></tr>
<tr><td>64</td><td>CAN
Brake pedal position</td><td>1</td><td>U8</td><td>In percentage
0-100%
255 (0xFF) – parameter is not read</td></tr>
<tr><td>65</td><td>CAN
Engine load</td><td>1</td><td>U8</td><td>In percentage
0-100%
255 (0xFF) – parameter is not read</td></tr>
<tr><td>66</td><td>CAN

Liquid Level in a Diesel Exhaust Filter</td><td>2</td><td>U16</td><td><table>
<tr><th>Digit</th><th>Value</th></tr>
<tr><td>0-14</td><td>If digit 15 is equal to one
0-100 in % (accuracy up to 1%)

If digit 15 is equal to one
0-32766 in 0,1 litres (0-3276,6 liters)

32767 (0x7FFF) – parameter is not read</td></tr>
<tr><td>15</td><td>1 — percent of the volume
0 — volume in tenths of a liter</td></tr>
</table></td></tr>
<tr><td>67</td><td>CAN
Full time of engine operation</td><td>4</td><td>U32</td><td>In seconds
0-2^{32} sec</td></tr>
<tr><td>68</td><td>CAN
Distance before technical inspection</td><td>2</td><td>I16</td><td>In kilometers
0-32767 * 5km
-1 (0xFFFF) – parameter is not read</td></tr>
<tr><td>69</td><td>CAN
Vehicle speed</td><td>1</td><td>U8</td><td>In kilometres an hour0-254
255 (0xFF) – parameter is not read</td></tr>
<tr><td colspan="5"><b>FLEX 2.0</b></td></tr>
<tr><td rowspan="2">70</td><td rowspan="2">Information about navigation</td><td rowspan="2">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>
</table>

<table>
<tr><td rowspan="6"></td><td rowspan="6"></td><td rowspan="6"></td><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 navigation module</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 navigation module</td><td>U8</td><td>0,1…25,0 (1-250, value multiplied by 10)</td></tr>
<tr><td>72</td><td>State of the additional high-precision navigation receiver</td><td>1</td><td>U8</td><td>Bit field:<table><tr><td>Digit</td><td>Value</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 doesn’t operate in RTK mode;
1 – receiver doesn’t operate 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 coordinates from a high-precision receiver</td><td rowspan="2">16</td><td>I64</td><td>Latitude angle recorded when receiving the last valid coordinates.
In destimillionths of a minute.</td></tr>
<tr><td>Longitude coordinates from a high-precision receiver</td><td>I64</td><td>Longitude angle recorded when receiving the last valid coordinates.
In ten millionths of a minute.</td></tr>
<tr><td>74</td><td>Height from high-precision receiver</td><td>4</td><td>I32</td><td>Height relative to sea level, recorded upon receipt of the last valid coordinates:
In millimeters
For example, 205 meters will be represented as 205,000 millimeters.</td></tr>
<tr><td>75</td><td>Rate from high-precision receiver</td><td>2</td><td>U16</td><td>Rate recorded when receiving the last valid coordinates.
In hundredths of a degree.
For example, 270 ° would be represented as 27,000.</td></tr>
<tr><td>76</td><td>Speed from high-precision receiver</td><td>4</td><td>Float</td><td>Speed recorded when receiving the last valid coordinates.
In km/h</td></tr>
<tr><td rowspan="9">77</td><td rowspan="5">Information about the current base station (LBS)</td><td rowspan="9">37</td><td>U32</td><td>Cell identifier (cell id)</td></tr>
<tr><td>U16</td><td>Local zone code (lac)</td></tr>
<tr><td>U16</td><td>Code of the country in which the base station is located (mcc)</td></tr>
<tr><td>U16</td><td>Cellular network code (mnc)</td></tr>
<tr><td>S8</td><td>The level of the radio signal received on this channel at the input to the telephone receiver. Measured in decibels to milliwatts (dBm). It fluctuates between approximately −35 dBm - −111 dBm.</td></tr>
<tr><td rowspan="4">Information about neighboor base station №1 (LBS)</td><td>U32</td><td>Cell identifier (cell id)</td></tr>
<tr><td>U16</td><td>Local zone code (lac)</td></tr>
<tr><td>U16</td><td>Code of the country in which the base station is located (mcc)</td></tr>
<tr><td>U16</td><td>Cellular network code (mnc)</td></tr>
</table>

			S8	The level of the radio signal received on this channel at the input to the telephone receiver. Measured in decibels to milliwatts (dBm). It fluctuates between approximately –35 dBm - –111 dBm.
	Information about neighboor base station №2 (LBS)		U32	Cell identifier (cell id)
			U16	Local zone code (lac)
			U16	Code of the country in which the base station is located (mcc)
			U16	Cellular network code (mnc)
			S8	The level of the radio signal received on this channel at the input to the telephone receiver. Measured in decibels to milliwatts (dBm). It fluctuates between approximately –35 dBm - –111 dBm.
	Time of the last received data from LBS		U32	Number of seconds starting from 1970
78	Temperature measured by fuel level sensor 1 RS485	1	I8	Temperature for FLS -128°C … +127°C
79	Temperature measured by fuel level sensor 2 RS485	1	I8	Temperature for FLS -128°C … +127°C
80	Temperature measured by fuel level sensor 3 RS485	1	I8	Temperature for FLS -128°C … +127°C
81	Temperature measured by fuel level sensor 4 RS485	1	I8	Temperature for FLS -128°C … +127°C
82	Temperature measured by fuel level sensor 5 RS485	1	I8	Temperature for FLS -128°C … +127°C
83	Temperature measured by fuel level sensor 6 RS485	1	I8	Temperature for FLS -128°C … +127°C
84	Fuel level measured by fuel level sensor 7 RS485	3	U16	Value relative to the level of digital sensor 0 – 65499 - level, 65500 – 65535 – error code (see Appendix E) **Notion:** 1. If a calibration for the sensor is loaded in the device, then 1 level unit corresponds to 0.1 l of fuel. 2. Depending on the setting, the entire range 0 - 65535 can be used for level transmission.
	Temperature measured by fuel level sensor 7 RS485		I8	Temperature value for FLS -128°C … +127°C
85	Fuel level measured by fuel level sensor 8 RS485	3	U16	Value relative to the level of digital sensor 0 – 65499 - level, 65500 – 65535 – error code (see Appendix E) **Notion:** 1. If a calibration for the sensor is loaded in the device, then 1 level unit corresponds to 0.1 l of fuel. 2. Depending on the setting, the entire range 0 - 65535 can be used for level transmission.
	Temperature measured by fuel level sensor 8 RS485		I8	Temperature value for FLS -128°C … +127°C
86	Fuel level measured by fuel level sensor 9 RS485	3	U16	Value relative to the level of digital sensor 0 – 65499 - level, 65500 – 65535 – error code (see Appendix E) **Notion:** 1. If a calibration for the sensor is loaded in the device, then 1 level unit corresponds to 0.1 l of fuel. 2. Depending on the setting, the entire range 0 - 65535 can be used for level transmission.
	Temperature measured by fuel level sensor 9 RS485		I8	Temperature value for FLS -128°C … +127°C
87	Fuel level measured by fuel level	3	U16	Value relative to the level of digital sensor

