Ruptela device protocol

version 1.113

Table of Contents

Change log

Date	Version	Description
2014-07-31	1.13	Updated: Identification string for 4th generation FM.
2014-08-12	1.14	Updated: Transparent Channel Times stamp field added.
2014-09-02	1.15	Updated: Delrecords sms command deletes records from internal flash memory only.
2014-09-15	1.16	Updated: Server communication Tacho command 11 data flow example corrected.
2014-09-30	1.17	Updated: Record example incorrect CRC example fixed.
2014-10-07	1.18	Added: Generate accident records command.
2014-10-08	1.19	Added: Weighting system command (32); Updated: SMS caninfo modified.

2014-10-13	1.20	Added: SMS cansinfo;
		Updated: SMS caninfo modified;
		Update: smard card ack updated.

2014-11-03	1.21	Updated: Tunnel Channel payload length set to 1005;
2014-12-04	1.22	Added: SETIO IO command;
2015-01-05	1.23	Added: Dual Tunnel Channel;

2015-01-26	1.25	Updated: Dual Tunnel Channel;
		Updated: SMS via GPRS example;

2015-01-27	1.26	Updated: Caninfo, Cansinfo SMS;
2015-02-11	1.27	Updated: Weighting system;

2015-02-11	1.28	Updated: SMS message: getAPN;
		Updated: Dual Tunnel Channel;
		Updated: Logo changed;

2015-03-09	1.29	Added: Extended protocol description;
2015-03-12	1.30	Updated: Timestamp extension single records limit;

2015-04-10	1.31	Updated: DTC command 109 example;
		Updated: Tunnel Channel command 14 example;

2015-05-04	1.32	Added: SD logging command (34/134);
2015-06-15	1.33	Updated: Transparent channel example;
2015-07-20	1.34	Added: SMS command: webcoords – Google maps hyperlink with coordinates; Added: SMS command: setiotime – set output for temporary period;
2015-07-20	1.35	Added: SMS command: Banned - temporary banned operators;
2015-09-08	1.36	Added: SMS command: Imei - get IMEI number;
2015-09-08	1.37	Edited: extended records example;
2015-10-02	1.40	Updated: FW update protocol flowchart;
2015-11-12	1.41	Edited: Virtual odometer 106 command;

		Added: Supported SMS commands table;
		Added: lastchange SMS, SMS during critical process;
		Added: Record sending without GPS;
		Added: Configuration password over GPRS workflow;
		Added: Description of setcfg, getcfg, setioparam and getioparam commands;

2015-11-12	1.42	Edited: Corrected minor grammatical errors;
2015-11-12	1.43	Edited: Tunnel channel server example;
2015-11-25	1.44	Added: Description of setvalue command;
2016-03-10	1.45	Added: Description of set3g; get3g; dfota commands;
2016-04-16	1.46	Removed: Unsupported A-GPS command removed;
2016-05-30	1.47	Added: SMS command: Enginevolt - Automatic engine detection;
2016-06-30	1.48	Added: SMS command: lcvselect - LCV auto select; Added: SMS command: getlog/stoplog - get device log remotely;
2016-08-01	1.49	Updated: Description of accinfo;
2016-09-02	1.50	Change log transferred into the document;
2016-09-21	1.51	Corrections: “Record example: Altitude”; In 3.2.5 “Firmware chapter number”;
2016-10-13	1.52	Corrections: Removed duplicate command in the “Supported SMS commands table”; „getlog“ SMS command description; Added: Example for „cansinfo“ SMS command; Added: SMS command: „info“ - get some specific information from the FM device via SMS;
2016-11-02	1.53	Corrections: Link to FMI protocols supported by Garmin;
2016-11-18	1.54	Corrections: Added missing configuration file extensions in chapter 5; Corrections: Command 120 and Command 21/121 are supported only by FM-Pro3; Added: SMS command: info - Customizable SMS command; Corrections: Switchip command description improved; Corrections: Gsminfo command description improved; Corrections: Added responses examples after unsuccessful operation for the Connect and Econnect SMS commands;
2016-12-05	1.55	Added: “Ignore empty space before the SMS command” feature description; Added: SMS commands supported by the FM-Basic added to the “Supported SMS commands table”;
2016-12-12	1.56	Removed: SMS command “Fastsleep” removed; Various minor error corrections and description improvements;

2016-12-20	1.57	Improved “info” SMS command description;
		Improved “getlog/stoplog” SMS command description;
		FM device configuration and firmware file extensions updated;

2017-01-30	1.58	Updated: “Supported SMS commands table”;
		Corrections: Command 2/102 description improvements;
		Updated: SMS command “lcvselect” description;
		Added: Command 35/135 – Accident detection data;

2017-02-14	1.59	Updated: “Supported SMS commands table”;

		Updated: SMS command “accinfo” description;
		Added: “Supported communication commands table”;
		Added: USSD SMS command description;

2017-02-22	1.60	Updated: “Supported SMS commands table”; Corrections: Command 2/102 description improvements;

2017-03-09	1.61	Updated: “Setio – set outputs” SMS command description updated;
		Corrections: “Command 134, sub command 2 (0x02) – Stop sending procedure” – ID number corrected;
		Updated: “setcfg” command example updated;

2017-03-14	1.62	Corrections: “Record example: Time stamp in UNIX format”;
		Corrections: Command 7/108 description improvements;
		Corrections: Command 9/109 description improvements;
		Corrections: Command 34/134 description improvements;

2017-03-29	1.63	Updated: Added one more response example for the “getlog” SMS command; Updated: “Supported SMS commands table”; Updated: “Command 10/110 – Tachograph Communication” changes related to ATOL tachograph support; Updated: Description for the “Command 9/109 – Diagnostic Trouble Codes” updated. DTC packet filed description updated; Added: “clear dtc” command description;
2017-04-21	1.64	Removed: “gsminfo”parameter C – link with server closed.
2017-05-18	1.65	Updated:Additional description for “setiotime” SMS command for stuck DOUT cases;
2017-06-07	1.66	Updated: Additional description of record structure Latitude and Longitude. Updated: Note regarding the extended protocol and reference to the FMIOData file. Removed: one *setioparam* error code.

2017-06-15	1.67	Updated: Added a note in the “GPS elemenet – HDOP” about HDOP values when GSM tracking is enabled.
		Updated: SMS alerts description for FM 4 devices;
		Updated: New SMS commands – doutreset, immobilizer, resetimmob;
		Updated: Command 33/133 – FLS communication channel between FLS device and server;

2017-08-07	1.68	Updated: Changes in Command 33/133;
		Updated: *lcvselect* command with new flag *enableIO;*
		Updated: Server command 110, added column “User ID”;
		Updated: Custom SMS text function;
		Updated: Changed note in “setio” SMS command section.
		Updated: New SMS commands - setdinmode, getdinmode;

2017-09-27	1.69	Updated: New SMS command - “setlcv”.
2017-12-19	1.70	Added: Command 16/116 – HeartBeat.
2018-02-12	1.71	Added: New SMS command – *forward*;
2018-03-08	1.72	Updated: Reviewed communication commands descriptions. Updated: SMS command *accreset.*
2018-03-26	1.73	Updated: SMS command – setdinmode;

		Updated: SMS command – getdinmode;
		Updated: supported SMS commands table;
		Updated: Various examples and descriptions of all commands.

2018-05-30	1.74	Corrected: Command 15/115 packet structure.
2018-06-21	1.75	Added: Description of negative acknowledgement packet for Commands 1/100 and 68/100.
2018-08-07	1.76	Added: Command 37/137 – Files; Added: SMS command *snapshot* – RS232 camera pictures on demand; Updated: Supported communication command table; Updated: Supported SMS commands table.

2018-10-02	1.77	Added: Description of IO packet data via RS232 in Command 14/114;
		Updated: SMS commands *accinfo* and *accreset;*
		Corrected: Command 7/108 packet examples.

2018-10-29	1.78	Added: Command 18/115 – Dynamic identification packet;
		Added: SMS command *ssl status* – SSL authentication status;
		Updated: Supported communication command table;
		Updated: Supported SMS commands table.

2018-11-19	1.79	Removed: All commands unique to FM3 devices; Removed: FM4 BT device extensions. Added: SMS command *optiver* – Optitemp version status; Updated: Supported communication command table; Updated: Supported SMS commands table. Updated: SMS command password description. Various formatting corrections.

2018-12-21	1.80	Added: SMS command *iqfinfo* – iQFreeze parameter status;
		Added: SMS command *iqfset* – iQFreeze configuration;
		Added: SMS command *uptime* – Device uptime;
		Updated: Payload length in Command 18/115;
		Updated: Supported SMS commands table.

2019-02-15	1.81	Updated: Command 37/137 – Files; Updated: SMS command *snapshot.*
2019-03-08	1.82	Updated: Supported SMS commands table.
2019-03-26	1.83	Added: Modem GNSS FW update command for MC60.

2019-05-14	1.84	Removed: All commands unique to FM-Basic and UBI devices;
		Removed: FM-Basic and UBI device extensions;
		Added: CRC8 calculation algorithm example;
		Added: IDs in SMS commands;
		Updated: SMS command *getapn.*

2019-08-01	1.85	Removed: ATOL tachograph communication examples in Command 10/100.
2019-08-23	1.86	Added: SMS command *setlock.*

2019-09-13	1.87	Added: Command 19/107 - 2nd generation Smart Card data size and timestamp; Updated: Supported communication command table; Corrected: Server response CRC16 examples for commands 5/107 and 6/107.

2019-09-27	1.88	Added: Firmware update commands for modem firmware, SSL certificates and private key;
		Updated: Command 3/103 payload description;
		Updated: SMS command *version;*
		SMS commands are now sorted alphabetically.

2019-11-14	1.89	Added: SMS command *plockinfo*; Updated: SMS command *setiotime* examples.
2020-01-13	1.90	Added: SMS command *authid.*
2020-02-18	1.91	New device: FM-Eco4 T; Updated: Supported communication command table; Updated: Supported SMS commands table.
2020-04-03	1.92	New device: Trace5; Added: SMS command *getgfwver;* Updated: Dynamic identification string parameters in Command 18/115; Updated: Firmware update commands; Updated: SMS commands *modrev, setdinmode, getdinmode, setlock* and *setvalue;* Updated: Supported communication command table; Updated: Supported SMS commands table.
2020-07-10	1.93	New devices: HCV5, LCV5, Pro5; Added: SMS commands *getnetw, setnetw;* Updated: Firmware update commands*;* Updated: SMS commands *getio, setio, info, setiotime, get3g, set3g;* Updated: Supported communication command table; Updated: Supported SMS commands table; Updated: SMS command *getcfg.*
2020-08-05	1.94	Updated: Command 37/137.
2020-08-10	1.95	Added: SMS commands *neconnect, nreset, settemp, gettemp*.
2020-09-25	1.96	Added: Command 35/135. Updated: Commands 15/115 and 18/115. Updated: Configuration upload process.
2020-10-08	1.97	Added: Command *setdevice*. Updated: Commands 15/115. Updated: SMS commands *setio* and *setlock*.
2020-11-06	1.98	Added: Firmware .fwp packet. Updated: Firmware. Updated: Commands *info, setioparam.* Updated: Command 18/115.
2020-11-27	1.99	Updated: Command *info.* Updated: Command 35/135.

2020-12-18	1.100	Updated: SMS command *modrev.* Updated: Command 15/115 – Identification packet. Updated: Command 19/107 – 2nd generation Smart Card data size and timestamp.
2020-01-26	1.101	Updated: Supported SMS commands table. Updated: Command 30/130 – Garmin Device Request Status. Updated: Command 18/115 – Dynamic identification packet.
2020-03-26	1.102	Updated: Command 18/115 – Dynamic identification packet.
2021-05-10	1.103	Updated: Firmware upload process. Updated: Configuration.
2021-05-31	1.104	Added: Command 38/138 – Beacon Data Sending via GPRS Updated: Configuration Data Packet.
2021-06-22	1.105	Added: SMS command *tachostatus*. Updated: Supported SMS commands table. Updated: Command 17/117 – Set IO value.
2021-08-09	1.106	Updated: Supported communication command table. Removed: SMS commands *gettemp, settemp, setdevice.*
2021-08-30	1.107	Updated: Configuration upload process, Configuration Download from tracking device process.
2021-09-30	1.108	Updated: SMS during critical process and Information Messages and Alerts header level and order.
2021-11-05	1.109	Updated: Command 15/115 – Identification packet. Updated: SMS command list. Updated: SMS commands *setvalue, info.* Updated: Firmware and Configuration file extensions.
2021-11-26	1.110	Added: SMS commands *ahj-on/ahj-off.* Added: Firmware packet for Trace-2G and Trace5-LTM (Rev.A) description. Updated: Command 15/115 – Identification packet. Updated: Command 18/115 – Dynamic identification packet. Updated: Supported SMS commands table.
2022-01-14	1.111	Added: SMS command *dmpfconnect.* Updated: Supported SMS commands table.
2022-02-18	1.112	Added: Trace5-LTE relevant information where applicable. Updated: 3.2.6 Command 5/107 – Smart Card Data. Updated: 3.2.7 Command 6/107 – Smart Card Data Size and Time stamp. Updated: 3.2.9 Command 9/109 – Diagnostic Trouble Codes. Updated: 3.2.10 Command 10/110 – Tachograph Communication. Updated: 3.2.11 Command 11/111 – Tachograph Data Packet. Updated: 3.2.17 Command 18/115 – Dynamic identification packet. Updated: 3.2.18 Command 19/107 – 2nd generation Smart Card data size and timestamp. Updated: 3.2.20 Command 31/131 – Garmin Device Data. Updated: 3.2.24 Command 35/135 – Accident reconstruction. Updated: 3.2.22 Command 33/133 – FLS communication channel between device and server. Updated: 3.2.24.1 Subcommand 0 (0x00) – Requesting data from log.

2022-05-20	1.113	New devices: HCV5 Lite/Pro5 Lite. Updated: Command 15/115 – Identification packet. Updated: Command 17/117 – Set IO value. Updated: SMS commands *getdinmode, getio, info, setdinmode, setvalue*. Updated: Firmware and configuration file extensions. Updated: Supported communication command table. Updated: Supported SMS commands table.

1 Records

1.1 Record structure

All records have defined structure which can vary in length. Record consists of 2 parts: header with fixed length (23 bytes) and body with varying length (4-103 bytes). Maximum record size is 126 bytes. All data is in hex format.

Header [23 B]	Body [4-103 B]

1.2 Record header

All headers have the same parameters' fields. These fields are showed below.

Time stamp [4B]	Time stamp extension [1B]	Priority [1B]	Longitude [4B]	Latitude [4B]	Altitude [2B]	Angle [2B]	Satellites [1B]	Speed [2B]	HDOP [1B]	Event ID which generates record [1B]

1.2.1 Time stamp

Time stamp – difference, in seconds, between the current time and midnight, January 1, 1970 UTC (Unix time stamp: http://www.unixtimestamp.com). Parameter length – 4 bytes.

1.2.2 Time stamp extension

Time stamp extension – an extra byte to separate records with same time stamp. If some records have same time stamp when time stamp extension will increase starting with zero <0x00>. If there are no records with same time stamp parameter will always be zero. Parameter length – 1 byte.

See more: *1.5 Extended records*.

1.2.3 Priority

Priority can be low or high. It depends on configuration of event which triggered record. Parameter length – 1 byte.

Priority	Explanation
High	Event is configured as high priority. Initiates data sending instantly. Value <0x01>
Low	Event is configured as low priority. Value <0x00>

1.2.4 GPS element – Longitude

It is a GPS element. Together with latitude and altitude it reveals the position of an object. Parameter length – 4 bytes. Longitude is a signed integer number which calculated by formula:

$$long = (d + \frac{m}{60} + \frac{s}{3600} + \frac{ms}{3600000}) \times p \qquad (1).$$

d – Degrees, m – Minutes, s – Seconds, ms – Milliseconds, p – Precision = 10000000. Result is multiplied by –1 if longitude is in west.