	sensor 10 RS485			0 – 65499 - level, 65500 – 65535 – error code (see Appendix E) **Notion:** 1. If a calibration for the sensor is loaded in the device, then 1 level unit corresponds to 0.1 l of fuel. 2. Depending on the setting, the entire range 0 - 65535 can be used for level transmission.
	Temperature measured by fuel level sensor 10 RS485		I8	Temperature value for FLS -128°C … +127°C
88	Fuel level measured by fuel level sensor 11 RS485	3	U16	Value relative to the level of digital sensor 0 – 65499 - level, 65500 – 65535 – error code (see Appendix E) **Notion:** 1. If a calibration for the sensor is loaded in the device, then 1 level unit corresponds to 0.1 l of fuel. 2. Depending on the setting, the entire range 0 - 65535 can be used for level transmission.
	Temperature measured by fuel level sensor 11 RS485		I8	Temperature value for FLS -128°C … +127°C
89	Fuel level measured by fuel level sensor 12 RS485	3	U16	Value relative to the level of digital sensor 0 – 65499 - level, 65500 – 65535 – error code (see Appendix E) **Notion:** 1. If a calibration for the sensor is loaded in the device, then 1 level unit corresponds to 0.1 l of fuel. 2. Depending on the setting, the entire range 0 - 65535 can be used for level transmission.
	Temperature measured by fuel level sensor 12 RS485		I8	Temperature value for FLS -128°C … +127°C
90	Fuel level measured by fuel level sensor 13 RS485	3	U16	Value relative to the level of digital sensor 0 – 65499 - level, 65500 – 65535 – error code (see Appendix E) **Notion:** 1. If a calibration for the sensor is loaded in the device, then 1 level unit corresponds to 0.1 l of fuel. 2. Depending on the setting, the entire range 0 - 65535 can be used for level transmission.
	Temperature measured by fuel level sensor 13 RS485		I8	Temperature value for FLS -128°C … +127°C
91	Fuel level measured by fuel level sensor 14 RS485	3	U16	Value relative to the level of digital sensor 0 – 65499 - level, 65500 – 65535 – error code (see Appendix E) **Notion:** 1. If a calibration for the sensor is loaded in the device, then 1 level unit corresponds to 0.1 l of fuel. 2. Depending on the setting, the entire range 0 - 65535 can be used for level transmission.
	Temperature measured by fuel level sensor 14 RS485		I8	Temperature value for FLS -128°C … +127°C
92	Fuel level measured by fuel level sensor 15 RS485	3	U16	Value relative to the level of digital sensor 0 – 65499 - level, 65500 – 65535 – error code (see Appendix E) **Notion:** 1. If a calibration for the sensor is loaded in the device, then 1 level unit corresponds to 0.1 l of fuel. 2. Depending on the setting, the entire range 0 -

				65535 can be used for level transmission.
	Temperature measured by fuel level sensor 15 RS485		I8	Temperature value for FLS -128°C … +127°C
93	Fuel level measured by fuel level sensor 16 RS485	3	U16	Value relative to the level of digital sensor 0 – 65499 - level, 65500 – 65535 – error code (see Appendix E) **Notion:** 1. If a calibration for the sensor is loaded in the device, then 1 level unit corresponds to 0.1 l of fuel. 2. Depending on the setting, the entire range 0 - 65535 can be used for level transmission.
	Temperature measured by fuel level sensor 16 RS485		I8	Temperature value for FLS -128°C … +127°C
94	Information about the 1st tyre pressure sensor	6	U8	Wheel № If there is no sensor = 0
			U8	Pressure in 0.1 bar If there is no sensor = 0
			I8	Temperature in degrees if there is no sensor = -128
	Information about the 2nd tyre pressure sensor		U8	Wheel № If there is no sensor = 0
			U8	Pressure in 0.1 bar If there is no sensor = 0
			I8	Temperature in degrees if there is no sensor = -128
95	Information about the 3rd tyre pressure sensor	12	U8	Wheel № If there is no sensor = 0
			U8	Pressure in 0.1 bar If there is no sensor = 0
			I8	Temperature in degrees if there is no sensor = -128
	Information about the 4th tyre pressure sensor		U8	Wheel № If there is no sensor = 0
			U8	Pressure in 0.1 bar If there is no sensor = 0
			I8	Temperature in degrees if there is no sensor = -128
	Information about the 5th tyre pressure sensor		U8	Wheel № If there is no sensor = 0
			U8	Pressure in 0.1 bar If there is no sensor = 0
			I8	Temperature in degrees if there is no sensor = -128
	Information about the 6th tyre pressure sensor		U8	Wheel № If there is no sensor = 0
			U8	Pressure in 0.1 bar If there is no sensor = 0
			I8	Temperature in degrees if there is no sensor = -128
96	Information about the 7th tyre pressure sensor	24	U8	Wheel № If there is no sensor = 0
			U8	Pressure in 0.1 bar If there is no sensor = 0
			I8	Temperature in degrees if there is no sensor = -128
	Information about the 8th tyre pressure sensor		U8	Wheel № If there is no sensor = 0
			U8	Pressure in 0.1 bar If there is no sensor = 0
			I8	Temperature in degrees if there is no sensor = -128
	Information about the 9th tyre		U8	Wheel №

	pressure sensor			If there is no sensor = 0
			U8	Pressure in 0.1 bar If there is no sensor = 0
			I8	Temperature in degrees if there is no sensor = -128
	Information about the 10th tyre pressure sensor		U8	Wheel № If there is no sensor = 0
			U8	Pressure in 0.1 bar If there is no sensor = 0
			I8	Temperature in degrees if there is no sensor = -128
	Information about the 11th tyre pressure sensor		U8	Wheel № If there is no sensor = 0
			U8	Pressure in 0.1 bar If there is no sensor = 0
			I8	Temperature in degrees if there is no sensor = -128
	Information about the 12th tyre pressure sensor		U8	Wheel № If there is no sensor = 0
			U8	Pressure in 0.1 bar If there is no sensor = 0
			I8	Temperature in degrees if there is no sensor = -128
	Information about the 13th tyre pressure sensor		U8	Wheel № If there is no sensor = 0
			U8	Pressure in 0.1 bar If there is no sensor = 0
			I8	Temperature in degrees if there is no sensor = -128
	Information about the 14th tyre pressure sensor		U8	Wheel № If there is no sensor = 0
			U8	Pressure in 0.1 bar If there is no sensor = 0
			I8	Temperature in degrees if there is no sensor = -128
97	Information about the 15th tyre pressure sensor	48	U8	Wheel № If there is no sensor = 0
			U8	Pressure in 0.1 bar If there is no sensor = 0
			I8	Temperature in degrees if there is no sensor = -128
	Information about the 16th tyre pressure sensor		U8	Wheel № If there is no sensor = 0
			U8	Pressure in 0.1 bar If there is no sensor = 0
			I8	Temperature in degrees if there is no sensor = -128
	Information about the 17th tyre pressure sensor		U8	Wheel № If there is no sensor = 0
			U8	Pressure in 0.1 bar If there is no sensor = 0
			I8	Temperature in degrees if there is no sensor = -128
	Information about the 18th tyre pressure sensor		U8	Wheel № If there is no sensor = 0
			U8	Pressure in 0.1 bar If there is no sensor = 0
			I8	Temperature in degrees if there is no sensor = -128
	Information about the 19th tyre pressure sensor		U8	Wheel № If there is no sensor = 0
			U8	Pressure in 0.1 bar

				If there is no sensor = 0
			I8	Temperature in degrees if there is no sensor = -128
	Information about the 20th tyre pressure sensor		U8	Wheel № If there is no sensor = 0
			U8	Pressure in 0.1 bar If there is no sensor = 0
			I8	Temperature in degrees if there is no sensor = -128
	Information about the 21st tyre pressure sensor		U8	Wheel № If there is no sensor = 0
			U8	Pressure in 0.1 bar If there is no sensor = 0
			I8	Temperature in degrees if there is no sensor = -128
	Information about the 22nd tyre pressure sensor		U8	Wheel № If there is no sensor = 0
			U8	Pressure in 0.1 bar If there is no sensor = 0
			I8	Temperature in degrees if there is no sensor = -128
	Information about the 23rd tyre pressure sensor		U8	Wheel № If there is no sensor = 0
			U8	Pressure in 0.1 bar If there is no sensor = 0
			I8	Temperature in degrees if there is no sensor = -128
	Information about the 24th tyre pressure sensor		U8	Wheel № If there is no sensor = 0
			U8	Pressure in 0.1 bar If there is no sensor = 0
			I8	Temperature in degrees if there is no sensor = -128
	Information about the 25th tyre pressure sensor		U8	Wheel № If there is no sensor = 0
			U8	Pressure in 0.1 bar If there is no sensor = 0
			I8	Temperature in degrees if there is no sensor = -128
	Information about the 26th tyre pressure sensor		U8	Wheel № If there is no sensor = 0
			U8	Pressure in 0.1 bar If there is no sensor = 0
			I8	Temperature in degrees if there is no sensor = -128
	Information about the 27th tyre pressure sensor		U8	Wheel № If there is no sensor = 0
			U8	Pressure in 0.1 bar If there is no sensor = 0
			I8	Temperature in degrees if there is no sensor = -128
	Information about the 28th tyre pressure sensor		U8	Wheel № If there is no sensor = 0
			U8	Pressure in 0.1 bar If there is no sensor = 0
			I8	Temperature in degrees if there is no sensor = -128
	Information about the 29th tyre pressure sensor		U8	Wheel № If there is no sensor = 0
			U8	Pressure in 0.1 bar If there is no sensor = 0
			I8	Temperature in degrees if there is no sensor = -128

<table>
<tr><td rowspan="3"></td><td rowspan="3">Information about the 30th tyre pressure sensor</td><td rowspan="3"></td><td>U8</td><td>Wheel №
If there is no sensor = 0</td></tr>
<tr><td>U8</td><td>Pressure in 0.1 bar
If there is no sensor = 0</td></tr>
<tr><td>I8</td><td>Temperature in degrees
if there is no sensor = -128</td></tr>
<tr><td>98</td><td>Tachograph data:

Driver activity and card slot status.</td><td>1</td><td>U8</td><td>Bit field:
<table>
<tr><td>Digit</td><td>Value</td></tr>
<tr><td>0..1</td><td>Activity of driver 1:
0 – Rest,
1 – Ready to work,
2 – Work does not connect with driving,
3 – Driving</td></tr>
<tr><td>2..3</td><td>Slot for card of Driver 1:
0 – No card,
1 – Not authorized,
2 - Authorized,
3 - Failed to retrieve</td></tr>
<tr><td>4..5</td><td>Activity of driver 2:
0 – Rest,
1 – Ready to work,
2 – Work does not connect with driving,
3 – Driving</td></tr>
<tr><td>6..7</td><td>Slot for card of Driver 1:
0 – No card,
1 – Not authorized,
2 - Authorized,
3 - Failed to retrieve</td></tr>
</table></td></tr>
<tr><td>99</td><td>Tachograph/card operation mode</td><td>1</td><td>U8</td><td>0 – Tachograph is off,
1 – Driver,
2 – Wizard,
3 – Controller,
4 - Enterprise,
5 – Carriage.</td></tr>
<tr><td>100</td><td>Status flags from the tachograph</td><td>2</td><td>U16</td><td>Bit field:
<table>
<tr><td>Digit</td><td>Value</td></tr>
<tr><td>0</td><td>Ignition:
0 — off;
1 — on.</td></tr>
<tr><td>1</td><td>DNS 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>Illumination:
0 — off;
1 — on.</td></tr>
<tr><td>5</td><td>Connection with tachograph error:
0 — no;
1 — yes.</td></tr>
<tr><td>6-8</td><td>Driver 1 operation mode detected in time:
0 – no warnings about operation time;
1 – restriction # 1: 15 minutes before 4.5 hours of continuous</td></tr>
</table></td></tr>
</table>

<table>
<tr><td></td><td></td><td></td><td></td><td><table>
<tr><td></td><td>driving;
2 - restriction # 2: exceeding 4.5 hours of continuous driving;
3 - restriction # 3: 15 min until additional Warning 1;
4 - restriction # 4: Warning 1 occurred;
5 - restriction # 5: 15 min until additional Warning 2;
6 - restriction # 6: Warning 2 occurred;
7 - reserve.</td></tr>
<tr><td>9-11</td><td>Driver 2 operation mode detected in time: See “Driver 1 operation mode detected in time”</td></tr>
<tr><td>12-15</td><td>Reserv</td></tr>
</table></td></tr>
<tr><td>101</td><td>Tachograph speed</td><td>1</td><td>U8</td><td>0-254 km/h
No value – 255.</td></tr>
<tr><td>102</td><td>Tachograph odometer</td><td>4</td><td>U32</td><td>Value in 1/10 km
No value – 0xFFFFFFFF.</td></tr>
<tr><td>103</td><td>Tachograph time</td><td>4</td><td>U32</td><td>Number of seconds starting from 1970</td></tr>
<tr><td>104</td><td>Current driver status received from the display module</td><td>1</td><td>U8</td><td>0 – driver status unknown (no display);
1 – «Driving»;
2 – «On route»;
3 – «Free»;
4 – «Waiting»;
5 – «Return»;
6 – «Reserve»;
7 – «Work»;
8 – «Pause»;
9 – «Ready»;
10 – «Dinner»;
11 – «Rest»;
12 – «Repair»;
13 – «Loading»;
14 – «Unloading»;
15 – «Breakdown»;
16 – «Accident».</td></tr>
<tr><td>105</td><td>Index of the last received / read message on the display module.</td><td>4</td><td>U32</td><td>0 – no received/ read messages;
0xFFFFFFFF – received / read message sent by NTCT command;
Rest values correspond to the index of the last received / read message</td></tr>
<tr><td>106</td><td>Increment to time relative to previous record</td><td>2</td><td>U16</td><td>0-65534 – 0.01 sec
0xFFFF – value is irrelevant, the duration is more than 655.34 seconds.</td></tr>
<tr><td rowspan="3">107</td><td>Linear acceleration at X axle</td><td rowspan="3">6</td><td>I16</td><td>-24000..+24000 (-24..+24 g multiplied by 1000)
No value: -32768</td></tr>
<tr><td>Linear acceleration at Y axle</td><td>I16</td><td>-24000..+24000 (-24..+24 g multiplied by 1000)
No value: -32768</td></tr>
<tr><td>Linear acceleration at Z axle</td><td>I16</td><td>-24000..+24000 (-24..+24 g multiplied by 1000)
No value: -32768</td></tr>
<tr><td>108</td><td>EcoDriving. Duration of threshold axceeding</td><td>2</td><td>U16</td><td>0-65534 – 0.01 of the second
0xFFFF – value is irrelevant if duration is more than 655.34 sec.</td></tr>
<tr><td rowspan="2">109</td><td>EcoDriving. Maximum value of positive acceleration for the period</td><td rowspan="2">6</td><td>I16</td><td>0..+24000 (0 … +24 multipled by 1000)</td></tr>
<tr><td>EcoDriving. Maximum value of negative (brake) acceleration for the period</td><td>I16</td><td>0..+24000 (0 … +24 multipled by 1000)</td></tr>
</table>

<table>
<tr><td></td><td>EcoDriving. Maximum value of lateral acceleration for the period</td><td></td><td>I16</td><td>0..+24000 (0 … +24 multipled by 1000)</td></tr>
<tr><td rowspan="2">110</td><td>Ridership counters sensor data 1</td><td rowspan="2">2</td><td>U8</td><td>0-254
255 – no data</td></tr>
<tr><td>Ridership counters sensor data 1</td><td>U8</td><td>0-254
255 – no data</td></tr>
<tr><td rowspan="2">111</td><td>Ridership counters sensor data 3</td><td rowspan="2">2</td><td>U8</td><td>0-254
255 – no data</td></tr>
<tr><td>Ridership counters sensor data 4</td><td>U8</td><td>0-254
255 – no data</td></tr>
<tr><td rowspan="2">112</td><td>Ridership counters sensor data 5</td><td rowspan="2">2</td><td>U8</td><td>0-254
255 – no data</td></tr>
<tr><td>Ridership counters sensor data 6</td><td>U8</td><td>0-254
255 – no data</td></tr>
<tr><td rowspan="2">113</td><td>Ridership counters sensor data 7</td><td rowspan="2">2</td><td>U8</td><td>0-254
255 – no data</td></tr>
<tr><td>Ridership counters sensor data 8</td><td>U8</td><td>0-254
255 – no data</td></tr>
<tr><td rowspan="2">114</td><td>Ridership counters sensor data 9</td><td rowspan="2">2</td><td>U8</td><td>0-254
255 – no data</td></tr>
<tr><td>Ridership counters sensor data 10</td><td>U8</td><td>0-254
255 – no data</td></tr>
<tr><td rowspan="2">115</td><td>Ridership counters sensor data 11</td><td rowspan="2">2</td><td>U8</td><td>0-254
255 – no data</td></tr>
<tr><td>Ridership counters sensor data 12</td><td>U8</td><td>0-254
255 – no data</td></tr>
<tr><td rowspan="2">116</td><td>Ridership counters sensor data 13</td><td rowspan="2">2</td><td>U8</td><td>0-254
255 – no data</td></tr>
<tr><td>Ridership counters sensor data 14</td><td>U8</td><td>0-254
255 – no data</td></tr>
<tr><td rowspan="2">117</td><td>Ridership counters sensor data 15</td><td rowspan="2">2</td><td>U8</td><td>0-254
255 – no data</td></tr>
<tr><td>Ridership counters sensor data 16</td><td>U8</td><td>0-254
255 – no data</td></tr>
<tr><td>118</td><td>Auto-informer status</td><td>1</td><td>U8</td><td>Bit field:
<table>
<tr><td>Digit</td><td>Value</td></tr>
<tr><td>0</td><td>0 – auto-informer off
1 - auto-informer on</td></tr>
<tr><td>1</td><td>0 – object outside the geofence
1 - object in the geofence</td></tr>
<tr><td>2</td><td>0 – geo-fence does not correspond the route
1 – geo-fence 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 errrors
1 – operation with SD card error</td></tr>
<tr><td>5</td><td>0 – no violation
1 - violation of the driving mode</td></tr>
<tr><td>6</td><td>0 – automatics mode
1 – manual mode</td></tr>
<tr><td>7</td><td>reserv</td></tr>
</table></td></tr>
<tr><td>119</td><td>Last geo-fence ID</td><td>2</td><td>U16</td><td>1-65535
0 – was not in geo-fence</td></tr>
<tr><td>120</td><td>Last stop ID</td><td>2</td><td>U16</td><td>1-65535
0 – no arrival 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 established</td></tr>
<tr><td>122</td><td>Camera status</td><td>1</td><td>U8</td><td>Bit field:
<table>
<tr><td>Digit</td><td>Value</td></tr>
</table></td></tr>
</table>