Example:

Hexadecimal value 0xD468914A is received, in binary system it would be 1101 0100 0110 1000 1001 0001 0100 1010. So we see, that first bit is 1 - it means, that number will be negative (with minus). Converted to a signed integer, the result will be: -731344566 (to convert the following number to a coordinate, it must be divided by 10000000: -73.1344566)

1.2.5 GPS element – Latitude

It is GPS element. Together with longitude and altitude it reveals the position of an object. Parameter length – 4 bytes. Parameter length – 4 bytes. Latitude is a signed integer number which calculated by formula:

$$lat = (d + \frac{m}{60} + \frac{s}{3600} + \frac{ms}{3600000}) \times p \qquad (2).$$

d – Degrees, m – Minutes, s – Seconds, ms – Milliseconds, p – Precision = 10000000. Result is multiplied by –1 if latitude is in south.

Example:

Hexadecimal value 0xEA028883 is received, in binary system it would be 1110 1010 0000 0010 1000 1000 1000 0011. So we see, that first bit is 1 - it means, that number will be negative (with minus). Converted to a signed integer, the result will be: -368932733 (to convert the following number to a coordinate, it must be divided by 10000000: -36.8932733)

1.2.6 GPS element – Altitude

Parameter is in meters above sea level. Value is multiplied by 10. Parameter length – 2 bytes.

1.2.7 GPS element – Angle

Parameter is in degrees. Value zero <0x0000> is north, increasing clock-wise. Value is multiplied by 100. Parameter length – 2 bytes.

1.2.8 GPS element – Satellites

It is a number of visible GPS or GLONASS satellites (depends on device configuration). Parameter length – 1 byte.

1.2.9 GPS element – Speed

Object's current speed in km/h. Parameter length – 2 bytes.

1.2.10 GPS element – HDOP

Horizontal Dilute Of Precision parameter is a factor in determining the relative accuracy of a horizontal position. Value is multiplied by 10. The smaller the DOP number, the better the geometry. (more:

http://en.wikipedia.org/wiki/Dilution_of_precision_(GPS)). Value is multiplied by 10. Parameter length – 1 byte.

If the FM device at the time when the record was generated did not have valid coordinates (there were no GPS/GLONASS fix in the moment of data acquisition) then parameters *Longitude*, *Latitude, Altitude, Angle* values would be last valid fix. *HDOP*, *Satellites* and *Speed* would be cleared to 0.

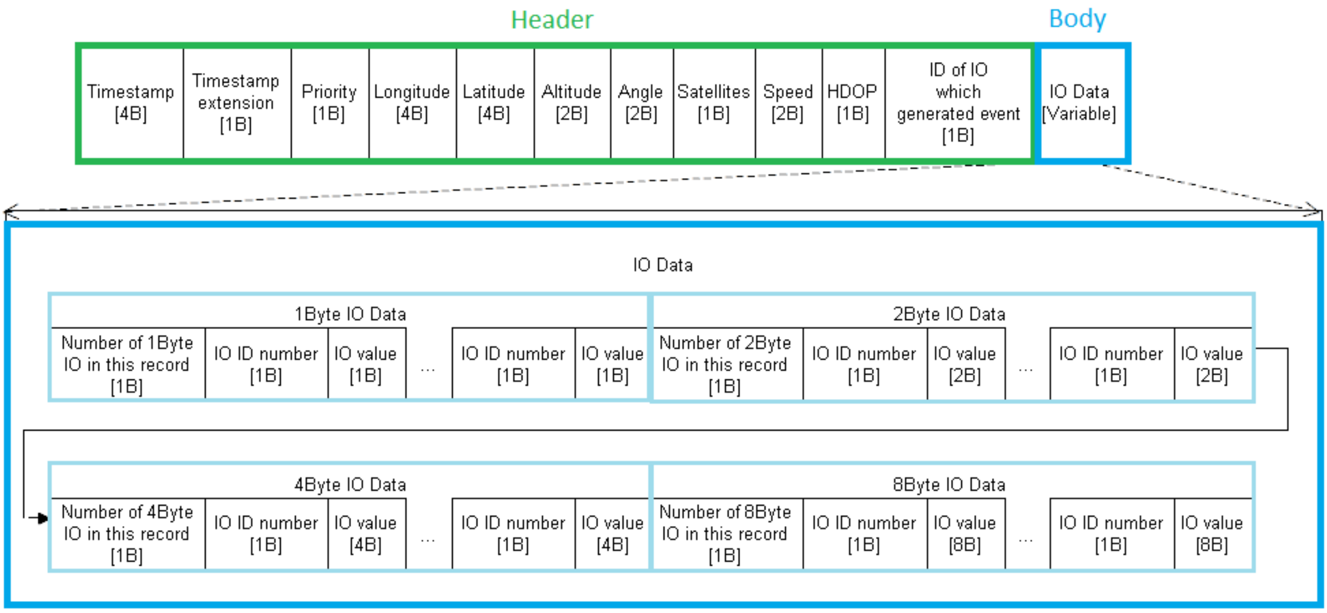
Note
In cases when GSM tracking functionality is enabled in the FM device, the HDOP value of received records will always be 0XFE (254 in decimal), because the coordinates are approximate.

1.2.11 Event ID which generates record

Parameter indicates why the record was created. Value is event ID number. Parameter length – 1 byte.

1.3 Record body

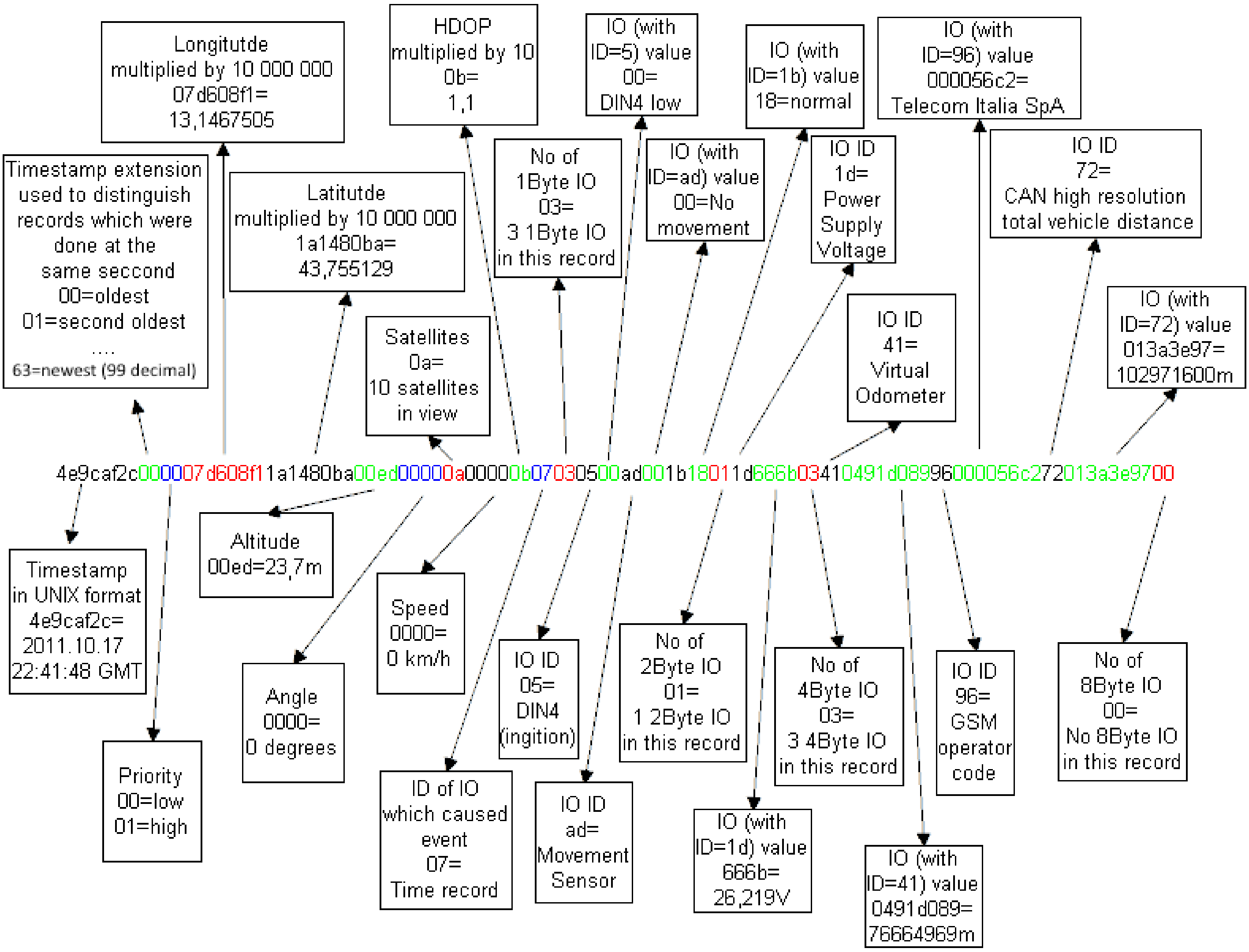
Record body can vary in length. Length depends on configuration of the FM device. Body is divided into segments of different length parameters (1, 2, 4 and 8 bytes long). Every segment holds the number of parameters with same size and ID numbers. Record structure is shown below.



When data collection without GPS fix is enabled and no GPS fix acquired, all GPS related fields which can be positive or negative will have <0x80..00> value. Fields which can only be positive will have <0xFF..FF> value.

1.4 Record example

Record example with explanations is shown below.



1.5 Extended records

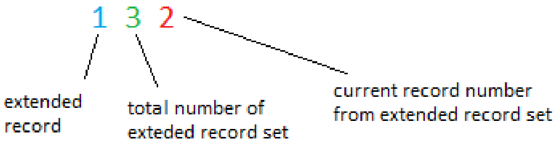
The size of all possible IO events is larger then a single record maximum size. That is why there is a technique to add many IO events in several records (with same header). Time stamp extension value must be interpreted as a decimal number of three digits – [a b c]. Each one of the 3 digits represents different information.

a) First digit (a) can be 0 or 1. Number 1 means that the record size of the current time stamp was exceeded and the records should be merged. Number 0 means that the record size of the current time stamp was not exceeded and no data have to be merged.

b) Second digit (b) can vary from 0 to 9. This digit represents how many records have to be merged of the current time stamp. For ex.: 0 means that there are total number of 1 record which has to be merged. Number 9 means that there are total number of 10 records which have to be merged. The maximum number of 10 records can be merged.

c) Third digit (c) can vary from 0 to 9. This digit represents which record of the current time stamp is of total (b) number of records. 0 means the first record. 9 means 10th record.

Example number of time stamp extension (0x84 = 132):


1 3 2
extended record
total number of exteded record set
current record number from extended record set

1.6 Record summary

Note
Extended protocol is mandatory for some of the IO parameters. The full list of IO parameters is available at the documentation website in the “FMIODATA+size+description” file. Parameters that require the use of the v1.1 protocol start from ID no. 256.

Record structure summary is shown in the table below.

Group	Parameter	Size	Description
Time	Time stamp	4 Bytes	UNIX time stamp
	Time stamp extension	1 Byte	Virtual milliseconds, enables to identify multiple records in one second. If three records are collected in one second first record will have time extension 00, second record will have time extension 01, third record will have time extension 02.
	Priority	1 Byte	High/Low
GPS/GLONASS	Longitude	4 Bytes	Longitude value multiplied by 10000000
	Latitude	4 Bytes	Latitude value multiplied by 10000000
	Altitude	2 Bytes	Altitude value multiplied by 10
	Angle	2 Bytes	Angle multiplied by 100
	Satellites	1 Byte	Number of satellites in use
	Speed	2 Bytes	Speed over ground km/h
	HDOP	1 Byte	HDOP multiplied by 10
	IO Data caused record	1 Byte	IO Data ID which caused record
IO element	No. of IO data 1Byte	1 Byte	Number of IO data elements which value is 1 Byte length
	IO Data ID	1 Byte	IO Data ID which value is 1 Byte

	IO Data value	1 Byte	IO Data value which value is 1 Byte length
	...	...	
	No. of IO data 2Byte	1 Byte	Number of IO data elements which value is 2 Byte length
	IO Data ID	1 Byte	IO Data ID which value is 2 Byte length
	IO Data value	2 Bytes	IO Data value which value is 2 Byte length
	...	...	
	No. of IO data 4Byte	1 Byte	Number of IO data elements which value is 4 Byte length
	IO Data ID	1 Byte	IO Data ID which value is 4 Byte length
	IO Data value	4 Bytes	IO Data value which value is 4 Byte length
	...	...	
	No. of IO data 8Byte	1 Byte	Number of IO data elements which value is 8 Byte length
	IO Data ID	1 Byte	IO Data ID which value is 8 Byte length
	IO Data value	8 Bytes	IO Data value which value is 8 Byte length
	...	...	

2 Extended Protocol Records

This chapter will cover only fields that are different from original record protocol described in Chapter 1.

2.1 Record structure

All records have defined structure which can vary in length. Record consists of 2 parts: header with fixed length (25 bytes) and body with varying length (4-101 bytes). Maximum record size is 126 bytes.

Header [25 B]	Body [4-101 B]

All data is in hex format.

2.2 Record header

All headers have the same parameters' fields. These fields are shown below.

Time stamp [4B]	Time stamp extension [1B]	Record extension [1B]	Priority [1B]	Longitude [4B]	Latitude [4B]	Altitude [2B]	Angle [2B]	Satellites [1B]	Speed [2B]	HDOP [1B]	Event ID which generates record [2B]

2.2.1 Time stamp extension

Time stamp extension – an extra byte to separate records with same time stamp. If some records have same time stamp when time stamp extension will increase starting with zero <0x00>. If there are no records with same time stamp parameter will always be zero. Parameter length – 1 byte.

See more: *2.4 Extended protocol extended records*.

2.2.2 Record extension

Record extension – an extra byte to separate records with same time stamp and time stamp extension. If some data does not fit into one record it is indicated by Record extension field. Parameter length – 1 byte.

See more: *2.4 Extended protocol extended records*.

2.2.3 Event ID which generates record

Parameter indicates why the record was created. Value is event ID number. Parameter length – 2 bytes (big endian).

2.3 Record body

Record body can vary in length. Length depends on configuration of the FM device. Body is divided into segments of different length parameters (1, 2, 4 and 8 bytes long). Every segment holds the number of parameters with same size and ID numbers. For extended record protocol IO ID numbers are 2 bytes long (big endian).

2.4 Extended protocol extended records

The size of all possible IO events is larger then a single record maximum size. That is why there is a technique to add many IO events in several records (with same header). Record extension value must be interpreted as a hexadecimal number of two digits – [0xmn]. Each one of the 2 BCD digits represents different information.

a) First BCD digit (m) can vary from 0 to 7. This digit represents how many records have to be merged of the current time stamp. For ex.: 0 means that there are total number of 1 record which has to be merged. Number 7 means that there are total number of 8 records which have to be merged.

b) Second BCD (n) can vary from 0 to 7. This digit represents which record of the current time stamp is of total (m) number of records. 0 means the first record. 7 means 8th record.

Example number of time stamp extension:

- 0x74 => 5th record of 8 records set;
- 0x20 => 1th record of 3 records set;
- 0x00 => 1th record of 1 record set (this is not an extended record);

2.5 Extended Protocol Record summary

Record structure summary is shown in the table below.

Group	Parameter	Size	Description
Time	Time stamp	4 Bytes	UNIX time stamp
	Time stamp extension	1 Byte	Virtual milliseconds, enables to identify multiple records in one second. If three records are collected in one second first record will have time extension 00, second record will have time extension 01, third record will have time extension 02.
	Record extension	1 Byte	Extended record indicator
	Priority	1 Byte	High/low
GPS/GLONASS	Longitude	4 Bytes	Longitude value multiplied by 10000000
	Latitude	4 Bytes	Latitude value multiplied by 10000000
	Altitude	2 Bytes	Altitude value multiplied by 10

	Angle	2 Bytes	Angle multiplied by 100
	Satellites	1 Byte	Number of satellites in use
	Speed	2 Bytes	Speed over ground km/h
	HDOP	1 Byte	HDOP multiplied by 10
	IO Data caused record	2 Byte	IO Data ID which caused record
IO element	No. of IO data 1Byte	1 Byte	Number of IO data elements which value is 1 Byte length
	IO Data ID	2 Byte	IO Data ID which value is 2 Byte
	IO Data value	1 Byte	IO Data value which value is 1 Byte length
	…	…	
	No. of IO data 2Byte	1 Byte	Number of IO data elements which value is 2 Byte length
	IO Data ID	2 Byte	IO Data ID which value is 2 Byte length
	IO Data value	2 Bytes	IO Data value which value is 2 Byte length
	…	…	
	No. of IO data 4Byte	1 Byte	Number of IO data elements which value is 4 Byte length
	IO Data ID	2 Byte	IO Data ID which value is 2 Byte length
	IO Data value	4 Bytes	IO Data value which value is 4 Byte length
	…	…	
	No. of IO data 8Byte	1 Byte	Number of IO data elements which value is 8 Byte length
	IO Data ID	2 Byte	IO Data ID which value is 8 Byte length
	IO Data value	8 Bytes	IO Data value which value is 8 Byte length
	…	…	

Note

The extended protocol is mandatory for some of the IO parameters. The full list of IO parameters is available at the documentation website in the “FMIODATA+size+description” file. Parameters that require the use of the v1.1 protocol start from ID no. 256.