<table>
<tr><td></td><td></td><td></td><td></td><td>
<table>
<tr><td>0</td><td>0 -camera not available
1 - camera available</td></tr>
<tr><td>1</td><td>Auto shooting:
0 – auto shooting off
1 - auto shooting on</td></tr>
<tr><td>2</td><td>Operation mode:
0 – regular operation mode
1 – number of stored pictures exceeded maximum</td></tr>
<tr><td>3</td><td>0 – no error
1 – operation with SD-card error</td></tr>
<tr><td>4..7</td><td>reserve</td></tr>
</table>
</td></tr>
<tr><td colspan="5"><b>FLEX 3.0</b></td></tr>
<tr><td>123</td><td>Device 2 status</td><td>1</td><td>U8</td><td>Bit field:
<table>
<tr><td>Digit</td><td>Value</td></tr>
<tr><td>0</td><td>Device tamper sensor:
0 – in norm;
1 – body is opened.</td></tr>
<tr><td>1-7</td><td>Reserve</td></tr>
</table>
</td></tr>
<tr><td>124</td><td>Status of function module 3</td><td>1</td><td>U8</td><td>Bit field:
<table>
<tr><td>Digit</td><td>Value</td></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>Connection state status</td><td>1</td><td>U8</td><td>Bit field:
<table>
<tr><td>Digit</td><td>Value</td></tr>
<tr><td>0-2</td><td>Connection status:
0 – no connection;
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>Operation with server №1
0 – not connected;
1 – connected.</td></tr>
<tr><td>4</td><td>Operation with server №2
0 – not connected;
1 – connected.</td></tr>
<tr><td>5</td><td>Operation with server №3
0 – not connected;
1 – connected.</td></tr>
<tr><td>6</td><td>Operation with RCS/RFU
0 – not connected;
1 – connected.</td></tr>
<tr><td>7</td><td>Operation with configurator via Bluetooth:
0 – not connected;
1 – connected.</td></tr>
</table>
</td></tr>
<tr><td>126</td><td>Current values of discrete sensors 3</td><td>1</td><td>U8</td><td>Bit field:
<table>
<tr><td>Digit</td><td>Value</td></tr>
<tr><td>0</td><td>input In17</td></tr>
</table>
</td></tr>
</table>

				1 input In18 2 input In19 3 input In20 4 input In21 5 input In22 6 input In23 7 input In24 0 — sensor operate in a normal state; 1 — sensor triggered.
127	Readings of pulse counter 3	4	U32	Number of pulses counted at the time of the event 0-2^{32}
128	Readings of pulse counter 4	4	U32	Number of pulses counted at the time of the event 0-2^{32}
129	Readings of pulse counter 5	4	U32	Number of pulses counted at the time of the event 0-2^{32}
130	Readings of pulse counter 6	4	U32	Number of pulses counted at the time of the event 0-2^{32}
131	Readings of pulse counter 7	4	U32	Number of pulses counted at the time of the event 0-2^{32}
132	Readings of pulse counter 8	4	U32	Number of pulses counted at the time of the event 0-2^{32}
133	Frequency on analog-frequency sensor 3	2	U16	Frequency value 0-20000 Hz
134	Frequency on analog-frequency sensor 4	2	U16	Frequency value 0-20000 Hz
135	Frequency on analog-frequency sensor 5	2	U16	Frequency value 0-20000 Hz
136	Frequency on analog-frequency sensor 6	2	U16	Frequency value 0-20000 Hz
137	Frequency on analog-frequency sensor 7	2	U16	Frequency value 0-20000 Hz
138	Frequency on analog-frequency sensor 8	2	U16	Frequency value 0-20000 Hz
139	Accelerometer virtual sensor status	1	U8	Bit field: Digit Value 0 SH1 sensor state 1 SH2 sensor state 2 SH3 sensor state 3 SH4 sensor state 4 WAKEUP sensor state 5..7 Reserve 0 – in norm; 1 – activated.
140	Internal tilt sensor. The angle of inclination relative to the local vertical	1	U8	In degrees from 0 to 25 in increment of 0,25 degree.
141	Internal tilt sensor. Angles of inclination relative to a steep line	2	I8 I8	Pitch angle in degrees -90 to 90 in increments of 1 Roll angle in degrees -180 to 180 in increments of 1.5
142	External tilt sensor. Axial deviations	3	U8 U8 U8	According to X axle in degrees from 0 to 180 in increment of 1 According to Y axle in degrees from 0 to 180 in increment of 1 According to Z axle in degrees from 0 to 180 in increment of 1
143	EcoDriving. Maximum value of vertical acceleration for the period	2	I16	0..+24000 (0..+24 g miltipled by 1000)
144	EcoDriving. Maximum value of vertical speed for the period	1	U8	0..254 km/h
145	EcoDriving. Speed threshold state	1	U8	Bit field: Digit Value

<table>
<tr><td></td><td></td><td></td><td></td><td><table><tr><td>0</td><td>State of threshold 1</td></tr><tr><td>1</td><td>State of threshold 2</td></tr><tr><td>2</td><td>State of threshold 3</td></tr><tr><td>3</td><td>State of threshold 4</td></tr><tr><td>4</td><td>State of threshold 5</td></tr><tr><td>5</td><td>State of threshold 6</td></tr><tr><td>6..7</td><td>Reserve</td></tr></table>0 – threshold in norm;
1 – threshold was activated.</td></tr>
<tr><td rowspan="3">146</td><td rowspan="3">EcoDriving. Acceleration threshold state</td><td rowspan="3">3</td><td>U8</td><td>Bit field:<table><tr><td>Digit</td><td></td></tr><tr><td>0</td><td>State of acceleration threshold 1</td></tr><tr><td>1</td><td>State of acceleration threshold 2</td></tr><tr><td>2</td><td>State of acceleration threshold 3</td></tr><tr><td>3</td><td>State of acceleration threshold 4</td></tr><tr><td>4</td><td>State of breaking threshold 1</td></tr><tr><td>5</td><td>State of breaking threshold 2</td></tr><tr><td>6</td><td>State of breaking threshold 3</td></tr><tr><td>7</td><td>State of breaking threshold 4</td></tr></table>0 – threshold in norm;
1 – threshold was activated</td></tr>
<tr><td>U8</td><td>Bit field:<table><tr><td>Digit</td><td>Value</td></tr><tr><td>0</td><td>Threshold of acceleration to the left 1</td></tr><tr><td>1</td><td>Threshold of acceleration to the left 2</td></tr><tr><td>2</td><td>Threshold of acceleration to the left 3</td></tr><tr><td>3</td><td>Threshold of acceleration to the left 4</td></tr><tr><td>4</td><td>Threshold of acceleration to the right 1</td></tr><tr><td>5</td><td>Threshold of acceleration to the right 2</td></tr><tr><td>6</td><td>Threshold of acceleration to the right 3</td></tr><tr><td>7</td><td>Threshold of acceleration to the right 4</td></tr></table>0 – threshold in norm;
1 – threshold was activated.</td></tr>
<tr><td>U8</td><td>Bit field:<table><tr><td>Digit</td><td>Value</td></tr><tr><td>0</td><td>State of the vertical acceleration threshold 1</td></tr><tr><td>1</td><td>State of the vertical acceleration threshold 2</td></tr><tr><td>2</td><td>State of the vertical acceleration threshold 3</td></tr><tr><td>3</td><td>State of the vertical acceleration threshold 4</td></tr><tr><td>4..7</td><td>Reserve</td></tr></table>0 – threshold in norm;
1 – threshold was activated.</td></tr>
<tr><td>147</td><td>Frequency on input 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 on input 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 on input 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 on input 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 on input FLS 485 №5</td><td>2</td><td>U16</td><td>0...65534 Hz
65535 – no data</td></tr>
<tr><td>152</td><td>Frequency on input FLS 485 №6</td><td>2</td><td>U16</td><td>0...65534 Hz
65535 – no data</td></tr>
<tr><td>153</td><td>Frequency on input FLS 485 №7</td><td>2</td><td>U16</td><td>0...65534 Hz
65535 – no data</td></tr>
<tr><td>154</td><td>Frequency on input FLS 485 №8</td><td>2</td><td>U16</td><td>0...65534 Hz
65535 – no data</td></tr>
</table>

<table>
<tr><td>155</td><td>Frequency on input FLS 485 №9</td><td>2</td><td>U16</td><td>0...65534 Hz
65535 – no data</td></tr>
<tr><td>156</td><td>Frequency on input FLS 485 №10</td><td>2</td><td>U16</td><td>0...65534 Hz
65535 – no data</td></tr>
<tr><td>157</td><td>Frequency on input FLS 485 №11</td><td>2</td><td>U16</td><td>0...65534 Hz
65535 – no data</td></tr>
<tr><td>158</td><td>Frequency on input FLS 485 №12</td><td>2</td><td>U16</td><td>0...65534 Hz
65535 – no data</td></tr>
<tr><td>159</td><td>Frequency on input FLS 485 №13</td><td>2</td><td>U16</td><td>0...65534 Hz
65535 – no data</td></tr>
<tr><td>160</td><td>Frequency on input FLS 485 №14</td><td>2</td><td>U16</td><td>0...65534 Hz
65535 – no data</td></tr>
<tr><td>161</td><td>Frequency on input FLS 485 №15</td><td>2</td><td>U16</td><td>0...65534 Hz
65535 – no data</td></tr>
<tr><td>162</td><td>Frequency on input FLS 485 №16</td><td>2</td><td>U16</td><td>0...65534 Hz
65535 – no data</td></tr>
<tr><td>163</td><td>High-precision temperature sensor 1</td><td>2</td><td>I16</td><td>Fom −273,15 to +1638,35 in increment of 0,05 °C
0x8000 – no data</td></tr>
<tr><td>164</td><td>High-precision temperature sensor 2</td><td>2</td><td>I16</td><td>From −273,15 to +1638,35 in increment of 0,05 °C
0x8000 – no data</td></tr>
<tr><td>165</td><td>High-precision temperature sensor 3</td><td>2</td><td>I16</td><td>From −273,15 to +1638,35 in increment of 0,05 °C
0x8000 – no data</td></tr>
<tr><td>166</td><td>High-precision temperature sensor 4</td><td>2</td><td>I16</td><td>From −273,15 to +1638,35 in increment of 0,05 °C
0x8000 – no data</td></tr>
<tr><td>167</td><td>High-precision humidity sensor 1</td><td>1</td><td>U8</td><td>From 0 to 100 in increment of 0,5 %
0xFF – no data</td></tr>
<tr><td>168</td><td>High-precision humidity sensor 2</td><td>1</td><td>U8</td><td>From 0 to 100 in increment of 0,5 %
0xFF – no data</td></tr>
<tr><td>169</td><td>High-precision humidity sensor 3</td><td>1</td><td>U8</td><td>From 0 to 100 in increment of 0,5 %
0xFF – no data</td></tr>
<tr><td>170</td><td>High-precision humidity sensor 4</td><td>1</td><td>U8</td><td>From 0 to 100 in increment of 0,5 %
0xFF – no data</td></tr>
<tr><td>171</td><td>Fuel level sensor. Sensor status.</td><td>2</td><td>U16</td><td>Bit field:
<table>
<tr><td>Digit</td><td>Value</td></tr>
<tr><td>0..3</td><td>Camera operation mode «Supply»</td></tr>
<tr><td>4..7</td><td>Camera operation mode «Processing»</td></tr>
<tr><td>8..11</td><td>Engine operation mode according to consumption</td></tr>
<tr><td>12..13</td><td>Power status</td></tr>
<tr><td>14..15</td><td>Reserve</td></tr>
</table>
Camera and engine operation modes:
0 – idling
1 – nominal mode
2 – overload
3 – fake counting
4 – negative
5 – interference
6..15 – resrve</td></tr>
<tr><td>172</td><td>Fuel consumption level.
Information about malfunctions</td><td>4</td><td>U32</td><td>Bit field:
<table>
<tr><td>Digits</td><td>Values</td></tr>
<tr><td>0..31</td><td>Reserve</td></tr>
</table></td></tr>
<tr><td>173</td><td>Fuel consumption level.
Total fuel consumption</td><td>4</td><td>U32</td><td>in 0.01 l.
No value: 0xFFFFFFFF</td></tr>
<tr><td>174</td><td>Fuel consumption level. Fuel consumption per trip</td><td>4</td><td>U32</td><td>in 0.01 l.
No value: 0xFFFFFFFF</td></tr>
<tr><td>175</td><td>Fuel consumption sensor. Current flow speed.</td><td>2</td><td>I16</td><td>in 0.1 l/h.
No value: 0x8000</td></tr>
<tr><td>176</td><td>Fuel consumption sensor. Total fuel amount in Supply chamber.</td><td>4</td><td>U32</td><td>in 0.01 l.
No value: 0xFFFFFFFF</td></tr>
<tr><td>177</td><td>Fuel consumption sensor. Curent speed of supply chamber</td><td>2</td><td>I16</td><td>in 0.1 l/h
No value: 0x8000</td></tr>
</table>

<table>
<tr><td>178</td><td>Fuel consumption sensor.
Temperature of supply chamber</td><td>2</td><td>I16</td><td>in 0.1 °C
No value: 0x8000</td></tr>
<tr><td>179</td><td>Fuel consumption sensor. Fuel total amount in processing camera</td><td>4</td><td>U32</td><td>in 0.01 l
No value: 0xFFFFFFFF</td></tr>
<tr><td>180</td><td>Fuel consumption sensor. Flow current speed in the processing camera</td><td>2</td><td>I16</td><td>in 0.1 l/h
No value: 0x8000</td></tr>
<tr><td>181</td><td>Fuel consumption sensor.
Temperature of processing chamber</td><td>2</td><td>I16</td><td>in 0.1 °C
No value: 0x8000</td></tr>
<tr><td rowspan="2">182</td><td rowspan="2">Refrigeration unit. Unit state</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>Connection with refrigirator:
0 -absent, 1 – on the line</td></tr>
<tr><td>1</td><td>Door state:
0 -close, 1 - open.</td></tr>
<tr><td>2..4</td><td>Installatin type:
0 – unknown;
1 – ThermoKing of 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>Opration mode:
0 – no data
1 – engine is off;
2 – heat;
3 – cooling;
4 – defrost;
129…170 – one of ThermoKing of SLX series unit state (см. Appendix A4);

Rest values- reserve.</td></tr>
<tr><td>183</td><td>Refrigeration unit.
Refrigerator temperature in section 1 (Refrigirator temperature)</td><td>2</td><td>I16</td><td>From -273,15 to 327,67 in increment of 0.01 °C
No value: 0x8000</td></tr>
<tr><td>184</td><td>Refrigeration unit.
Refrigerator temperature in section 2</td><td>2</td><td>I16</td><td>From -273,15 to 327,67 in increment of 0.01 °C
No value: 0x8000</td></tr>
<tr><td>185</td><td>Refrigeration unit.
Refrigerator temperature in section 3</td><td>2</td><td>I16</td><td>From -273,15 to 327,67 in increment of 0.01 °C
No value: 0x8000</td></tr>
<tr><td>186</td><td>Refrigeration unit.
Specified temperature 1</td><td>2</td><td>I16</td><td>From -273,15 to 327,67 in increment of 0.01 °C
No value: 0x8000</td></tr>
<tr><td>187</td><td>Refrigeration unit.
Specified temperature 2</td><td>2</td><td>I16</td><td>From -273,15 to 327,67 in increment of 0.01 °C
No value: 0x8000</td></tr>
<tr><td>188</td><td>Refrigeration unit.
Specified temperature 3</td><td>2</td><td>I16</td><td>From -273,15 to 327,67 in increment of 0.01 °C
No value: 0x8000</td></tr>
<tr><td>189</td><td>Refrigeration unit.
Ambient air temperature</td><td>2</td><td>I16</td><td>From -273,15 to 327,67 in increment of 0.01 °C
No value: 0x8000</td></tr>
<tr><td>190</td><td>Refrigeration unit.
Coolant temperature</td><td>2</td><td>I16</td><td>From -273,15 to 327,67 in increment of 0.01 °C
No value: 0x8000</td></tr>
<tr><td>191</td><td>Refrigeration unit.
Battery voltage</td><td>2</td><td>U16</td><td>From 0 to 65534 in increment of 1 mV
No value: 0xFFFF</td></tr>
<tr><td>192</td><td>Refrigeration unit.
Battery current</td><td>2</td><td>U16</td><td>From 0 to 65534 in increment of 1 mA
No value 0xFFFF</td></tr>
<tr><td>193</td><td>Refrigeration unit.
Engine hours</td><td>4</td><td>U32</td><td>From 0 to 42949672,94 in increment of 0.01 h
No value: 0xFFFFFFFF</td></tr>
<tr><td>194</td><td>Refrigeration unit.
Engine hours from the network</td><td>4</td><td>U32</td><td>From 0 to 42949672,94 in increment of 0.01 h
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. Code of the most</td><td>U16</td><td>0…65535</td></tr>
</table>