3 FM & Server protocol

3.1 Protocol Structure

There is a standard Ruptela protocol for communication between FM device and server. All messages between server and device are sent by this standard. General protocol description is shown below.

Device to server:

Field	Packet length	IMEI	Command ID	Payload Data	CRC16
Size (bytes)	2	8	1	[1-1011]	2

Server to device:

Field	Packet length	Command ID	Payload data	CRC16
Size (bytes)	2	1	[1-1018]	2

Remark:

- Difference is that server does not have IMEI field.
- Maximum data packet size is 1 kB (1024 bytes).

3.1.1 Packet length

Parameter indicates the size of all packet size: all fields except itself and CRC16.

3.1.2 IMEI

IMEI is a 64 bit variable. It is unique for every FM device.

3.1.3 Command

Command ID is a 8 bit variable. Command ID describes what type of command is received.

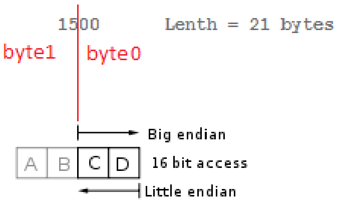
3.1.4 Payload

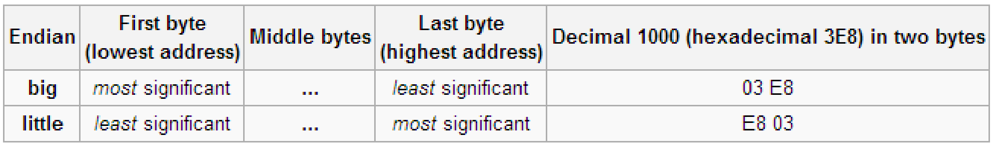
All data which is sent.

3.1.5 CRC16

CRC16 is calculated using CRC-CCITT (Kermit) algorithm (http://www.lammertbies.nl/comm/info/crc-calculation.html).

Parameter length – 2 bytes. Format is big endian. That is why bytes 0 and 1 are “switched”.





Endian	First byte (lowest address)	Middle bytes	Last byte (highest address)	Decimal 1000 (hexadecimal 3E8) in two bytes
big	*most* significant	...	*least* significant	03 E8
little	*least* significant	...	*most* significant	E8 03

Note: CRC is calculated without including the packet length.

CRC16 calculation algorithm is shown below (C programming language):

```
/*---------------------------------------------------------------------------
 *      FUNCTION: CRC16
 *---------------------------------------------------------------------------*/
unsigned short crc_16_rec (unsigned char *pucData, unsigned short ucLen) {
        //--------------------------------------------------------------------
        unsigned int i;
        unsigned char ucBit, ucCarry;
        //--------------------------------------------------------------------
        unsigned short usPoly = 0x8408;//reversed 0x1021
        unsigned short usCRC  = 0;
        //--------------------------------------------------------------------
        for (i = 0; i < ucLen; i++) {
        usCRC ^= pucData[i];
        for (ucBit = 0; ucBit < 8; ucBit++) {
            ucCarry = usCRC & 1;
            usCRC >>= 1;
            if (ucCarry) {
                usCRC ^= usPoly;
            }
        }
    }
        //--------------------------------------------------------------------
        return usCRC;
        //--------------------------------------------------------------------
}
```

3.1.6 CRC8

CRC8 is used when sending IO packet data via the RS232 interface using the transparent channel functionality.

CRC8 calculation algorithm is shown below (C programming language):

```
unsigned char crc_8(char *pucData, unsigned short ucLen) {
    //---------------------------------------------------------------------
    unsigned int  i;
    unsigned char ucBit, ucCarry;
    //---------------------------------------------------------------------
    unsigned char ucPoly = 0xE0; // reversed 0x07
    unsigned char ucCRC  = 0;
    //---------------------------------------------------------------------
    for (i = 0; i < ucLen; i++) {
        ucCRC ^= pucData[i];
        for (ucBit = 0; ucBit < 8; ucBit++) {
            ucCarry = ucCRC & 1;
            ucCRC >>= 1;
            if (ucCarry) {
                ucCRC ^= ucPoly;
            }
        }
    }
    //---------------------------------------------------------------------
    return ucCRC;
    //---------------------------------------------------------------------
}
```

3.2 Communication commands

Communication between FM device and server always is initiated by FM device. There is a possibility to force the FM device to connect to the server by sending an SMS command (see *SMS commands: connect, econnect*).
There is always a response message from receiver.

3.2.1 Command 1/100 – Records

FM device uses command 1 (0x01). FM device sends records according to the format which is shown below.

Field	Packet length	IMEI	Command ID	Payload			CRC16
				Records left	Number of records	Data (see *Records*)	
Size (bytes)	2	8	1	1	1	[1-1009]	2
Example hex	0113	00000B1A29F64B1A	01	00	0C	12 records data...	ADE9
Example dec	275	12207001062170	1	0	12	12 records data...	-

Records left flag - Two possible values: 0 or 1. 0 – there are no records left in flash, 1 – there are records left in flash.

Number of records - Describes how many records are in Record DATA field.

Data - Records with variable length. Record consists of non variable length field: time stamp, priority, GPS elements and variable length field: IO elements.

Server should use command 100 (0x64) for records response (acknowledge). Answer from server:

Field	Packet length	Command	ACK	CRC16
Size (bytes)	2	1	1	2
Example hex	0002	64	01	13BC
Example dec	2	100	1	-

ACK possible values: 0 – negative acknowledgement (NACK), 1 – positive acknowledgement (ACK).

When positive acknowledgement (ACK) is received, the device deletes all sent records from the memory. When negative acknowledgement (NACK) is received, the sending of record packets is delayed. If record packets are continuously rejected by the server, the delay period increases incrementally by 1, 5, 10, 15, 30 and 60 minutes after each NACK. After a delay of 60 minutes is reached, it will no longer increase and remain for subsequent packets. The delay is reset to 0 if one of the following conditions is met:

- If the server responds to a record packet with an ACK;

- If the device is restarted;
- If the connection to the server was initiated with SMS commands “econnect”, “connect”, “switchip” or GPRS command 105 “Set connection parameters”;
- If parameters IP1, port1, IP2 or port2 were changed by SMS commands “setcfg” or “setconnection” or by uploading the configuration file via USB.

Example of records data packet (raw data 825 bytes):

```
033500000C076B5C208F01011E5268CEF20000196E3A3A0AEF3E934F3E2D780000000007000000005268CEFD0000196E3A3A0AEF3
E934F3E2D780000000007000000005268CF080000196E3A3A0AEF3E934F3E2D780000000007000000005268CF130000196E3A3A0A
EF3E934F3E2D780000000007000000005268CF1E0000196E3A3A0AEF3E934F3E2D780000000007000000005268CF290000196E3A3
A0AEF3E934F3E2D780000000007000000005268CF340000196E3A3A0AEF3E934F3E2D780000000007000000005268CF3F0000196E3
A3A0AEF3E934F3E2D780000000007000000005268CF4A0000196E3A3A0AEF3E934F3E2D780000000007000000005268CF55000019
6E3A3A0AEF3E934F3E2D780000000007000000005268CF600000196E3A3A0AEF3E934F3E2D780000000007000000005268CF6B000
0196E3A3A0AEF3E934F3E2D780000000007000000005268CF730000196E36630AEF42CE4F6D0BF40400022208000000005268CF7E0
000196E36B60AEF42BE4F6D0BF40000000007000000005268CF890000196E36B60AEF42BE4F6D0BF40000000007000000005268CF9
40000196E36B60AEF42BE4F6D0BF40000000007000000005268CF9F0000196E36B60AEF42BE4F6D0BF40000000007000000005268C
FAA0000196E36B60AEF42BE4F6D0BF40000000007000000005268CFB50000196E36B60AEF42BE4F6D0BF4000000000700000000052
68CFC00000196E36B60AEF42BE4F6D0BF40000000007000000005268CFCB0000196E36B60AEF42BE4F6D0BF400000000070000000
05268CFD60000196E36B60AEF42BE4F6D0BF40000000007000000005268CFD70000196E3C710AEF5EFF4F690BF4040001170800000
0005268CFE20000196E3B980AEF601A4F690BF40000000007000000005268CFED0000196E3B980AEF601A4F690BF40000000007000
000005268CFF80000196E3B980AEF601A4F690BF40000000007000000005268D0030000196E3B980AEF601A4F690BF400000000070
00000005268D00E0000196E3B980AEF601A4F690BF40000000007000000005268D0190000196E3B980AEF601A4F690BF400000000
07000000005268D0240000196E3B980AEF601A4F690BF4000000000700000000046E2
```

Packet length – 0x0335 = 821

IMEI – 0x00000C076B5C208F = 13226005504143

Command ID – 0x01 = 1

Records left in device's flash memory – 0x01 = 1

Number of records in packet – 0x1E = 30

30 Records...

CRC16 – 0x46E2 = 18146

3.2.2 Command 68/100 – Extended protocol records

FM device uses command 68 (0x44). FM device sends extended protocol records according to the format which is shown below.

Field	Packet length	IMEI	Command ID	Payload			CRC16
				Records left	Number of records	Data (see Extended Protocol *Records*)	
Size (bytes)	2	8	1	1	1	[1-1009]	2
Example hex	0113	00000B1A29F64B1A	44	00	0C	12 records data...	ADE9
Example dec	275	12207001062170	68	0	12	12 records data...	-

Records left flag - Two possible values: 0 or 1. 0 – there are no records left in flash, 1 – there are records left in flash.

Number of records - Describes how many records are in Record DATA field.

Data - Records with variable length. Record consists of non variable length field: time stamp, priority, GPS elements and variable length field: IO elements.

Server should use command 100 (0x64) for records response (acknowledge). Answer from server:

Field	Packet length	Command	ACK	CRC16
Size (bytes)	2	1	1	2
Example hex	0002	64	01	13BC
Example dec	2	100	1	-

ACK possible values: 0 – negative acknowledgement (NACK), 1 – positive acknowledgement (ACK).

When positive acknowledgement (ACK) is received, the device deletes all sent records from the memory. When negative acknowledgement (NACK) is received, the sending of record packets is delayed. If record packets are continuously rejected by the server, the delay period increases incrementally by 1, 5, 10, 15, 30 and 60 minutes after each NACK. After a delay of 60 minutes is reached, it will no longer increase and remain for subsequent packets. The delay is reset to 0 if one of the following conditions is met:

- If the server responds to a record packet with an ACK;
- If the device is restarted;
- If the connection to the server was initiated with SMS commands “econnect”, “connect”, “switchip” or GPRS

command 105 “Set connection parameters”;

- If parameters IP1, port1, IP2 or port2 were changed by SMS commands “setcfg” or “setconnection” or by uploading the configuration file via USB.

3.2.3 Command 2/102 – Device Configuration Data

This command is used to configure the FM device: send configuration packets & control messages (see full description *5 Configuration*). Data in payload field is terminated with <0x0D><0x0A>. Server uses command 102 (0x66).

Field	Packet length	Command ID	Payload	CRC16
Size (bytes)	2	1	Not fixed	2
Example hex	000E	66	236366675F7374617274400D0A	0947
Example dec	14	102	-	-
Example ASCII	-	-	#cfg_start@	-

FM device uses command 2 (0x02) to send response.

Field	Packet length	IMEI	Command ID	Payload	CRC16
Size (bytes)	2	8	1	Not fixed	1
Example hex	16	0003124D0AC0BB1C	02	406366675F7374732331300D0A	4B58
Example dec	22	864547032316700	2	-	-
Example ASCII	-	-	-	@cfg_sts#10	-

3.2.4 Command 3/103 – Device Version Info

Server uses command 103 (0x67). Command asks for the information about current firmware, hardware versions.

Field	Packet length	Command ID	CRC16
Size (bytes)	2	1	2
Example hex	0001	67	17B9
Example dec	1	103	-

FM device responds with command 3 (0x03).

Field	Packet length	IMEI	Command ID	Payload	CRC16
Size (bytes)	2	8	1	Not fixed	2
Example hex	22	0003124D0AC0BB1C	03	353432432C30302E30332E30392E31302C3636382C32322C31	E7EC
Example dec	34	864547032316700	3	-	-
Example ASCII	-	-	-	542C,00.03.09.10,668,22,1	-

Payload content example:

- Bootloader version - **542C**,00.03.09.10,668,22,1
- Firmware version - 542C,**00.03.09.10**,668,22,1
- Hardware version - 542C,00.03.09.10,**668**,22,1
- GSM signal level – 542C,00.03.09.10,668,**22**,1 (ranges from 0 (no signal) to 31 (strong signal))
- Voltage status – 542C,00.03.09.10,668,22,**1** (0 – voltage too low, 1 – voltage OK)

3.2.5 Command 4/104 – Device Firmware Update

Command is used to update FM device's firmware: send configuration packets & control messages (see 6 *Firmware*). Data in payload field is terminated with <0x0D><0x0A>. Server uses command 104 (0x68).

Field	Packet length	Command ID	Payload	CRC16
Size (bytes)	2	1	[1-1019]	2
Example hex	000C	68	7C46555F535452542A0D0A	B66B
Example dec	12	104	-	-
Example ASCII	-	-	|FU_STRT*	-

FM device uses command 4 (0x04).

Field	Packet length	IMEI	Command ID	Payload	CRC16
Size (bytes)	2	8	1	[7-10]	2
Example hex	0012	0003124D0AC0BB1C	04	2A46555F4F4B7C0D0A	75DB
Example dec	18	864547032316700	4	-	-
Example ASCII	-	-	-	*FU_OK|	-

3.2.6 Command 5/107 – Smart Card Data

Command 5 (0x05) is used by the FM device to send smart card data (fragments of DDD file) to server.

Field	Packet length	IMEI	Command ID	Payload	CRC16
Size (bytes)	2	8	1	[1-512]	2
Example hex	0012	000315A07F44865A	05	Raw data segment of DDD file	**CRC16**
Example dec	18	868204004279898	5	Raw data segment of DDD file	**CRC16**

Answer from server command 107 (0x6B).

Field	Packet length	Command	ACK	CRC16
Size (bytes)	2	1	1	2
Example hex	0002	6B	01	9074
Example dec	2	107	1	-

ACK possible values: 0 – negative acknowledgement, 1 – positive acknowledgement, 2 – card data rejected.

The decision that whole *.DDD* file is sent should be done by server. Server checks all received smart card data size. If the data size is equal to *smart card size parameter* sent by command 6 (smart card data size and time stamp) that means the end of DDD file. All smart card data packets' payloads are equal to 512 bytes except the last one. It is also an indication for the last packet (end of file).

3.2.7 Command 6/107 – Smart Card Data Size and Time stamp

FM uses this command 6 (0x06) to send information about smart card *.DDD* file and the time stamp when the file was created. For 2nd generation smart cards, command 19/107 (0x13) is used instead.

Field	Packet length	IMEI	Command ID	Payload		CRC16
				Size	Time stamp	
Size (bytes)	2	8	1	2 + 4 = 6		2
Example hex	000B	000315A07F44865A	06	5428	4E9CAF2C	F5B3
Example dec	11	868204004279898	6	21544	1318891308	-

Answer from server command 107 (0x6B).

Field	Packet length	Command	ACK	CRC16
Size (bytes)	2	1	1	2
Example hex	0002	6B	01	9074
Example dec	2	107	1	-

ACK possible values: 0 – negative acknowledgement, 1 – positive acknowledgement, 2 – card data rejected.

3.2.8 Command 7/108 – SMS via GPRS

Command is used to send SMS messages to the device via GPRS. Message text in payload is the same as a message via GSM network. Command 108 (0x6C) is used by the server. The maximum message length is 160 characters.

Field	Packet length	Command ID	Payload*	CRC16
Size (bytes)	2	1	Not fixed	2
Example hex	000A	6C	736574696F20322C31	59B9
Example dec	10	108	-	-
Example ASCII	-	-	setio 2,1	-

Command 7 (0x07) is used by the device to send a response back to the server.

Field	Packet length	IMEI	Command ID	Payload*	CRC16
Size (bytes)	2	8	1	Not fixed	2
Example hex	0024	0003124D0AC0BB1C	07	534554494F20636F6E666967757261746 96F6E2064617461206F6B	C419
Example dec	36	864547032316700	7	-	-
Example ASCII	-	-	-	SETIO configuration data ok	-

3.2.9 Command 9/109 – Diagnostic Trouble Codes

Command is used to send vehicle Diagnostic Trouble Codes (DTCs). FM device initiates sending session and server has to acknowledge packets. FM device starts to send DTC when it finds any changes in vehicle's diagnostic trouble codes (at least one new trouble code should appear or existing code should disappear). DTCs read frequency depends on configuration. Command 9 (0x09) is used by FM device. All currently read DTCs gains status – current (0x01). All others have DTC status – history (0x02).

Field	Packet length	IMEI	Command ID	Payload		CRC16
				Number of DTC in packet	DTCs	
Size (bytes)	2	8	1	1	Not fixed	2
Example hex	0030	00000B1A29F64B1 A	09	02	FF4E9CAF2C07D608F11A1480BA 015030303130FF4E9CAF2C07D6 08F11A1480BA025030303131	8C91
Example dec	48	12207001062170	9	2	*	-

*There are two DTC packets in this example's payload *DTCs* field.
1st packet: Time stamp – 1318891308; Longitude – 13,1467505; Latitude – 43,7551290; DTC status – current DTC; DTC – P0010.
2nd packet: Time stamp – 1318891308; Longitude – 13,1467505; Latitude – 43,7551290; DTC status – history DTC; DTC – P0011.

Payload fields are described below.

<table>
<tr><th>Field</th><th>Description</th></tr>
<tr><td><b>Number of DTC in packet</b></td><td>Indicates how many separate Diagnostic Troubles Codes are in payload's sub field “DTCs”.</td></tr>
<tr><td><b>DTCs</b></td><td>This field holds Diagnostic Trouble Codes in packets shown below.
DTC packet:
<table>
<tr><th>Field</th><th>Length (bytes)</th><th>Description</th></tr>
<tr><td>DTC source</td><td>1</td><td>0xFF – OBD
0x01 – J1939
0x02 - J1708</td></tr>
<tr><td>Time</td><td>4</td><td>UNIX time stamp when the DTC was read.</td></tr>
<tr><td>Longitude</td><td>4</td><td>Longitude when DTC was read.</td></tr>
<tr><td>Latitude</td><td>4</td><td>Latitude when DTC was read.</td></tr>
<tr><td>DTC status</td><td>1</td><td>Indicates the status of particular diagnostic trouble code:
1 – current DTC (vehicle still have it).
2 – history DTC (vehicle had this DTC at specified time (field '<i>Time</i>') in the past).</td></tr>
<tr><td>OBD DTC (text)
J1939 and J1708 DTC (hex)</td><td>5</td><td>For OBD data source diagnostic trouble code is in ASCII format (text).
For J1939 and J1708 data sources diagnostic trouble code is in HEX format.</td></tr>
</table>
</td></tr>
</table>

Server answer with command 109 (0x6D).

Field	Packet length	Command	ACK	CRC16
Size (bytes)	2	1	1	2
Example hex	0002	6D	01	C4A4
Example dec	2	109	1	-

ACK field can have three possible values. 0 – negative acknowledgement, 1 – positive acknowledgement, 2 – request DTC data (start force DTC data sending).

3.2.10 Command 10/110 – Tachograph Communication

Command is used to start reading (tachograph initialization) tachograph data via FM device. Command packet has extra fields compared to other FM – server commands. These fields in data packet from FM device are described below.

Field	Description
Status	Bit field tachograph task status. Status first byte indicating which object from tachograph is available. First byte is bit-field value. Bits indication: 1 – tachograph available, 2 – Card in card slot 1 is available, 3 – Card in card slot 2 is available.
Command SubID	1 – Status response (long status) in DATA field ***(Reserved for future use).***
	2 – ATR response (***in case of error***).
	3 – APDU packet in SubID payload field.
	4 – Finished authenticate.
	5 – Tachograph read period.
	6 – Unused.
	7 – Process stop response (end communication with tacho)
	8 – Upload data to tachograph.
	9 – Reading Tacho Data process (for error status).
	10 – Reading Card Data from Tacho process (for error status).
Packet Status	0 – NACK (not acknowledged).
	1 – ACK (acknowledgement OK).
	2 – ERROR.
	3 – Task not created.
	4 – Timeout (response from tachograph).
	5 – SubID incorrect (not founded).
	6 – Another tacho data file is in flash. Task not created.
	7 – Response from server timeout.
	8 – no tacho task.

	9 – interface extender not available.
	10 – wrong communication settings in FM device.
	11 – all repeats are exceeded, read failed.
SubID payload	SubID payload data or nothing (if NACK of ERROR is in packet's *Status* field).

Tachograph read routine is initiated by server using command 110 (0x6E), SubID 2 (0x02). After this FM device takes control of communication and server needs only to respond correctly. According to *SubID* command value *SubID payload* data can give different kind of information.

Server command 110 (0x6E):

Field	Packet length	Command ID	Command SubID	Reserved	SubID command payload	CRC16
Size (bytes)	2	1	1	2	Not fixed	2
Example hex	0020	6E	02	0000	3B9A96C01031FE5D0064057B01023180900076015130B20B518CFB07	A416
Example dec	32	110	2	0000		-

Name	File ID	Digital signature	Changing	Sending
EF ICC	0002	–	–	*
EF IC	0005	–	–	*
EF Application Identification	0501	+	–	*
EF Card Certificate GOST	C200	–	–	+
EF Key Identificators	C201	+	–	*
EF CA Certificate GOST	C208	–	–	*
EF Identification	0520	+	–	–
EF Card Download	050E	+	+	+
EF Driving Licence Info	0521	+	–	*
EF Events Data	0502	+	+	+
EF Faults Data	0503	+	+	+
EF Driver Activity Data	0504	+	+	+
EF Vehicles Used	0505	+	+	+
EF Places	0506	+	+	+
EF Current Usage	0507	+	+	+
EF Control Activity Data	0508	+	+	+
EF Specific Conditions	0522	+	+	+

(+) mandatory to send
(-) forbidden to send
(*) optional to send

FM device command 10 (0x0A):

Field	Packet length	IMEI	Command ID	Status	Command SubID	Packet Status	Reserved	SubID command payload	CRC16
Size (bytes)	2	8	1	4	1	1	2	Not fixed	2
Example hex	0018	000315A07F440B1D	0A	0	03	1	0	00A4020C020002	65D7
Example dec	24	868204004248349	10	0	3	1	0		-

Possible status values:

Decimal value	Explanation
0	OK
-1	Bad value
-2	Bad parameter
-3	Timeout
-4	Negative answer
-5	Busy
-6	Driver card in flash
-7	Bad SubID
-8	Internal error

After successful initialization FM device will inform server about tachograph packet data with command 12 (0x0C). Server should answer ACK message with command 111 (0x6F). *DDD* file packets is then send by FM device to server using command 11 (0x0B). Server should send ACK message (0x6F) to every FM device message. Flowchart for getting tachograph data is shown in file: *Server-FM Tacho_authentication_and_read_Flowchart_02.png*.

A more detailed explanation can be found in the *DOC_Tacho read detailed protocol.xls* document (*confidential*). Contact tech. support to gain access to this file.

3.2.11 Command 11/111 – Tachograph Data Packet

Command is used to send *.DDD* file packets from FM device to server. FM device uses command 11 (0x0B).

Field	Packet length	IMEI	Command ID	Packet index	Payload	CRC16
Size (bytes)	2	8	1	2	[1-1009]	2
Example hex	03FC	000315A07F440B1D	0B	0000	1st fragment of *.DDD* file data	**CRC16**
Example dec	1020	868204004248349	11	0	1st fragment of *.DDD* file data	-

Example raw data of this packet:

```
03 FC 00 03 15 A0 7F 44 0B 1D 0B 00 01 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 00 20 20 20 20 20 20 20 20 20 20 20
20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 02 15 53 56 54 20 30 36 20 20 30 30 31 36 20 30 30 30 FF FF
FF FF 51 39 D5 11 00 00 00 00 51 39 D5 89 00 00 00 15 01 41 42 43 31 32 33 20 20 20 20 20 20 20 51 39 D3 FE 00 00 08 20 00 02 CD
22 D3 02 D4 22 D5 02 D9 22 DB A0 00 03 02 15 53 56 54 20 30 36 20 20 30 30 31 36 20 30 30 30 51 39 D3 81 01 15 00 00 00 00 02 15
53 56 54 20 30 36 20 20 30 30 31 36 20 30 30 30 51 39 D3 FE 01 15 00 00 00 00 02 15 53 56 54 20 30 36 20 20 30 30 31 36 20 30 30
30 51 39 D5 88 01 15 00 00 00 00 00 00 5A 7C D4 64 54 F9 6F FD 17 18 50 48 20 8D 99 69 74 A0 2A E7 17 82 B6 4D F1 99 25 DA E6 04
02 F6 48 D2 F9 AE 45 6D 94 09 BE 73 3A 3D 69 5A 16 DE 44 D6 B9 FE DE 9A B7 FC D8 7B AC 38 2D 28 59 4E 73 94 E0 54 41 09 DB F5
7A DF 57 58 6E BF 4A D9 20 9B B3 3C EC 61 F7 5C 82 47 49 80 49 ED 63 20 A8 45 25 92 6A 1F EF AB 92 37 CC 81 EC 19 78 16 7F DE BF
F9 CA D0 54 19 BE 5F 65 87 F3 7D 56 A7 76 02 51 47 AA 7F 00 00 08 00 04 01 4D 75 73 74 65 72 66 72 61 75 20 34 33 20 20 20 20 20
20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 01 4B 72 69 73 74 61 20 34 33 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
20 20 20 20 20 20 20 20 20 20 01 12 33 31 30 30 30 30 30 30 30 36 31 33 30 30 30 30 63 D5 B7 00 51 47 17 28 00 00 00 00 51 47 17
B1 00 00 00 00 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF 00 01 4D 75 73 74 65 72 66 72 61 75 20 34 33 20 20 20 20 20 20
20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 01 4B 72 69 73 74 61 20 34 33 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
20 20 20 20 20 20 20 20 20 01 12 33 31 30 30 30 30 30 30 30 36 31 33 30 30 30 30 63 D5 B7 00 51 47 18 63 00 00 00 00 51 47 19 04
00 00 02 15 01 41 42 43 31 32 33 20 20 20 20 20 20 20 51 47 17 B0 00 01 4D 75 73 74 65 72 66 72 61 75 20 34 33 20 20 20 20 20 20
20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 01 4B 72 69 73 74 61 20 34 33 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
20 20 20 20 20 20 20 20 20 01 12 33 31 30 30 30 30 30 30 30 36 31 33 30 30 30 30 63 D5 B7 00 51 47 2A 6F 00 00 02 00 51 47 2B 61
00 00 02 15 01 41 42 43 31 32 33 20 20 20 20 20 20 20 51 47 19 03 00 01 4D 75 73 74 65 72 66 72 61 75 20 34 33 20 20 20 20 20 20
20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 01 4B 72 69 73 74 61 20 34 33 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
20 20 20 20 20 20 20 20 20 01 12 33 31 30 30 30 30 30 30 30 36 31 33 30 30 30 30 63 D5 B7 00 51 47 3D 3C 00 00 02 00 51 47 3E FB
00 00 08 15 01 41 42 43 31 32 33 20 20 20 20 20 20 20 51 47 2B 60 00 00 14 20 00 03 2B 23 2D 03 30 1B 31 23 33 03 7D 1B 80 23 81
03 CD 1B CE 03 D1 23 D5 A0 00 AB 31 A3 33 AB 80 A3 81 AB CE A3 D1 03 01 12 33 31 30 30 30 30 30 30 30 36 31 33 30 30 30 30 51
47 19 03 01 15 00 00 00 02 01 12 33 31 30 30 30 30 30 30 30 36 31 33 30 30 30 30 51 47 2B 60 01 15 00 00 00 02 01 12 33 31 30 30
30 30 30 30 30 36 31 33 30 30 30 30 51 47 3E FA 01 15 00 00 00 08 00 00 05 B5 2F B4 AF 69 5C 30 C0 82 D5 78
```

Answer from server command 111 (0x6F).

Field	Packet length	Command ID	ACK	Packet index	CRC16
Size (bytes)	2	1	1	2	2
Example hex	0004	6F	01	0000	71C1
Example dec	4	111	1	0	-

3.2.12 Command 12/111 – Information Packet About Tachograph Data

Command is used to send information about tachograph data packet (see 2.2.9). Device uses command 12 (0x0C).

Field	Packet length	IMEI	Command ID	Payload							CRC16
				Data storage type	Data size	Data read time stamp	Data period start	Data period end	Data CRC16	Packet index	
Size (bytes)	2	8	1	1	4	4	4	4	2	2	2
Example hex	000B	00000B1A29F64B1A	0C	01	00013880	521C67CE	5130B20B	518CFB07	2A56	FFFF	39B0
Example dec	11	12207001062170	12	1	80000	1377593294	1362145803	1368193799	10838	65535	-

FM device command 12 payload explanation:

Sub field	Description
Data storage type	1 – in internal flash memory, 2 – in SD card memory.
Data size	Tachograph data size in flash or SD card (internal FM device memory).
Data read time stamp	Time stamp when tachograph was read (UNIX time stamp).
Data period start	Start date when tachograph data was read (UNIX time stamp).
Data period end	End date when tachograph data was read (UNIX time stamp).
Data CRC16	CRC16 of tachograph data (*.DDD*) which was read.
Packet index	Packet index. If data upload starts from beginning then index is 0xFFFF. Otherwise last index is '*last_index*-1'.

Server responds with command 111 (0x6F).

Field	Packet length	Command ID	ACK	Packet index	CRC16
Size (bytes)	2	1	1	2	2
Example hex	0004	6F	01	FFFF	8179
Example dec	4	111	1	65535	-

Extra fields are explained below:

Field	Description
ACK	Values: 1 – positive acknowledgement, 2 – data rejected & FM device will delete all information from its memory. Other values are reserved.
Packet index	Packet index for ACK. First packet (info about tacho data) has index FFFF (response with this index to first packet).

Packet index

Tachograph data transfer command 111 (0x6F) has packet index field. This index is used by server to indicate which tacho data packet is expected/acknowledged. Server receives packet with index 0xFFFF if new data download has been started. If communication between server and FM device is terminated (ex.: GPRS link was broken) when data download is still in progress (not ended). Next time FM device will send packet *"Information packet about tachograph data"* (command 12) packet index value will be '*last index-1*'. If Server respond to FM device with just received index (*last index-1*) then next packet which will be sent to server from FM device will be with index increased by 1. If server respond with different index value (compared to received index value) then FM device will send requested index packet.

Data flow example between server and FM device is shown below (command Ids are in bold):

Action*	Full packet
Info: FM to Server	00 1E 00 03 15 A0 7F 44 0B 1D **0C** 01 00 01 B3 BA 51 9F 2F D3 51 30 B2 0B 51 8C FB 07 00 00 FF FF 87 D4
Server to FM	00 04 **6F** 01 FF FF 81 79
FM to Server	03 FC 00 03 15 A0 7F 44 0B 1D **0B** 00 00 …
Server to FM	00 04 **6F** 01 00 00 71 C1
FM to Server	03 FC 00 03 15 A0 7F 44 0B 1D **0B** 00 01 20 20 20 20 20 **…**
Server to FM	00 04 **6F** 01 00 01 60 48
…	**…**
…	**…**
FM to Server	03 FC 00 03 15 A0 7F 44 0B 1D **0B** 00 6D 21 21 21 **…**
Server to FM	00 04 **6F** 01 00 6D C9 22
Last packet: FM to Server	02 37 00 03 15 A0 7F 44 0B 1D **0B** 00 6E 02 15 53 …
Last packet: Server to FM	00 04 **6F** 01 00 6E FB B9

*This data transfer flow happens after successful tachograph initialization (commands 10/110).