<table>
<tr><td></td><td>important error</td><td></td><td></td><td></td></tr>
<tr><td rowspan="2">196</td><td>Refrigeration unit. Code of the 2-nd most important error</td><td rowspan="2">4</td><td>U16</td><td>0…65535</td></tr>
<tr><td>Refrigeration unit. Code of the 3-rd most important error</td><td>U16</td><td>0…65535</td></tr>
<tr><td rowspan="3">197</td><td>Refrigeration unit. Code of the 4-th most important error</td><td rowspan="3">6</td><td>U16</td><td>0…65535</td></tr>
<tr><td>Refrigeration unit. Code of the 5-th most important error</td><td>U16</td><td>0…65535</td></tr>
<tr><td>Refrigeration unit. Code of the 6-th most important error</td><td>U16</td><td>0…65535</td></tr>
<tr><td rowspan="2">198</td><td rowspan="2">Refrigeration unit.
Engine state</td><td rowspan="2">3</td><td>U8</td><td>Operation mode:
0 – no data;
1 – no revolutions, engine stopped;
2 – dieael runs, low speed;
3 – dieael runs, рщпр speed;
4 – Operates from network. Powered by network</td></tr>
<tr><td>U16</td><td>Engine revolutions:
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 stop of engine;
2 – Continuous – continuous operation mode
129…156 – one of ThermoKing of SLX series unit state (см. Appendix A4);

Rest values- reserve.</td></tr>
<tr><td>200</td><td>Information about being in geofences</td><td>2</td><td>U16</td><td>Bit field:
<table>
<tr><td>Digit</td><td>Value</td></tr>
<tr><td>0</td><td>Geo-fence 1</td></tr>
<tr><td>1</td><td>Geo-fence 2</td></tr>
<tr><td>2</td><td>Geo-fence 3</td></tr>
<tr><td>3</td><td>Geo-fence 4</td></tr>
<tr><td>4</td><td>Geo-fence 5</td></tr>
<tr><td>5</td><td>Geo-fence 6</td></tr>
<tr><td>6</td><td>Geo-fence 7</td></tr>
<tr><td>7</td><td>Geo-fence 8</td></tr>
<tr><td>8</td><td>Geo-fence 9</td></tr>
<tr><td>9</td><td>Geo-fence 10</td></tr>
<tr><td>10..15</td><td>Reserve</td></tr>
</table>
0 – device outside the geofence;
1 - device in the geofence.</td></tr>
<tr><td>201</td><td>CAN. Status flags</td><td>2</td><td>U16</td><td>Status flags.
Bit field:
<table>
<tr><td>Digits</td><td>Values</td></tr>
<tr><td>0</td><td>Igniton is on</td></tr>
<tr><td>1</td><td>Car alarm is armed (in panic mode)</td></tr>
<tr><td>2</td><td>Vehicle is closed using the standard key fob</td></tr>
<tr><td>3</td><td>Key is in the ignition</td></tr>
<tr><td>4</td><td>Dinamic ignition 2 is on</td></tr>
<tr><td>5</td><td>Front passenger door is open</td></tr>
<tr><td>6</td><td>Rear passenger doors are open</td></tr>
<tr><td>7</td><td>Clutch is released</td></tr>
<tr><td>8</td><td>Driver door is open</td></tr>
<tr><td>9</td><td>Passengers doors are open</td></tr>
<tr><td>10</td><td>Boot is open</td></tr>
<tr><td>11</td><td>Bonnet is open</td></tr>
<tr><td>12</td><td>Handbrake lever tightened</td></tr>
<tr><td>13</td><td>Foot-brake is pressed</td></tr>
</table></td></tr>
</table>

<table>
<tr><td></td><td></td><td></td><td></td><td><table><tr><td>14</td><td>Engine is on</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 – Vehicle is closed using the standard key fob;
2 – Vehicle is opened using the standard key fob
3 – Boot is opened using the standard key fob
4 – module sent a reset to the alarm signal
5 - Reserved
6 - Reserved
7 – modul switched to the energy conservation mode «sleep mode»
8…255 - Reserved</td></tr>
<tr><td>203</td><td>CAN. Accident indicators</td><td>4</td><td>U32</td><td>Bit field:<table><tr><td>Bit</td><td>Value</td></tr><tr><td>0</td><td>STOP</td></tr><tr><td>1</td><td>Pressure / oil level</td></tr><tr><td>2</td><td>Temperature / Coolant level</td></tr><tr><td>3</td><td>Hand brake system</td></tr><tr><td>4</td><td>Battery charge</td></tr><tr><td>5</td><td>AIRBAG (pillow of safety)</td></tr><tr><td>6</td><td>ESP is off</td></tr><tr><td>7</td><td>ESP indicator is on</td></tr><tr><td>8</td><td>Check engine</td></tr><tr><td>9</td><td>Illumination failure</td></tr><tr><td>10</td><td>Low tyre pressure</td></tr><tr><td>11</td><td>Worn-out brake pads</td></tr><tr><td>12</td><td>Warning</td></tr><tr><td>13</td><td>ABS (anti-lock system)</td></tr><tr><td>14</td><td>Low fuel level</td></tr><tr><td>15</td><td>Approaching service maintenance</td></tr><tr><td>16</td><td>ESP (electronic stability control)</td></tr><tr><td>17</td><td>Spark-plug indicator</td></tr><tr><td>18</td><td>FAP (particle filter)</td></tr><tr><td>19</td><td>EPC (electrical pressure control)</td></tr><tr><td>20</td><td>Position lamps</td></tr><tr><td>21</td><td>Dipped headlights</td></tr><tr><td>22</td><td>Main beam headlamps</td></tr><tr><td>23</td><td>Reserved</td></tr><tr><td>24</td><td>Readiness to start movement</td></tr><tr><td>25</td><td>Cruise control</td></tr><tr><td>26</td><td>Automatic retarder</td></tr><tr><td>27</td><td>Hand retarder</td></tr><tr><td>28</td><td>Conditioner is on</td></tr><tr><td>29</td><td>Reserved</td></tr><tr><td>30</td><td>Driver 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. Information about malfunctions</td><td rowspan="2">5</td><td>U8</td><td>Malfunction indication lamp state.
Bit field:<table><tr><td>Bit</td><td>Value</td></tr><tr><td>0</td><td>Mulfanction indication lamp is on</td></tr><tr><td>1</td><td>Red lamp «Stop» is on</td></tr><tr><td>2</td><td>Yellow lamp «Warning» is on</td></tr><tr><td>3</td><td>Lamp «Protection» is on</td></tr><tr><td>4</td><td>Mulfanctin lamp is blinking</td></tr><tr><td>5</td><td>Red lamp «Stop» is blinking</td></tr><tr><td>6</td><td>Yellow lamp «Warning» is blinking</td></tr><tr><td>7</td><td>Lamp «Protection» is blinking</td></tr></table></td></tr>
<tr><td>U32</td><td>Diagnostic trouble code (Diagnostics Trouble Code).</td></tr>
</table>