3.2.13 Command 14/114 – Transparent Channel data

Command is used by FM device to send transparent channel data to server (command 14). Packet is defined in the table below. Timestamp (UNIX timestamp) field indicates when the data was received by FM device from RS232 interface. Device does not start sending other tunnel channel data until ACK (acknowledgement) is received from server (command 114). Server can send data anytime after the link is established with the FM device. If server send a packet with non empty payload FM device will transmit payload content to RS232 device. Tco4 device supports dual tunnel channel mode: two channels can work independently.

FM device sends transparent packet with command 14 (0x0E).

Field	Packet length	IMEI	Command ID	Port ID	Reserved	Payload		CRC16
						Timestamp	Data	
Size (bytes)	2	8	1	1	2	4	[1-1004]	2
Example hex	0018	0003124D0 AC0BB1C	0E	01	0000	5A6098A5	61626331 32330D0A	BD80
Example dec	24	8645470323 16700	14	1	0	1516279973	-	-
Example ASCII	-	-	-	-	-	-	abc123	-

Extra fields are explained below:

Field		Description
Port ID		Values: 0 – port A, 1 – port B, 2 – port C (4th generation).
Reserved		Reserved bytes for future. Should leave zeros.
Payload	Timestamp	Timestamp of the payload data.
	Data	Raw data from port.

Server sends Acknowledgement to FM device with payload data (command 114 = 0x72).

Field	Packet length	Command ID	Port ID	Reserved	Payload	CRC16
Size (bytes)	2	1	1	2	[0-1016]	2
Example hex	000C	72	01	0000	6162633132330D0A	A897
Example dec	12	114	1	0	-	-
Example ASCII	-	-	-	-	abc123	-

As we can see from the examples (tables above) FM device sends data 'abc123' which was received from RS232 device (port B) with (command 14). Then server acknowledges this transmission (command 114) and additionally sends data 'abc123'. This packet is received by FM device and data 'abc123' is transmitted to RS232 device (port B). 4th generation devices can operate both ports simultaneously.

3.2.13.1 IO packet data via RS232

When data is sent from the FM device to another device connected via the RS232 interface and the “Send IO data through RS232” option is selected, the data is sent using the following structure: Record length (1B), record data (according to chapters 1 and 2) and CRC8 (1B). An example is provided below, record length is marked in blue and CRC8 is marked in red (**Note:** this example contains two records):

```
7B 5B 8F B2 44 00 10 00 0F 08 DF 75 20 A0 C5 2E 09 1F 7E 04 0D 00 00 0A 00 07 0A 00 05 00 00 1B 1B 00 02 00 00 03 00 00 1C 01 00
20 1C 00 AD 00 00 73 00 00 CF 00 00 82 00 07 00 1D 3B B5 00 1E 0F EC 00 16 00 0E 00 17 00 0C 00 74 00 00 00 C5 00 00 00 D2 00 00
06 00 41 00 00 04 53 00 96 00 00 60 1A 00 5C 00 00 18 06 00 72 00 00 00 D0 00 CB 00 00 00 00 00 D0 00 00 00 0C 00 75 53 5B 8F B2
44 00 11 00 0F 08 DF 75 20 A0 C5 2E 09 1F 7E 04 0D 00 00 0A 00 07 04 00 86 00 00 87 00 00 88 00 00 24 00 09 00 83 00 00 00 84 00
00 00 85 00 00 00 89 00 00 00 8B 00 00 02 0F 00 00 02 10 00 00 01 93 00 00 01 92 00 00 01 00 8A 00 00 00 00 00 2A
```

3.2.14 Command 15/115 – Identification packet

The command is used by the device to send identification packet (information) to the server. The identification packet payload is defined in the table below. The device does not start sending other data until ACK (acknowledgement) is received from the server (command 115) and breaks the open link. Firmware (cmd 104) and configuration (cmd 102) commands still work even if no ACK is received. All other commands from the server are discarded if the identification packet is not acknowledged.

The tracking device sends packet with command 15 (0x0F).

Field	Packet length	IMEI	Command ID	Payload	CRC16
Size (bytes)	2	8	1	37	2
Example hex	002E	0003124D0AC0BB1C	0F	5463303430302E30332E30392E3130000 0DFC1388631180000601A000003E80000 000A003C	DA26
Example dec	46	864547032316700	15	-	-

A server sends Acknowledgement to the tracking device with command 115(0x73).

Field	Packet length	Command ID	ACK	CRC16
Size (bytes)	2	1	1	2
Example hex	0002	73	01	CB25
Example dec	2	115	1	-

If the tracking device is unauthorized, the server sends a NACK response with command 115 (0x73). The tracking device then breaks the link with the server and does not make any further connection attempts until the delay time has passed.

Field	Packet length	Command ID	ACK	Delay (in minutes)	CRC16
Size (bytes)	2	1	1	1	2
Example hex	0003	73	02	B4	AFA3
Example dec	3	115	2	180	-

Note:
The response for the unauthorized tracking device is not applicable for the FM-Eco4 and FM-Plug4 devices.

The delay timer is reset to 0 if:

- The server responds with a positive response to the identification packet;
- The device is restarted;
- The connection to the server was initiated with *econnect*, *connect* or *switchip* SMS commands or GPRS command 105 *Set connection parameters*;
- The server parameters IP1, Port1, IP2 or Port2 were changed by SMS commands *setcfg* or *setconnection* or by uploading the configuration file via USB.

Identification packet payload:

Field	Device type	Firmware version	IMSI code	GSM operator code	Distance coefficient	Time coefficient	Angle coefficient
Size (bytes)	4	11	8	4	4	4	2
Example hex	54633034	30302E30332E30392E3130	0000DFC138863118	0000601A	000003E8	0000000A	003C
Example dec	-	-	246020970000664	24602	1000	10	60
Example ASCII	**Tc04**	**00.03.09.10**	-	-	-	-	-

As you can see from the example, some parameters (**bold text**) must be interpreted as ASCII characters. All other parameters are hex numbers. Parameter values are taken from the currently active profile.

Device types in ASCII:

- HCV5 – Tc05
- LCV5 – Lc05
- Pro5 – Pr05
- HCV5 Lite/Pro5 Lite - Tl05
- Trace5-LTM (Rev.A) / Trace5-2G / Trace5-LTE – We05
- Trace5-LTM – Wp05
- Trace5NA – Us05
- Trace5GL – Ws05
- FM-Tco4 HCV – Tc04
- FM-Tco4 LCV – Lc04
- FM-Pro4 – Pr04
- FM-Eco4 – Ec04
- FM-Eco4 S/T – Es04
- FM-Eco4 RS T - Rs04
- FM-Plug4 – OBD1

3.2.15 Command 16/116 – HeartBeat

If configured, FM device sends command 16 (0x10) to the server once a timeout has passed after the last record was sent, to inform the server that it is functioning. If a record is sent before the timeout has passed, the timeout resets. Command 16 (0x10):

Field	Packet length	IMEI	Command ID	CRC16
Size (bytes)	2	8	1	2
Example hex	0009	000310F561749007	10	BD93
Example dec	9	863071018192903	16	-

Server sends acknowledgement to the FM device with command 116 (0x74):

Field	Packet length	Command ID	ACK	CRC16
Size (bytes)	2	1	1	2
Example hex	0002	74	01	862D
Example dec	2	116	1	-

3.2.16 Command 17/117 – Set IO value

Command 117 (0x75) is used by a server to set tracking device IO to a specific value. Usually it is used to reset specific IO value to zero.

Field	Packet length	Command ID	Payload		CRC16
			IO ID	IO value	
Size (bytes)	2	1	4	4	2
Example hex	0009	75	000000AF	00000064	39CC
Example dec	9	117	175	100	-

After setting specific IO to a new value tracking device sends acknowledgement that ID is 17 (0x11).

Field	Packet length	IMEI	Command ID	Payload	CRC16
				ACK	
Size (bytes)	2	8	1	1	2
Example hex	000A	000310F56139B984	11	00	9799
Example dec	10	863071014336900	17	0	-

Possible ACK field values and their meanings:

- 0 – IO value was changed;
- 1 – FM device failed to change IO value;
- 2 – the value change for specified IO is not supported.

Parameters that can be set using command 17/117 (IDs provided in decimal):

Parameter	Trace5	Advanced family	Eco family	Plug4
Virtual odometer (ID:65)	●	●	●	●
ECO Absolute idling time (ID:175)	●	●**	●	●
CANBUS Distance (ID:114)*		●		
DIN1 hours accumulated (ID:577)	●	●	●	
DIN2 hours accumulated (ID:578)	●	●	●	
DIN3 hours accumulated (ID:579)	●	●	●	
DIN4 hours accumulated (ID:580)	●	●**	●	

** – Value will be overwritten by valid message from CAN-Bus (if available in specific vehicle).*

*** - Not available for HCV5 Lite/Pro5 Lite devices*

3.2.17 Command 18/115 – Dynamic identification packet

This command is used by the FM device to send a dynamic identification packet. This command is used instead of command 15/115 if *Dynamic* identification string is enabled in the configuration. The device does not start sending other data until ACK (acknowledgement) is received from server (command 115). Firmware (command 104) and configuration (command 102) commands still work even if no ACK is received. All other commands from server are discarded if the identification packet is not acknowledged.

The FM device should use command 18 (0x12):

Field	Packet length	IMEI	Command ID	Payload (5-1011 bytes)								CRC16
				Version	Param. count	Param. 1 ID	Param. 1 length	Param. 1 value	Param. 2 ID	Param. 2 length	Param. 2 value	
Size (bytes)	2	8	1	1	1	1	1	[Param. 1 length]	1	1	[Param. 2 length]	2
Example hex	0012	0003124D 0AC0BB1C	12	01	02	01	01	00	06	02	0441	8B10
Example dec	18	864547032 316700	18	1	2	1	1	0	6	2	1089	-

Description of payload fields:

- *Version* – the version of the dynamic packet command, determines how the payload is parsed;
- *Param. count* – how many parameters are included into the payload;
- *Param. X ID* – parameter ID;
- *Param. X length* – parameter length, defines the length of *Param. X value*;
- *Param. X value* – parameter value.

Parameter values are taken from the currently active profile.

List of parameters that can be included into the payload:

ID	Parameter name	Length (bytes)	Data type
1	Device type	4	String
2	FW version	11	String
3	IMSI	8	Unsigned
4	GSM operator	4	Unsigned
5	Distance coefficient	4	Unsigned
6	Time coefficient	4	Unsigned
7	Angle coefficient	2	Unsigned
8	OCSP status	1	Unsigned
9	Bootloader version	2	Unsigned
10	HW version	2	Unsigned
11	CFG tag	0-32	String
15	Last CFG change (applicable for 5th generation devices)	4	Unsigned, UNIX timestamp
255	Error	0	Payload exceeds 1011 bytes

Note: For the *Last CFG change* parameter the timestamp bytes are switched (big-endian byte sequence).

The server responds with command 115 (0x73):

Field	Packet length	Command ID	ACK	CRC16
Size (bytes)	2	1	1	2
Example hex	0002	73	01	CB25
Example dec	2	115	1	-

If the tracking device is unauthorized, the server sends a NACK response with command 115 (0x73). The tracking device then breaks the link with the server and does not make any further connection attempts until the delay time has passed.

Field	Packet length	Command ID	ACK	Delay	CRC16
Size (bytes)	2	1	1	1	2
Example hex	0003	73	02	B4	AFA3
Example dec	3	115	2	180	-

Note:
The response for the unauthorized tracking device is not applicable for the FM-Eco4 and FM-Plug4 devices.

The delay timer is reset to 0 if:

- The server responds with a positive response to the identification packet;
- The device is restarted;
- The connection to the server was initiated with *econnect*, *connect* or *switchip* SMS commands or GPRS command 105 *Set connection parameters*;
- The server parameters IP1, Port1, IP2 or Port2 were changed by SMS commands *setcfg* or *setconnection* or by uploading the configuration file via USB.

3.2.18 Command 19/107 – 2nd generation Smart Card data size and timestamp

FM uses this command 19 (0x13) to send information about 2nd generation smart card .DDD files and the timestamp when the file was created.

Field	Packet length	IMEI	Command ID	Payload		CRC16
				Size	Time stamp	
Size (bytes)	2	8	1	4 + 4 = 8		2
Example hex	0011	000315A07F44865A	13	00005428	4E9CAF2C	5590
Example dec	17	868204004279898	19	21544	1318891308	-

The server responds with command 107 (0x6B).

Field	Packet length	Command	ACK	CRC16
Size (bytes)	2	1	1	2
Example hex	0002	6B	01	9074
Example dec	2	107	1	-

ACK possible values: 0 – negative acknowledgement, 1 – positive acknowledgement, 2 – card data rejected.

3.2.19 Command 30/130 – Garmin Device Request Status

Command is used by server to get the status about Garmin device and to know if Garmin device is correctly connected to FM device. Server sends a request with command 130 (0x82).

Field	Packet length	Command ID	CRC16
Size (bytes)	2	1	2
Example hex	0001	82	A71A
Example dec	1	130	-

FM device should answer with command 30 (0x1E).

Field	Packet length	Command ID	Payload (Status)	CRC16
Size (bytes)	2	1	1	2
Example hex	0002	1E	01	1E08
Example dec	2	30	1	-

Garmin status can have three different states:

- 0x01 - Garmin is not responding (not connected to FM device).
- 0x02 - Garmin is responding (connected to FM device).
- 0x03 - Garmin is responding and supports unicode protocol (connected to FM device and is compatible with Garmin protocol).

3.2.20 Command 31/131 – Garmin Device Data

Commands are used for communication between server and Garmin via FM device. If Garmin is correctly connected to FM device then FM will act as a clear channel between Garmin and server. Everything what is in command packet payload will be sent to Garmin. Garmin uses FMI standard for communication ((https://developer.garmin.com/fleet-management/protocol-support/)). So there should be FMI message in the payload in command 131 (0x83) packet (table below).

Field	Packet length	Command ID	Payload (Garmin device data – FMI message)	CRC16
Size (bytes)	2	1	Not fixed	2
Example hex	001D	83	10A11640000600000035303030313100000000000000000000DC1003	753C
Example dec	29	131	-	-

Answer from Garmin is also a FMI message in payload in command 31 (0x83) packet.

Field	Packet length	IMEI	Command ID	Payload (Garmin device data – FMI message)	CRC16
Size (bytes)	2	8	1	Not fixed	2
Example hex	003A	00000B1A2A3C833E	1F	100602A100571003578B002500000B1A2A3C833E1F10A11641000602000035303030313100000000000000000000D91003	C112
Example dec	58	12207005664062	31	-	-

FM device just forwards FMI messages between server and Garmin device. Server should format FMI packet correctly, otherwise where will be no answer from Garmin and FM device.

3.2.21 Command 32/132 – Weighting system data

Commands are used for communication between server and weighting system via FM device. If weighting system is correctly connected to FM device (via RS232 interface) then FM will send system's data strings to server using command 32 (0x20). Additionally FM device puts „record data“ information. An example is shown below.

Field	Packet length	IMEI	Command ID	Record data	Payload (Weighting system data)	CRC16
Size (bytes)	2	8	1	14	Not fixed (min: 9; max 100 bytes)	2
Example hex	0033	00000B1A2A3C833E	20	5268CEF200196E3A3A0AEF3E0101	30303030303030303330303a303030303030303730303a474c4153530D	9F91
Example dec	51	12207005664062	32	-	-	-
ASCII	-	-	-	-	0000000300:0000000700:GLASS<0xOD>	-

Record data field:

Record data		
Parameter	**Size (bytes)**	**Value**
Timestamp	4	Unix time-stamp
Longitude	4	Longitude value multiplied by 10000000
Latitude	4	Latitude value multiplied by 10000000
GPS fix	1	1 – if device has GPS fix, 0 – if not
PORT	1	0 – PORT A, 1 – PORT B

Server will send acknowledgement command 132 (0x84) to FM device.