				0x00000000…0xFFFFFFFF
205	Users engine hours 1 (operation under load)	4	U32	0-2^{32} sec
206	Users engine hours 2 (operation without load)	4	U32	0-2^{32} sec
207	Users parameter 1 byte №1	1	U8, S8	Universal field, which can be represented as: 1. Unsigned integer 0 ... 255; 2. Signed integer is -128 ... 127; 3. Bit field with bits 0..7.
208	Users parameter 1 byte №2	1	U8, S8	
209	Users parameter 1 byte №3	1	U8, S8	
210	Users parameter 1 byte №4	1	U8, S8	
211	Users parameter 1 byte №5	1	U8, S8	
212	Users parameter 1 byte №6	1	U8, S8	
213	Users parameter 1 byte №7	1	U8, S8	
214	Users parameter 1 byte №8	1	U8, S8	
215	Users parameter 1 byte №9	1	U8, S8	
216	Users parameter 1 byte №10	1	U8, S8	
217	Users parameter 1 byte №11	1	U8, S8	
218	Users parameter 1 byte №12	1	U8, S8	
219	Users parameter 1 byte №13	1	U8, S8	
220	Users parameter 1 byte №14	1	U8, S8	
221	Users parameter 1 byte №15	1	U8, S8	
222	Users parameter 1 byte №16	1	U8, S8	
223	Users parameter 2 bytes №1	2	U16, S16	Universal field, which can be represented as: 1. Unsigned integer 0…65535; 2. Signed integer is -32768…32767; 3. Bit field with bits 0..15.
224	Users parameter 2 bytes №2	2	U16, S16	
225	Users parameter 2 bytes №3	2	U16, S16	
226	Users parameter 2 bytes №4	2	U16, S16	
227	Users parameter 2 bytes №5	2	U16, S16	
228	Users parameter 2 bytes №6	2	U16, S16	
229	Users parameter 2 bytes №7	2	U16, S16	
230	Users parameter 2 bytes №8	2	U16, S16	
231	Users parameter 2 bytes №9	2	U16, S16	
232	Users parameter 2 bytes №10	2	U16, S16	
233	Users parameter 2 bytes №11	2	U16, S16	
234	Users parameter 2 bytes №12	2	U16, S16	
235	Users parameter 2 bytes №13	2	U16, S16	
236	Users parameter 2 bytes №14	2	U16, S16	
237	Users parameter 2 bytes №15	2	U16, S16	
238	Users parameter 4 bytes №1	4	U32, S32, float	Universal field, which can be represented as: 1. Unsigned integer 0…4294967295; 2. Signed integer is -2147483648…2147483647; 3. Bit field with bits 0..31. 4. Float number -3.40282347e+38…-1.17549435e-38, 0, 1.17549435E-38e …3.40282347e+38
239	Users parameter 4 bytes №2	4	U32, S32, float	
240	Users parameter 4 bytes №3	4	U32, S32, float	
241	Users parameter 4 bytes №4	4	U32, S32, float	
242	Users parameter 4 bytes №5	4	U32, S32, float	
243	Users parameter 4 bytes №6	4	U32, S32, float	
244	Users parameter 4 bytes №7	4	U32, S32, float	
245	Users parameter 4 bytes №8	4	U32, S32, float	
246	Users parameter 4 bytes №9	4	U32, S32, float	
247	Users parameter 4 bytes №10	4	U32, S32, float	
248	Users parameter 4 bytes №11	4	U32, S32,	

			float	
249	Users parameter 4 bytes №12	4	U32, S32, float	
250	Users parameter 4 bytes №13	4	U32, S32, float	
251	Users parameter 4 bytes №14	4	U32, S32, float	
252	Users parameter 4 bytes №15	4	U32, S32, float	
253	Users parameter 8 bytes №1	8	U64, S64, double	Universal field which can be represented as: 1. Unsigned integer 0…18446744073709551615; 2. Signed integer -9223372036854775808… 9223372036854775807; 3. Bit field with bits 0..63. 4. Floating number -1.7976931348623158e+308… -2.2250738585072014e-308, 0, 2.2250738585072014e-308e… 1.7976931348623158e+308
254	Users parameter 8 bytes №2	8	U64, S64, double	
255	Users parameter 8 bytes №3	8	U64, S64, double	

Appendix A.2. Structure of additional telemetric records of the FLEX 2.0 extension

<table>
<tr><th>№</th><th>Record field</th><th>Record item size</th><th>Data format</th><th>Received value</th></tr>
<tr><td>1</td><td>Data length</td><td>2</td><td>U16</td><td>Total size of the package fields (numbered 2-15)</td></tr>
<tr><td>2</td><td>Data structure version</td><td>1</td><td>U8</td><td>Data Structure Version
the static part of the package to identify the compatibility of the format of the transmitted data on the server and in the device.
For version 1.0 10 (0x0A)</td></tr>
<tr><td>3</td><td>Data length</td><td>1</td><td>U8</td><td>The length of the static part of the package (fields 4-14)</td></tr>
<tr><td>4</td><td>Write number in nonvolatile memory</td><td>4</td><td>U32</td><td>Starts from zero, increments with every record. Never diminishes.</td></tr>
<tr><td>5</td><td>Event code corresponding to this record</td><td>2</td><td>U16</td><td>Codes indicated in the table are recorded in the protocol.</td></tr>
<tr><td>6</td><td>Event time</td><td>4</td><td>U32</td><td>Number of seconds since 1970</td></tr>
<tr><td>7</td><td>GPS/GLONASS navigation sensor state</td><td>1</td><td>U8</td><td>Bit field:
<table>
<tr><td>Digits</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 0-32</td></tr>
</table></td></tr>
<tr><td>8</td><td>Time of the last valid coordinates (before event occurred)</td><td>4</td><td>U32</td><td>Number of seconds since 1970</td></tr>
<tr><td>9</td><td>Last valid latitude</td><td>4</td><td>I32</td><td>Latitude angle which was fixed when the last valid coordinates were received.
In ten thousandths of a minute.
For example,
55° 42,2389'
It will be represented as
33422389</td></tr>
<tr><td>10</td><td>Last valid longitude</td><td>4</td><td>I32</td><td>Longitude angle which was fixed when the last valid coordinates were received.
In ten thousandths of a minute.
For example,
37° 41,6063'
It will be represented as
22616063</td></tr>
<tr><td>11</td><td>Last valid height</td><td>4</td><td>I32</td><td>Altitude relative to sea level, recorded upon receipt of the last valid coordinates:
In decimetres
For example
205 meters
It will be represented as
2050 decimeters</td></tr>
<tr><td>12</td><td>Speed</td><td>4</td><td>Float</td><td>Speed fixed when receiving the last valid coordinates.
In km/h</td></tr>
<tr><td>13</td><td>Rate</td><td>2</td><td>U16</td><td>Rate fixed when receiving the last valid coordinates.0° … 360°</td></tr>
</table>

14	Current mileage	4	Float	Mileage recorded at the time of the event, calculated at the time of receipt of valid navigation data. In km.
15	Dinamic part of the package	-	U8	Data set depends on the event that occurred

Structure of the dynamic part of the optional FLEX 2.0 package

Dinamic part of the package contains a set of fields with different formats and contents.

General field format

№	Record field	Record item size	Data format	Принимаемые значение
1	Field type	1	U8	A variable that defines the format of the next field.
2	Field len	1	U8	Field length
3	Field contents	-	-	Data depending on the type of field and the size specified by the "Long field"

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

0x01 –Touch Memory key.

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

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

№	Record field	Record item size	Data format	Принимаемые значение
3	Driver card number	16	U8[16]	Identification card number for the tachograph.

Event codes for which this package is formed allow you to determine which card was installed.

Code (in hexadecimal number system)	Code (in decimal number system)	Decryption
2530	9520	Installing a card in slot No. 1
2531	9521	Installing a card in slot No. 2

0x03 –RFID tag code.

№	Record field	Record item size	Data format	Принимаемые значение
3	Код радиометки RFID	8	U64	RFID tag code read by the device at the time of the event.