Field	Packet length	Command ID	Payload	CRC16
Size (bytes)	2	1	Not fixed	2
Example hex	0002	84	01	FA25
Example dec	2	132	1	-

3.2.22 Command 33/133 – FLS communication channel between device and server

The commands are used for communication between server and FLS via FM device. If the FLS is connected correctly to the FM device, then the device will act as a clear channel between FLS and the server. Everything that is included in FLS package will be sent to the addressed FLS on Port X.

Communication from server to FM device, command 133 (0x85).

Field	Packet length	Command ID	SubCmdID	Port X	FLS package	CRC16
Size (bytes)	2	1	1	1	Length is not fixed	2
Example hex	000D	85	1, 2, 3	-	-	00A1
Example desc.	-	133	-	-	-	-

Communication from FM to server, command 33 (0x21 hex)

Description	Packet length	IMEI	Command ID	Answer to SubID	Port X	FLS package	CRC16
Size (Bytes)	2	8	1	1	1	Length is not fixed	2
Example hex	000D	00000B1A29F64B1A	21	-	-	-	-
Example value decimal	-	12207001062170	33	-	-	-	-

Field descriptions:

Field	Description
Packet length	Packet length field: all fields except itself and CRC16 (except Packet length field and CRC16 field). Calculated as with FM protocol
Command ID	Command ID is 8 bit variable. Command ID describes what type of command is received, and how to handle DATA field.
SubCmdID	1 – open RFLS CHANNEL*; 2 – send data through the RFLS channel; 3 – close RFLS channel
PORT X	It depends on the field SubCmID. For SubCmdID '1' available values: 0 – PORTA, 1 – PORTB, 2 – PORTC. Communication port with FLS.
FLS package	This data field is described in protocol description parts: Common protocol for fuel level sensor data exchange located in the “Ruptela FLS protocol section 2” & Internal (Ruptela) communication protocol structure located in this document at section 2 located at doc.ruptela.lt

**When the SubCmdID is 1 the „FLS package“ has different value. This bytes would contain the address of all the FLS that are connected to the device. In total there are 32 bytes. Starting from the MSB to the LSB. 1 indicate FLS connected and 0 not connected. This is bitfield from 255 bit to 0 bit that each bit correspond to the address where the FLS is connected (if is set). See below table „Address bits field“*

Address bits field:

Bit 255	Bit 254	Bit 253	...	Bit 2	Bit 1	Bit 0
0	1	0	...	0	1	0
Found 2 FLS with addresses 0x01, 0x40						
40 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 02						

Communication examples:

SubCmdID 1: Open RFLS Channel

(TX) Server requests to open a communication channel “*open RFLS CHANNEL*”:

Field	Packet length	Command ID	SubCmdID	Port X	FLS package	CRC16
Size (bytes)	2	1	1	1	Length is not fixed	2
Example hex	00 03	85	01	01	NE*	3D 00

*: NE – does not exist.

(RX) FM answered with a positive answer “opened RFLS CHANNEL”. FM founded one FLS with an address 0x01:

Field	Packet length	IMEI	Command ID	Answer to SubID	Port X	FLS package	CRC16
Size (bytes)	2	8	1	1	1	Length is not fixed	2
Example hex	00 2B	00 03 13 61 2D A2 CF C6	11	01	01	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 02	DD6A

SubCmdID 2: Send data through the RFLS channel:

(TX) Server sends data through the communication channel. Server tries to get an information about device parameters:

Field	Packet length	Command ID	SubCmdID	Port X	FLS package	CRC16
Size (bytes)	2	1	1	1	Length is not fixed	2
Example hex	00 03	85	02	01	PAYLOAD	1768

(RX)FM sends answer from FLS through RFLS channel:

Field	Packet length	IMEI	Command ID	Answer to SubID	Port X	FLS package	CRC16
Size (bytes)	2	8	1	1	1	Length is not fixed	2
Example hex	00 24	00 03 13 61 2D A2 CF C6	21	02	01	PAYLOAD	E00A

SubCmdID 3: Close RFLS channel

(TX) Server requests to close a communication channel “close RFLS CHANNEL”:

Field	Packet length	Command ID	SubCmdID	Port X	FLS package	CRC16
Size (bytes)	2	1	1	1	-	2
Example hex	00 03	85	03	01	-	0EB0

(RX)FM sends answer from FLS through RFLS channel:

Field	Packet length	Command ID	SubCmdID	Port X	FLS package	CRC16
Size (bytes)	2	1	1	1	Length is not fixed	2
Example hex	00 24	31	03 or -03	01	-	CRC16

3.2.23 Command 34/134 – SD card logging functionality

Command is used by server to request log entries from device and to control logging functionality.

ATTENTION: internal structure of this command is very similar to Command 1/100 (Records), but this command has SubCommand in payload section. SubCommand is used to separate log requests from control commands and standard protocol log entries from extended.

3.2.24.1 Subcommand 0 (0x00) – Requesting data from log

Subcommand is used by server to get the log records stored on SD card. Server sends a request with command 134 (0x86) and subcommand 0 (0x00).

Field	Packet length	Command ID	Subcommand ID	Start timestamp	End timestamp	Reserved	CRC16
Size (bytes)	2	1	1	4	4	2	2
Example hex	000C	86	00	553F44F2	553F45A6	0000	B0FA
Example dec	12	134	0	1430209778	1430209958	0000	-

Start timestamp – UTC time to read log records FROM (UNIX time stamp).
End timestamp – UTC time to read log records TO (UNIX time stamp).

FM device sends log records according to the format which is shown below.

Field	Packet length	IMEI	Command ID	Payload				CRC16
				Subcommand ID	Records left	Number of records	Data (see *Records*)	
Size (bytes)	2	8	1	1	1	1	[1-1009]	2
Example hex	0113	00000B1A29F64B1A	22	40	00	0C	12 records data...	**CRC16**
Example dec	275	12207001062170	34	64	0	12	12 records data...	-

Subcommand ID – two possible values: 64 (0x40) – SD card log records in **standard** protocol; 128 (0x80) – SD card log records in **extended** protocol.

Records left flag – two possible values: 0 or 1. 0 – there are no log records left in selected time interval, 1 – there are log records left in selected time interval.

Number of records – describes how many records are in Record DATA field.

Data – records with variable length. Record consists of non variable length field: time stamp, priority, GPS elements and variable length field: IO elements.

When session is closed during log transmission to server (timeout), upon next session, regular records are sent. SD card log must be requested from server again, with a respect to what records we already have in server.

In case if device don't have any records in requested time interval it will sends packet with empty payload (*Records left flag* and *Number of records* fields will have values 0).

NOTICE: event ID field of each record in Data payload block – for all log records is 252.

Answer from server:

Field	Packet length	Command	SubCommand ID	CRC16
Size (bytes)	2	1	1	2
Example hex	0002	86	01	C995
Example dec	2	134	1	-

Server should use command 134 (0x86) with subcommand 1 (0x01) for log records response (acknowledge).

After that FM device will send next log records pack from requested time interval if unsent log records available or send packet with empty payload (*Records left flag* and *Number of records* fields will have values 0) to show that there is no log records to send.

3.2.24.2 Subcommand 2 (0x02) – Stop sending procedure

Subcommand is used by server to stop log records sending and clear requested time interval. Server sends a request with command 134 (0x86) and subcommand 2 (0x02).

Field	Packet length	Command ID	Subcommand ID	CRC16
Size (bytes)	2	1	1	2
Example hex	0002	86	02	FB0E
Example dec	2	134	2	-

There is no answer to this command from device.

3.2.24.3 Subcommand 16 (0x10) – Erase SD card data

Subcommand is used by server to erase whole SD card (used to initialize SD card or to switch between Event records and Log records mode, only same type of data can be stored on SD card at same time and for ex. if user need to enable SD card logging functionality and at least one event record present on the SD card – to prevent missing Event records FM device generates temporary error state so user will able to download all event records to server and then execute this command to enable SD card logging). Server sends a request with command 134 (0x86) and subcommand 16 (0x10).

Field	Packet length	Command ID	Subcommand ID	CRC16
Size (bytes)	2	1	1	2
Example hex	0002	86	10	C89D
Example dec	2	134	16	-

FM device should answer with command 34 (0x22) and subcommand 16 (0x10).

Field	Packet length	IMEI	Command ID	Payload		CRC16
				Subcommand ID	Status	
Size (bytes)	2	8	1	1	1	2
Example hex	000B	00000B1A29F64B1A	22	10	01	85DF
Example dec	11	12207001062170	34	16	1	-

Status field can have two different states:

- 0x00 – SD card erase error.
- 0x01 – SD card erase completed.

3.2.24.4 Subcommand 32 (0x20) – Enable/Disable SD card logging functionality

Subcommand is used by server to remotely enable or disable SD card logging functionality (for example if accident happened and to prevent log data overwriting).

NOTICE: This command only enable/disable logging temporary (until device restarts or profile reload happens).

Server sends a request with command 134 (0x86) and subcommand 32 (0x20).

Field	Packet length	Command ID	Payload			CRC16
			Subcommand ID	Parameter set	Logging interval	
Size (bytes)	2	1	1	1	4	2
Example hex	0007	86	20	01	553F44F2	D247
Example dec	7	134	32	1	1430209778	-

FM device should answer with command 34 (0x22) with Subcommand 32 (0x20).

Field	Packet length	IMEI	Command ID	Payload		CRC16
				Subcommand ID	Status	
Size (bytes)	2	8	1	1	1	2
Example hex	000B	00000B1A29F64B1A	22	20	01	337D
Example dec	11	12207001062170	34	32	1	-

Status field can have three different states:

- 0x00 – SD card logging stopped.
- 0x01 – SD card logging started.
- 0x02 – SD card logging can't start (problems with SD card or it contains regular records)

3.2.24 Command 35/135 – Accident reconstruction

This command is used by the tracking device to send the accident reconstruction data to the server.

The accident reconstruction command uses 2 messages: announcment and data message.

The announcement message (0x02 – data version) - structure:

Field	Packet length	IMEI	Command ID	Data version	Payload data	CRC16
Size (bytes)	2	8	1	1	29	2
Example hex	29	0003107BD952952E	23	02	Data	-
Example dec	41	862549043221806	35	2	Data	-

The announcement message contains configured accelerometer, gyroscope, and GNSS recording frequencies and recording duration before and after the IO event is generated. Also, the accelerometer calibration matrix is recorded at the end.

The annoucment data example:

Field	Acc frequency	ACC duration before	Acc duration after	Gyro frequency	Gyro duration before	Gyro duration after	GNSS frequency	GNSS duration before	GNSS duration after	Acc XX	Acc XY	Acc XZ	Acc YX	Acc YY	Acc YZ	Acc ZX	Acc ZY	Acc ZZ
Size (bytes)	2	1	1	2	1	1	1	1	1	2	2	2	2	2	2	2	2	2
Example hex	0064	0F	05	0000	00	00	0A	0F	0A	006B	FC1E	FFFC	FC1F	FF95	FFE2	001D	0008	FC19
Example dec	100	15	5	0	0	0	10	15	10	107	-994	-4	-993	-107	-30	29	8	-999

The data message (0x03 – data version) structure:

Field	Packet length	IMEI	Command ID	Data version	Payload data				CRC16
					Records left	Type of data	Numbers of records	Data	
Size (bytes)	2	8	1	1	1	1	1	1-1007	2
Example hex	0400	0003107BD952952E	23	03	13	0	14	Data	-
Example dec	1024	862549043221806	35	3	19	0	20	Data	-

Type of data specifies from which source the data record is generated:

0x00 – accelerometer
0x01 – gyroscope
0x02 – GNSS

Example of the accelerometer data:

Field	ID	Acc X	Acc Y	Acc Z	ID	Acc X	Acc Y	Acc Z
Size (bytes)	2	2	2	2	2	2	2	2
Example hex	0000	000C	FFFB	FC04	0001	000A	FFFD	FC19
Example dec	0	12	-5	-1020	1	10	-3	-999

Example of the GNSS data:

Field	ID	Latitude	Longtitude	Timestamp	Speed	ID	Latitude	Longtitude	Timestamp	Speed
Size (bytes)	2	4	4	4	2	2	4	4	4	2
Example hex	0000	20A0D3E4	0F08B6E6	5416F6E1	0032	0001	20A0D5F5	0F08B6E6	5416F6E2	0033
Example dec	0	547410916	252229350	1410791137	50	1	547411445	252229350	141079113	51

Answer to the command from a server:

Field	Packet length	Command ID	ACK	CRC16
Size (bytes)	2	1	1	2
Example hex	0002	87	01	D04D
Example dec	2	135	1	-

3.2.25 Command 37/137 – Files

This command is used by the server to manage pictures taken by peripherals with cameras.

The tracking device should use Command 37 (0x25):

Field	Packet length	IMEI	Command ID	Subcommand ID	Source ID	Subcommand payload	CRC16
Size (bytes)	2	8	1	1	1	Not fixed	2
Example hex	0300	000314F82AF6BBB5	25	00	00	-	-
Example dec	768	867481035389877	37	0	0	-	-

The server should use Command 137 (0x89):

Field	Packet length	Command ID	Subcommand ID	Source ID	Subcommand payload	CRC16
Size (bytes)	2	1	1	1	Not fixed	2
Example hex	0400	89	00	00	-	-
Example dec	1024	137	0	0	-	-

Subcommand ID can have the following values:

- 0 – Quantity;
- 1 – Name;
- 2 – Transfer;
- 3 – Delete by name;
- 4 – Delete by timestamp range.

Source ID can have the following values:

- 0 – Camera A folder on SD card;
- 1 – Camera B folder on SD card;
- 2 – PortA camera memory;
- 3 – PortB camera memory;
- 4 – Fatigue sensor folder on SD card;
- 5 – Fatigue sensor memory. (This value is used only by the FM-Eco4 RS T device, other values for this device are not applicable)

3.2.25.1 Subcommand 0 (0x00) – Quantity

This subcommand is used by the server to get the quantity of stored pictures in tracking device SD card or camera

memory for a time interval.
The server sends a request:

Field	Packet length	Command ID	Subcommand ID	Source ID	Timestamp start	Timestamp end	CRC16
Size (bytes)	2	1	1	1	4	4	2
Example hex	000B	89	00	00	5B346380	5B3481E8	54FE
Example dec	11	137	0	0	1530160000	1530167784	-

The tracking device responds:

Field	Packet length	IMEI	Command ID	Subcommand ID	Source ID	Quantity	CRC16
Size (bytes)	2	8	1	1	1	2	2
Example hex	000D	000314F82AF6BBB5	25	00	00	000A	1821
Example dec	13	867481035389877	37	0	0	10	-

3.2.25.2 Subcommand 1 (0x01) – Name

This subcommand is used to retrieve file names of stored pictures in tracking device SD card or camera memory for a time interval.

The server sends a request:

Field	Packet length	Command ID	Subcommand ID	Source ID	Timestamp start	Timestamp end	CRC16
Size (bytes)	2	1	1	1	4	4	2
Example hex	000B	89	01	00	5B346380	5B3481E8	78D9
Example dec	11	137	1	0	1530160000	1530167784	-

The tracking device responds:

Field	Packet length	IMEI	Command ID	Subcommand ID	Source ID	Quantity	File names (ASCII)	CRC16
Size (bytes)	2	8	1	1	1	1	Quantity x 8	2
Example hex	0300	000314F82AF6BBB5	25	01	00	02	35423334363338303542333438314538	-
Example dec	768	867481035389877	37	1	0	2	-	-

Note: If the file name is shorter than 8 symbols, empty spaces will be added at the end, e.g. E280AD46696C6531322020 – "File12 "

3.2.25.3 Subcommand 2 (0x02) – Transfer

This subcommand is used to transfer pictures to the server.

The server sends a request:

Field	Packet length	Command ID	Subcommand ID	Source ID	File name(ASCII)	Packet number	CRC16
Size (bytes)	2	1	1	1	8	2	2
Example hex	000D	89	02	00	3542333436333830	0000	94CE
Example dec	13	137	2	0	-	0	-

Note: For FM-Eco4 RS T devices, the server sends the following value to the device instead of *File name*: 2B *Alert ID* + 6B 0x000000000000 (reserved).

The tracking device responds:

Field	Packet length	IMEI	Command ID	Subcommand ID	Source ID	File name (ASCII)	Total packets	Current packet	Data	CRC16
Size (bytes)	2	8	1	1	1	8	2	2	Not fixed	2
Example hex	0300	000314F82AF6BBB5	25	02	00	3542333436333830	0051	0000	-	-
Example dec	768	867481035389877	37	2	0	-	81	0	-	-

Note: If *Source ID* is 2 or 3, it means that the pictures are sent directly from camera memory. The server should start from packet 0 and request subsequent packets. After the last packet is received, the server requests the picture to be deleted from camera memory.

In case of a wrong request:

- Source ID – the device returns the same source ID as requested with 0xFF values for file name, 0x00 for total packets, 0x00 for current packet and empty data;
- File name – the device returns the same file name, 0x00 for total packets, 0x00 for current packet and empty data;
- Packet number – the device gives an answer with an empty data field.

Note: For FM-Eco4 RS T devices, the server sends the following value to the device instead of *File name*: 2B *Alert ID* + 6B 0x000000000000 (reserved).

3.2.25.4 Subcommand 3 (0x03) – Delete by name

This subcommand is used to delete pictures by filename.

The server sends a request:

Field	Packet length	Command ID	Subcommand ID	Source ID	File name(ASCII)	CRC16
Size (bytes)	2	1	1	1	8	2
Example hex	000B	89	03	00	3542333436333830	9032
Example dec	11	137	3	0	-	-

The tracking device responds:

Field	Packet length	IMEI	Command ID	Subcommand ID	Source ID	File name (ASCII)	Status	CRC16
Size (bytes)	2	8	1	1	1	8	1	2
Example hex	0014	000314F82AF6BBB5	25	03	00	3542333436333830	01	9B9C
Example dec	20	867481035389877	37	3	0	-	1	-

Status can have the following values:

- 0 – Error;
- 1 – OK.

3.2.25.5 Subcommand 4 (0x04) – Delete by timestamp range

This subcommand is used to delete pictures by timestamp range.

The server sends a request:

Field	Packet length	Command ID	Subcommand ID	Source ID	Timestamp start	Timestamp end	CRC16
Size (bytes)	2	1	1	1	4	4	2
Example hex	000B	89	04	00	5B346380	5B3481E8	E462
Example dec	11	137	4	0	1530160000	1530167784	-

The tracking device responds:

Field	Packet length	IMEI	Command ID	Subcommand ID	Source ID	Status	CRC16
Size (bytes)	2	8	1	1	1	1	2
Example hex	000C	000314F82AF6BBB5	25	04	00	01	927A
Example dec	12	867481035389877	37	4	0	1	-

Status can have the following values:

- 0 – Error;
- 1 – OK.

3.2.26 Command 38/138 – Beacon Data Sending via GPRS

This command is used to send the beacon records data to a server via GPRS.

Field	Packet length	IMEI	Command ID	Number of records (how many records to expect)	Records left flag (1 - yes, 0 - no)	Beacon records (up to 4 records in one message)	CRC16
Example hex	03FC	000310f561749007	26	25	01	-	-
Example dec	1020	863071018192903	38	37	01	-	-

Where:

Number of records - indicates how many records have to be sent, including those that were received with this message.

Records left flag - indicates if there are any unsent messages.

Beacon record data sample (up to 6 beacons in one record)

Field	Timestamp	Coordinates		Message type	Number of beacons	Beacon #1 RSSI	Beacon #1 type	Beacon #1 data	Beacon #1 flag
		Longitude	Latitude						
Size (bytes)	4	4	4	1	1	1	1	37	1
Example hex	5FE31AA8	07D608F1	1A1480BA	01	01	B5	03	f5647c0365dd0201060c09502049442030303533303320000000000000000000000000000000	54
Example dec	1608719016	131467505	437551290	01	01	181	3	-	0

Note: To get the RSSI in decibels, subtract 256 from the decimal RSSI value, e.g. 181 – 256 = -75 dBm.

Beacon types:

0x01 – iBeacon

0x02 – Eddystone

0x03 – MAC only

Beacon flag:

0x00 – no flag is set in the Beacon list or the beacon is not used for any additional features that use flags.

0x44 – the *C* flag is set in the Beacon list, it gives additional indicator that this beacon is marked as cargo.

0x54 – the *T* flag is set in the Beacon list and it is used for the Wireless trailer ID.

Response structure:

Field	Packet length	Command ID	ACK	CRC16
Size (bytes)	2	1	1	2
Example hex	0002	8A	1	-
Example dec	2	138	1	-

3.2.27 Command 105 – Set Connection Parameters

Command is used to set temporary connection settings (connection IP address, port and protocol type). Server uses command 105 (0x69).

Field	Packet length	Command ID	Payload	CRC16
Size (bytes)	2	1	Not fixed	2
Example hex	0015	69	3139322E3136382E302E312C393031352C544350	1763
Example dec	21	105	-	-
Example ASCII	-	-	192.168.0.1,9015,TCP	-

Tracking device does not send a response. Next time (for one time) tracking device will connect to server using received connection parameters.

3.2.28 Command 106 – FM device Odometer Set

Command 106 (0x6A) is used by server to set specific value of tracking device's virtual odometer. Usually it is used to reset odometer value to zero (0x00000000).

Field	Packet length	Command ID	Payload	CRC16
Size (bytes)	2	1	4	2
Example hex	0005	6A	12345678	8AEB
Example dec	5	106	305419896	-

Tracking device does not send a response. Next generated record will have new odometer value.

3.2.29 Supported communication command table

	5th generation devices					4th generation devices						
Command	**HCV5**	**LCV5**	**Pro5**	**HCV5 Lite Pro5 Lite**	**Trace5**	**Eco4**	**Eco4 S**	**Eco4 T**	**Pro4**	**Tco4 LCV**	**Tco4 HCV**	**Plug4**
1/100 – Records						●	●	●	●	●	●	●
68/100 – Extended protocol records	●	●	●	●	●	●	●	●	●	●	●	●
2/102 – Device Configuration Data	●	●	●	●	●	●	●	●	●	●	●	●
3/103 – Device Version Info	●	●	●	●	●	●	●	●	●	●	●	●
4/104 – Device Firmware Update	●	●	●	●	●	●	●	●	●	●	●	●
5/107 – Smart Card Data	●			●							●	
6/107 – Smart Card Data Size and Time stamp	●			●							●	
7/108 – SMS via GPRS	●	●	●	●	●	●	●	●	●	●	●	●
9/109 – Diagnostic Trouble Codes	●	●								●	●	●
10/110 – Tachograph Communication	●			●							●	
11/111 – Tachograph Data Packet	●			●							●	
12/111 – Information Packet About Tachograph Data	●			●							●	
14/114 – Transparent Channel data	●	●	●						●	●	●	
15/115 – Identification packet						●	●	●	●	●	●	●
16/116 - HeartBeat	●	●	●	●	●	●	●	●	●	●	●	●
17/117 – Set IO value	●	●	●	●	●	●	●	●	●	●	●	●
18/115 – Dynamic identification packet	●	●	●	●	●		●	●	●	●	●	
19/107 – 2nd generation smart card data size and timestamp	●			●							●	
30/130 – Garmin Device Request Status	●	●	●						●	●	●	
31/131 – Garmin Device Data	●	●	●						●	●	●	
32/132 – Weighting system data	●										●	
33/133 – FLS communication channel between device and server	●	●	●						●	●	●	
34/134 – SD card logging	●	●	●						●	●	●	

	5th generation devices					4th generation devices						
Command	**HCV5**	**LCV5**	**Pro5**	**HCV5 Lite Pro5 Lite**	**Trace5**	**Eco4**	**Eco4 S**	**Eco4 T**	**Pro4**	**Tco4 LCV**	**Tco4 HCV**	**Plug4**
functionality												
35/135 – Accident reconstruction	●	●	●									
37/137 – Files	●	●	●						●	●	●	
38/138 – Beacon list	●	●	●									
105 – Set Connection Parameters	●	●	●	●	●	●	●	●	●	●	●	●
106 – FM device Odometer Set	●	●	●	●	●	●	●	●	●	●	●	●

4 SMS

There is a possibility to send SMS commands to the FM device. The FM device answers with an SMS message as well. This is an easy way to quickly interact with the FM device. SMS commands are used:

- to get specific information from the FM device;
- to reconfigure some parameters of the FM device;
- to influence how the FM device functions.

SMS command structure: *<password> <command> <command text>*

The password, command and command text are separated by a space symbol. Command text parameters are usually separated by a comma (,) symbol. The password can contain up to 16 symbols. The following symbols are permitted:

- All digits (0-9);
- Letters A-Z (both lowercase and uppercase);
- "_", "@", "#", "/", "%", "-".

Note
The password cannot match any names of SMS commands. Please refer to the Supported SMS commands table for a full list of SMS commands.

Note
SMS commands that are sent over GPRS do not require a password.

It is possible to include IDs in SMS commands, so that the server knows to which SMS command the device responds. To do so, the **Use ID in SMS commands** functionality must be enabled in the configuration. If the functionality is enabled, the ID must be added to all SMS commands, otherwise the commands will not be processed.

SMS command structure with ID: *<password> <command> <command text> ID<ID string>*

<ID string> - a 16-32 hexadecimal symbol tag. It must be entered right after ID (no spaces).

Example: *password reset ID575D6468E934C867*

Response: *Resetting device ID575D6468E934C867*

Note
The ID must be unique. If the device receives several SMS commands with the same ID, only the first one will be processed.

Note
SMS commands *gsminfo, getapn* and *accreset -r* will NOT return an ID even if it is included in these SMS commands.

4.1 Commands

4.1.1 accinfo

This SMS command is used to check if the accelerometer is calibrated for the Eco-Drive functionality. Only state information should be interpreted.

Parameter	Description
State	2 – accelerometer calibration has been initiated 3 – the gravitational vector has been determined 11 – accelerometer calibrated
XYZo, Xc, Yc, Zc	Parameters used for accelerometer event calculations.

Example: *pass accinfo*

Response example: *AXL state:11 XYZo:120 -39 -945 Xc:131 -92 -94 -7 Yc:-138 97 -96 16 Zc:953 120 -39 -945*

4.1.2 accreset

This SMS command is used to reset the existing accelerometer (used for the Eco-Drive functionality) calibration and initiate a new automatic calibration process. For automatic calibration to be successful, the data course must be set to *ACC*, *CAN+OBD* or *OBD+ACC*.

Example: *pass accreset*

Response example: *Acc reset OK*

An additional parameter can be included in the main SMS command which will reset the existing calibration, initiate a new automatic calibration process and inform the user of the ongoing calibration status:

Example: *pass accreset r*

Response example if the conditions are correct to start calibration: *Acc calibration initiated. Drive normally to calibrate.;*

Response example if the conditions are incorrect: *Configuration error;*

Afterwards the device will send SMS messages based on the calibration status:

Up direction found. Drive normally to finish calibration. – the gravitational vector has been determined and acceleration events will be gathered.

Calibration successful – the calibration process was finished successfully.

4.1.3 ahj-on/ahj-off

The SMS command *ahj-on* is used to activate the Anti-hijack sequence and *ahj-off* is used to deactivate it in accordance with the Anti-hijack workflow. For a more detailed description of the workflow and mode functionality, refer to the Digital Output User Manual.

ahj-on SMS command structure: *ahj-on <A>, <B>, <C>*

Where:

- *<A>* - Activation speed, km/h (Range: *0–250)*
- *<B>* - Deactivation timeout, s (Range: 1–65000)
- *<C>* - Reactivation timeout, s (Range: 0–65000)

ahj-on SMS command structure: *ahj-on **

When "*" is provided instead of the mentioned parameters, the corresponding values will be set from the configuration.

ahj-off SMS command structure: *ahj-off <A>, <B>*

Where:

- *<A>* - Guaranteed unrestricted time period, s (Range: *0* - 65000)
- *<B>* - Unconditionally initiate after guaranteed unrestricted period expires (Values: 0 – no, 1 – yes)

Responses:

If the *ahj-on* command is accepted: *ahj-on OK*

If the *ahj-off* command is accepted: *ahj-off OK*

If the *ahj-on* command is declined (the mode is not configured, invalid values, invalid structure):

ahj-on ERROR

If the *ahj-off* command is declined (the mode is not configured, invalid values, invalid structure):

ahj-off ERROR

4.1.4 authid

This SMS command is used to add/remove IDs to/from the expanded list. Up to 4 IDs can be modified at a time.

Command syntax to add IDs: *password authid add <ID1>, <ID2>, <ID3>, <ID4>*

<IDX> – the ID in hexadecimal format

Response example if 2 IDs were added successfully: *add <ID1> OK <ID2> OK*

Note

The ID length must be exactly 8 B or 16 B, empty bytes will not be filled with zeros.

Command syntax to delete IDs: *password authid del <ID1>, <ID2>, <ID3>, <ID4>*

<IDX> – the ID in hexadecimal format

Response example if 2 IDs were deleted successfully: *del <ID1> OK <ID2> OK*

If the IDs were not modified successfully, you will receive one of the following error responses:

- *Not enough memory* – the device does not have enough memory for all IDs

- *Incorrect length* – the entered IDs are not 8 B/16 B long
- *Too many IDs* – more than 4 IDs were entered
- *Not found* – the ID cannot be deleted because it was not found in the ID list
- *Incorrect syntax* – the command was entered incorrectly
- *Not configured* – the expanded authorized IDs list is disabled
- *Device busy* – the device is currently busy with another process
- *Already exists* – the ID cannot be added because it was already found in the ID list
- *Repeating IDs* – the same ID was entered in the *authid* command several times

4.1.5 banned

This SMS command is used to get information about temporary banned operators. Answer SMS format: *Already banned:X, Newly banned:Y, ops:ZZZ.*

Parameter	Description
X	Number of times when FM tried to ban operator which has already been in the banned list.
Y	Number of times when FM added operator in the banned list.
ZZZ	List of currently banned operators (which are still in the list).

Example: *pass banned*

Response example: *Already:1, Newly banned:1, ops:24602,*

4.1.6 caninfo

This SMS command is used to get information about the settings of the CAN1 interface. These settings are used to see exact CAN interface setup in configuration file (it doesn't necessarily reflect actual mode of operation). Answer SMS format: *CAN enable: X Manufacturer Y Type Z Active A*

Parameter	Description
CAN enable	0 – CAN is disabled / 1 – CAN is enabled, FMS standard mode / 2 – CAN is enabled, LCV mode / 3 – CAN is enabled, OBD mode / 4 – CAN is enabled, Tachograph mode.

<table>
<tr><td>Manufacturer*</td><td>Manufacture group of Light Commercial Vehicles (number value).
<table>
<tr><th>Value</th><th>Name</th></tr>
<tr><td>1</td><td>VAG</td></tr>
<tr><td>2</td><td>Mercedes</td></tr>
<tr><td>3</td><td>Citroen</td></tr>
<tr><td>4</td><td>Ford</td></tr>
<tr><td>5</td><td>Fiat</td></tr>
<tr><td>6</td><td>Opel</td></tr>
<tr><td>7</td><td>Renault</td></tr>
<tr><td>8</td><td>Toyota</td></tr>
<tr><td>9</td><td>FMS Tractor</td></tr>
</table>
</td></tr>
<tr><td>Type</td><td>Type of Light Commercial Vehicle (number value).</td></tr>
<tr><td>Active</td><td>0 – CAN mode is silent mode, 1 – CAN mode is active mode</td></tr>
</table>

Example: *pass caninfo*

Response example: *CAN enable: 2 Manufacturer 2 Type 1 Active 0*

* CAN mode is LCV, vehicle manufacture group Mercedes, type is Mercedes1, CAN is in silent mode.

4.1.7 cansinfo

This SMS command is used to get information about CAN1 and CAN2 interface settings of FM device. These settings are used to see the exact dual CAN interface setup in configuration file (it doesn't necessarily reflect actual mode of operation). Answer SMS format: *CAN1 enable: X Manufacturer Y Type Z Active A; CAN2 enable: X Manufacturer Y Type Z Active A (see caninfo message for fields description)*

Example: *pass cansinfo*

Response example: *CAN1 enable: 2 Manufacturer 2 Type 1 Active 0; CAN2 enable: 2 Manufacturer 2 Type 1 Active 0*

4.1.8 clear dtc

This SMS command allows to clear J1939 and J1708 DTC codes from the memory of the FM device.