0x04 … 0xFF – Reserved for futher expansion

Appendix A.3. Examples of basic FLEX packages

Пример пакета согласования протоколов FLEX 2.0

Byte number	Byte	Value
1	0x40	@NTC – preamle.
2	0x4e	
3	0x54	
4	0x43	
5	0x01	1 — receiver identifier.
6	0x00	
7	0x00	
8	0x00	
9	0x00	0 — sender identifier.
10	0x00	
11	0x00	
12	0x00	
13	0x1a	26 — number of data bytes in a package.
14	0x00	
15	0x1b	Data checksum.
16	0x19	Preamble checksum.
17	0x2a	*>FLEX
18	0x3e	
19	0x46	
20	0x4c	
21	0x45	
22	0x58	
23	0xb0	FLEX protocol symbol
24	0x14	2.0 — protocol version.
25	0x14	2.0 — data structure version.
26	0x7a	122 — configuration field size
27	0xf2	A bit array with information about transmitted fields of the data structure. In this case, the following fields are selected: - cross-cutting number in nonvolatile memory; - event code corresponding to this record; - event time; - device status; - GSM level; - voltage at the main power source.
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 Protocol Telemetry Example Package

Byte number	Byte	Value
1	0x7e	~A
2	0x41	
3	0x01	1 — telemetry record in the package.
4	0x1e	30 — record number in black box
5	0x00	
6	0x00	
7	0x00	
8	0xb0	6064 — code of occered event.
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 source.
17	0x2f	
18	0x24	8-bit byte CRC8 package.

Example of additional telemetry package of FLEX 2.0 protocol

Byte number	Byte	Value
1	0x7e	~E
2	0x45	
3	0x01	1 – one package in the archived message.
4	0x39	Total size of the first package is 57 bytes.
5	0x00	
6	0x0a	1.0 - version of the static part of the package.
7	0x25	Length of the static part of the package is 37 bytes.
8	0x10	Record cross-cutting number - 16.
9	0x00	
10	0x00	
11	0x00	
12	0x30	Event code - 2530 (card installation in slot №1).
13	0x25	
14	0x05	Event time.
15	0x37	
16	0xd7	
17	0x55	
18	0x33	Navigation sensor state.
19	0xec	Time of the last valid coodinates.
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 height.
32	0x07	
33	0x00	
34	0x00	
35	0x00	Speed.
36	0x00	
37	0x00	
38	0x00	
39	0x00	Rate.
40	0x00	
41	0x00	Current mileage.
42	0x00	
43	0x00	
44	0x00	
45	0x02	Type 2 subpackage - driver card.
46	0x10	Subpackage length - 16 byte.
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 package.

Appendix A.4. Compressor configuration and status for ThermoKing Units

Compressor configuration 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)

State 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

Appendix B. CRC8 checksum calculation algorithm

In order to verify 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 package (including the '~' character) to the last byte of data and is usually written at the end of the package. In order to calculate CRC8 it is possible to use a table function (programming language C).

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

Or to 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;
}
```

Appendix C. CRC16 Checksum Algorithm

In order to verify the integrity of the data in encrypted messages, a checksum is calculated using the CRC16 algorithm with a polynomial 0x1021 and an initial value of 0. The checksum is calculated from the beginning of the package (including the '#' character) to the last byte of data and is written at the end of the package. To calculate CRC16 it is possible to use a table function (programming language C).

```
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 (CEC - CRC) tabular method.
          * DESCRIPTION
          * of 16-bit cyclic redundancy code is calculated usind a tabular method.
          * PARAMETERS
          * crc - initial value of the CRC, 0 on the first call;
          * msg - message block pointer;
          * msg_len - number of bytes in a block.
          * RETUNRN VALUE
          * 16-bit CRC.
          */

        unsigned char temp;

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

Appendix D. NTCB Short Message List

Message	Description
System	
*?V	Request device model and version.
*?VGPS	Request information about firmware version of the GPS receiver
*?S	Request a unique device identifier.
*>S	Message with device individual identifier string.
*!DEV_RESET	Command to restart the device.
*?USSD	Arbitrary USSD-request
*!BEEP	Arbitrary buzzer sound indication command
*!CHNGSIM	Command to change SIM-cards
*?ICCID	Request a unique SIM serial number
*!O	Microphone listening command
Output lines	
*!1Y	Command to turn on output line 1.
*!1N	Command to turn off output line 1.
*!2Y	Command to turn on output line 2.
*!2N	Command to turn off output line 2.
*!3Y	Command to turn on output line 3.
*!3N	Command to turn off output line 3.
*!4Y	Command to turn on output line 4.
*!4N	Command to turn off output line 4.
*!SETOUT	Command to change output lines
Input lines	
*!OFF	Command to block input lines.
*!ON	Command to unlock input line.
Telemetry	
~A	Array of telemetry messaged in FLEX format.
~T	The structure of the package for sending an extraordinary message in FLEX format.
~C	The structure of the package for sending the current state in FLEX format.
~E	Array of additional telemetry messages FLEX 2.0.
~X	Structure of the package for sending additional FLEX 2.0 extraordinary messages.
*?L	Request for telemetry recording at the nearest moment before the specified date and time.
*?R	Request for telemetru records at the nearest moment after the specified date and time.
*?I	Request for telemetry recording by its index.
*?A	Request current state of the device.
*?ES	Request for device status with resending on SMS.
*!SYNC	Command of confirmation black box synchronization with the server.
*!REP_FL	Command to resend telemetry from black box
*!REP_SD	Command to resend telemetry from SD-card
Device operation mode	

*!GY	Arming command mode 1. (basic mode)
*!GN	Disarming command.
*!G2	Arming command mode 2. (service mode)
*!M	Command to change device operation mode
*?M	Request the current operation mode
RCS, RFU Service	
*!CNCT_RCS	Command to connect with configurator via RCS service.
*!CNCT_RFU	Command to connect with RFU sevice for device firmware uptade.
Touch Memory keys	
*>TMKEY	The structure of the NTCB package for sending the code of an unregistered Touch Memory key
*?TM	Request for the code of the last Touch Memory key read by the device
*?ERFT	Request for the last active radiotag
~O	Module code: 0x7D Command code: 4 – command to edit TM key.
Tachgraph (Module code: 0x81)	
~Q	Request code: 0 – request for information about the current state of tachpgraph; 2 – request for information about card №1; 3 – request for information about card №2; 4 – request for registration number of the vehicle.
~O	Command code: 0 – device authorization in tachograph; 1 – installation of a new authorization key; 2 – upload file formation; 3 – formation of the upload file with subsequent sending to email.
~G	Request identifier: 0 – DDD file block request.
Driver display (Module code: 0x82)	
~O	Command code: 0 – sending message to the driver.
*!DV	Command to send message to the driver display.
Auto-informer	
*!AINF	Command code: 0x01 – change of the current route; 0x02 – start sound file playback; 0x03 – change of the current motion mode.
*!AINF:	Command to change route.
*!AINF!	Command to start file playback.
*!AINF#	Command to change speed modes.
*&AINF	Alerts about Auto-informer events.
Camera (Module code: 0x80)	

~O	Command code:
	0 – automatic shooting control;
	1 – take a shot;
	2 – take a shot and then send it on email.

~Q	Request code: 0 – receiving information about camera; 1 – request for information about an image.

~N	Notification code: 0 – notification about creation a new image.
~G	Request code: 0 – request for review image; 1 – request for main image.
Data exchange between external interfaces and servers (Module code: 0x7F)	
*!U2S, *>U2S	Command to transfer data via a USB device to the server
*!S2U, *>S2U	Command to transfer data from the server via USB.
~P	Message code: 0x00 – data transfer in the transparent mode from USB; 0x01 – transparent data transmission from RS232; 0x02 – transparent data transmission from RS485.
*!UC	Users command RS485/RS232
CAN-LOG	
*!CANLOG	Setting the program number of the CAN-LOG device.
Operation with built-in accelerometer	
*!ACL_C	Accelerometer calibration command
Traffic fixation	
*?KRAI	Request for information about fixed traffic
*!KRAI	Overwrite reset command

Appendix E. Digital FLS error code table

Error code	Decription
Errors from the SIGNAL / SMART device	
65535 (or -1)	Invalid response from the sensor or invalid response prefix (general error code)
65534 (or -2)	nvalid command code in response
65533 (or -3)	Invalid sensor address in response
65532 (or -4)	CRC error
65531 (or -5)	Sensor is not initialized
65530 (or -6)	Break line, no response from sensor
Technoton FLS errors	
65529 (or -7)	Calibration error (Techoton)
65528 (or -8)	Hardware error (Technoton)
Errors from ultrasonic FLS “TS SENSOR” UZI-0.8 and UZI-2.5	
65527 (or -9)	Cable disconnection from sensor
65526 (or -10)	No signal
65525 (or -11)	Low battery
65524 (or -12)	Low battery + cable break from the sensor
65523 (or -13)	Low battery + no signal
Errors from FLS «LLS»	
65522 (or -14)	Invalid level value (general error code)