SMS command

SMS command structure: *pass clear dtc*

Responses

- If command was performed successfully, response will be: *DTC codes were cleared*
- If DTC reading is not configured, response will be: *DTC reading is not configured*

4.1.9 clear obd

This SMS command is used to clear all OBD (On-board diagnostic) related data in device memory (not configuration). It can be assumed as OBD values reset command.
Example: *pass clear obd*
Response example: ***OBD parameters and DTC cleared***

4.1.10 connect

This SMS command is used to force FM device to connect (for one time) to server with custom IP, port and protocol settings. FM device creates dummy record just with header part. Triggered event ID = 0 (zero). SMS has three parameters. SMS format: *pass connect IP,Port,Protocol*

Parameter	Description
IP	32-bit number, commonly known as an Internet Protocol address (xxx.xxx.xxx.xxx).
Port	16-bit number, commonly known as the port number (xxxxx).
Protocol	The principal of communication. There are two available protocols: TCP and UDP.

Example: *pass connect 192.168.0.1,7011,TCP*
Response example: *connection data ok*
After unsuccessful operation device responds: ***connection data incorrect***

4.1.11 coords

This SMS command is used to get current GPS status and coordinates. Response has 8 parameters.

Parameter	Description
Time	Current GMT date & time.
lat.	Current latitude.
long.	Current longitude.
alt.	Current altitude (meters).
sat.	Currently visible satellites.
dir.	Current angle.

hdop	Current HDOP level.
state	Current GNSS state: 1-off, 2-on no fix, 3-on got fix, 4-not responding, 5-sleep, 6-disabled.

Example: *pass coords*
Response example: *2013-04-24 07:01, lat. 46.1443183, long. 11.881766, alt. 217.5, sat. 8, dir. 198.10, hdop 100, state 3*

4.1.12 delrecords

This SMS command is used to delete all records from internal flash memory of the FM device.
Example: *pass delrecords*
Response example: *all records deleted*

4.1.13 dfota

This SMS command is used to update modem firmware of a device with 3G GSM modem. Firmwares of 3G modems are in diff form so it only sends differences between new and current firmware. Because of this, current modem firmware has to be known before the update. Binary file of Modem firmware must be placed in FTP or HTTP server before update.

Modem firmware over the FTP server

The structure of the command would be:

password dfota ftp://<user_name>:<password>@<server_URL>:<port>/<file_path>

<user_name> - String type, the user name for authentication. The maximum size of the parameter is 50 bytes

<password> - String type, the password for authentication. The maximum size of the parameter is 50 bytes

<server_URL> **-** String type, the IP address or domain name of the FTP server. The maximum size of the parameter is 50 bytes

<port> - Integer type, the port of the FTP server. The default value is 21. The range is 1 – 65535

<file_path> **-** String type, the file name in FTP server. The maximum size of the parameter is 50 bytes

Example of FTP update command:

password dfota ftp://test:test@demo.com:21/update/delta.bin

Modem firmware over the HTTP server

The structure of the command would be:

password dfota http://<http_server_URL>:<http_port>/<http_file_path>

<http_server_URL> **-** String type, the IP address or domain name of the HTTP server

<http_port> - Integer type, the port of the HTTP server. The default value is 80. The range is 1 – 65535

<http_file_path> - String type, the file name in HTTP server

Example of HTTP update command:

password dfota http://www.example.com:80/delta.bin

Note: total SMS message size with password, command name and link to file can be maximum 160 symbols long!

Responses to sent commands

After device receives *dfota* command it responds:

When modem update was started successfully:

- *Modem FW update process started. Check modem FW version with "modrev" command later*

When modem firmware update does not start because of bad SMS input:

- *ERROR: DFOTA FW update NOT supported on this modem* – when trying to update modem that is not 3G
- *ERROR: link string too short* – when minimum link requirements are not met
- *ERROR: wrong command syntax* – something wrong with command syntax
- *ERROR: DFOTA FW update process already started* – when update process was started previously
- ERROR: other reason – when is another error not described above

After modem FW update is finished FM device responds (does not work on SMS via GPRS):

Modem firmware updated successfully:

- *Modem FW update process successful. Current modem version: xx*

Modem firmware update failed:

- *Modem FW update process FAILED. Error code: xx*

  *(*error codes are specified in modem documentation)

4.1.14 dmpfconnect

This SMS command is used to immediately connect the device to DMP without waiting for the configured connection frequency.

SMS command

SMS command structure: *pass dmpfconnect*

Response

If command was performed successfully: *OK.*

Once the response is received, the device will attempt to immediately connect to DMP. If there are any update commands pending for that device – these commands will start executing after the device establishes a connection with DMP.

4.1.15 doutreset

This SMS command allows to reset the state of activated DOUTs. This command works in correlation with the “DOUT activation with conditions” functionality, and requires it to be enabled and configured to work.

SMS command

SMS command structure: *pass doutreset*

Responses

- If command was performed successfully, response will be: *DOUT activation mechanism was reset*
- If DOUTs are not active at the time, response will be : *DOUT activation mechanism is inactive*
- If the “DOUT activation with conditions” functionality is not configured, response will be : *DOUT activation is not configured*

4.1.16 econnect

This SMS command is used to force FM device to connect (for one time) to server with custom APN, user, pass, IP, port and protocol settings. FM device creates dummy record just with header part. Triggered event ID = 0 (zero). SMS has five parameters. SMS format: *pass econnect apn,apnLogin,apnPassword,IP,Port,Protocol*

Parameter	Description
APN	An Access Point Name (APN) is the name of a gateway between a GPRS (or 3G, etc.) mobile network and another computer network, frequently the public Internet.
User	User name for APN settings.
Pass	Password for APN settings.
IP	32-bit number, commonly known as an Internet Protocol address (xxx.xxx.xxx.xxx).
Port	16-bit number, commonly known as the port number (xxxxx).
Protocol	The principle of communication. There are two available protocols: TCP and UDP.

Example: *pass econnect apn,apnlogin,apnpass,192.168.0.1,7011,TCP*
Response example: *Emergency connection data ok*
After unsuccessful operation device responds: *Emergency connection data incorrect*

4.1.17 enginevolt

This SMS command is used to automatically measure engine power voltage upper level, when engine is on and lower level, when engine is off. With these voltage levels the device will automatically detect, when the engine is on and when it is off. Measured voltage levels are used in all profiles. Before sending this SMS command, the engine must be on.

The structure of the command would be:

password enginevolt

<password> - String type, the password for authentication. The maximum size of the parameter is 50 bytes

<enginevolt> - Automatic engine detection command.

If message structure is wrong, FM-device answers "*Unknown command*". If message structure is correct, then after FM-device measured upper voltage level, device answers "*Turn off engine*". After this response, engine must be turned off. When after 30 s, device measures lower voltage level. If engine detection is done, device answers "*Engine detection done*". If engine detection is not successful, response from the device is "*Engine detection error*".

4.1.18 forward

This SMS command allows to forward an SMS via the FM device.

SMS command

SMS command structure: *pass forward <phone_number> <text>*

<phone_number> - must not contain “0” or “+” in the beginning of the number (Not +370 or 00370, but 370);

<text> - Supports only GSM 03.38 characters.

Total length of whole command with text may not exceed 160 symbols.

Responses

- *SMS-Forward OK*
- *SMS-Forward Fail*

4.1.19 get3g

The structure of the command would be:

password get3g

No fields are required to fill into the SMS.

Example: *password get3g*

Following responses are provided for the number, which sent the get3g by SMS (response from FM device).

Response format:

automatic/disabled/enabled 3G mode, connected to 3G/GSM (1/2/3/4-profile)

- If device has enabled 3G and status is OK. For all examples current profile is 3:

“enabled 3G mode, connected to 3G (3-profile)”

- If device has enabled 3G but it is not connected to the 3G network:

"enabled 3G mode, connected to GSM (3-profile)”

- If device has disabled 3G:

"disabled 3G mode, connected to GSM (3-profile)”

- If device has automatic selection 3G and it is working in 3G:

"automatic 3G mode, connected to 3G (3-profile)”

- If device has automatic selection 3G and it is working in GSM:

"automatic 3G mode, connected to GSM (3-profile)"

- If HW version does not support 3G:

"Get3G: hw does not support 3G"

4.1.20 getapn

This SMS command is used to get APN (*APN)*, username (*USER*), password (*PSW*), IPs (*IP1, IP2*), ports (*Port1, Port2*) and protocol (*TCP/UDP*) settings from FM device.
Example: *pass getapn*
Response example: *APN: banga User: PSW: IP1: 92.62.134.38 Port1: 9021 IP2: 195.14.173.3 Port2: 9000 TCP/UDP: 0*
*TCP/UDP: 0 – TCP, 1 – UDP.

Note
If the *getapn* response text is very long (for example, the APN, username and password are 32 symbols long each and two servers are used), it might not fit into a single message and be cropped.

4.1.21 getcfg

Structure of getcfg SMS:
password **getcfg** id

This command is only used for getting the current status of the selected parameter. Maximum simultaneously received parameters in one SMS message is limited by SMS message length (Complete answer: 125 symbols. Incomplete answer: 105 symbols).

When parameter ID is correct, FM-device answer: "*ID: XXX,value:XXX;*"

Example: '*ID: 96,value:1000;*'
Value can be string or number.

Multiple parameter values can be requested in one SMS message and when response does not fit in one message then in the end of the message there will be a phrase: "*other values not fit*".

Example:
Request:
"password getcfg 100,101,102,110,120,111,121,130"

Response:
"ID:100,value:1; ID:101,value:aerospace; ID:102,value:laguna; ID:110,value:101.16.17.245; ID:120,value:23451; ID:111,value:m2m.member.com; other values not fit"

Parameters configuration password and SMS password are not reachable with this SMS. In this case, the device answer: *"ERROR: parameter(s) read is forbidden"*

If parameter ID is invalid, answer would be: *„ID:XXX,value:requested ID not found;“*
Example: “*ID:556,value:requested ID not found;“*

Other possible replies when something wrong:
“ERROR: request is empty”
“ERROR: wrong request syntax”
“ERROR: allowed numbers and commas only”
“ERROR: requested parameter ID too big”
“ERROR: requested parameter ID is low”

If SMS configuration is disabled in configuration tool or configuration has password:
“You do not have permission to read the settings“

4.1.22 getdinmode

This SMS command allows to receive information on the configuration of the specified DIN. This applies only to configurable DINs.

SMS command
SMS command structure: *pass getdinmode <DIN#>*
<DIN#> - can have the following values:

- 1 – information for DIN1
- 3 – information for DIN3
- 4 – information for DIN4

SMS command example

pass getdinmode 3 – This SMS command will provide information on the DIN3 configured mode.

Responses

- *DIN# - positive mode;*
- *DIN# - ground mode;*
- *Configurable DIN functionality is not available for your HW version.*

4.1.23 getgfwver

This SMS command is used to check if the GNSS module firmware is updated for FM-Eco4 S devices.
SMS command structure: *pass getgfwver*
Response if the GNSS module firmware has been updated to the latest version (CAR01A11.gfw):

- *AXN_5.1.6_3333_19010200,0002,MC60-GNSS,1.0*00*

Responses if the GNSS module firmware is older:

- *AXN_5.10_3333_17062100,0002,MC60-GNSS,1.0*20*
- *AXN_3.82_3333_16103100,0002,MC60-GNSS,1.0*2A*

Response for other devices: *Not applicable for this device*

4.1.24 getio

This SMS command is used to get the values/statuses of inputs and outputs. Values: 1 – high, 0 – low. Analog inputs – millivolts.
SMS format: *pass getio*
Answer SMS format:
DIN1=X,DIN2=X,DIN3=X,DIN4=X,DOUT1=X,DOUT2=X,DOUT3=X,DOUT4=X,AIN1=Y,AIN2=Y,AIN3=Y,AIN4=Y
X – Digital value: 1 – high, 0 – low.
Y – analog value in millivolts.
Example: *pass getio*
Response example for HCV5, LCV5, Pro5 devices:
DIN1=0,DIN2=1,DIN3=1,DIN4=1,DOUT1=0,DOUT2=0,DOUT3=0,DOUT4=0,AIN1=24210,AIN2=28600,AIN3=18600,AIN4=28600
Response example for HCV5 Lite/Pro5 Lite devices:
DIN1=0,DIN2=1,DIN3=1,DOUT1=0,DOUT2=0,AIN1=24210,AIN2=28600
Response example for 4th gen devices (except FM-Plug4):
DIN1=0,DIN2=1,DIN3=1,DIN4=1,DOUT1=0,DOUT2=0,AIN1=24210,AIN2=28600

4.1.25 getioparam

Structure of getioparam SMS:
password **getioparam** id,profile

Example: *password getioparam 28,3*

This command is only used for get the current status of the IO parameter selected.

- If IO parameter was enabled, the FM-device would answer:

Example: ‘‘*id=28,profile=3,enable=1,level=0,delta=0,average=1000,eventon=2,include=0,priority=0,switch=0,edge=0”*

- If IO parameter was not enabled, the FM-device would answer:

“I/O ID XXX is NOT enabled“

- If IO parameter ID is invalid, the FM-device would answer:

“ERROR: I/O ID XXX does not exist“

- Other possible replies when is something wrong:

“ERROR: I/O ID XXX read is forbidden”
“ERROR: more than one I/O parameter with same ID was found”
“ERROR: wrong request syntax”
Where XXX – IO ID

- If SMS configuration is disabled in configuration tool or configuration has password:

“You do not have permission to read the settings“

4.1.26 getlog/stoplog

Parameter	Description
IP	32-bit number, commonly known as an Internet Protocol address (xxx.xxx.xxx.xxx).
Port	16-bit number, commonly known as the port number (xxxxx).
Mode	1 – send full log to server. All other values reserved for future functionalities.
Time	Time period for log sending. Defined in minutes. Range from 1 to 60 minutes.

SMS structure: *pass getlog <ip>,<port>,<mode>,<Time>*
Example: *password getlog 100.101.102.103,1111,1,15*

When device gets this command, it should open a link with defined server and send log according to defined mode. All other FM device processes (e.g. records sending, peripheral device communication) should keep working. Only device restart can stop log data sending process.
If data sending is disabled by configuration, log will not be sent and the device's response will be: "Data sending is disabled."
Log will be sent as ASCII. Log starts from device type [HCV/LCV/PRO], selected mode, defined time, bootloader version, firmware version, hardware version, GSM signal level and Voltage status: 0 –lower than 8 Volts (bad), 1 –higher than 8 Volts (OK) and device IMEI.

Getlog
This command initiates logging process.
Example: *FMTco4 HCV, 100.101.102.103:1111, mode=1, time=3min; version:472C,00.02.25.21,431,26,1; imei:868324027387005*

Responses:

- If getlog SMS format is correct, the device will respond: *getlog ok*
- If getlog SMS format is incorrect, the device will respond: *getlog data incorrect*
- If logging proces is already running, the device will respond: *getlog already running*

Stoplog
This SMS command stops log sending to the server.
Example: *pass stoplog*

Responses:

- When device gets this command and log sending is in progress, it should stop sending log to server and reply with SMS: *stoplog ok*
- If device gets this command and the log is not running, then the reply should be: *getlog already stopped*

Disclaimer: In some cases remote debug might have to log data about many different events and processes. This would force remote debug to generate large internal data streams. Device might not be able to send it all via GPRS. This might lead to data loss.

4.1.27 getnetw

This SMS command is used to get the information about the cellular network type. This command uses the following structure:

pass getnetw

Response structure can be the following: *<network mode> connected to <network> <profile>*
<network mode> can be one of the following:

- *automatic mode* – indicates the automatic network mode
- *2G mode* – indicates the 2G network mode
- *3G mode* – indicates the 3G network mode
- *LTE mode* – indicates the LTE Cat M1 network mode

<network> can be the following:

- *2G* – indicates the 2G network
- *3G* – indicates the 3G network
- *LTE* – indicates the LTE Cat M1 network

<profile> - can be one of the following:

- *(1-profile)* – the network mode is set in Profile 1
- *(2-profile)* - the network mode is set in Profile 2
- *(3-profile)* - the network mode is set in Profile 3
- *(4-profile)* - the network mode is set in Profile 4

Response example if the device is set to automatic network selection and is connected to 2G in Profile 1:

- *automatic mode, connected to 2g (1-profile)*

4.1.28 getsd

This SMS command is used to get information about SD card inserted into FM Tco or FM Pro device. There are 4 available answers.

SD card is inserted and used for records: *Using SD Card for Records. Size: sector_count x sector_size B, H: SDrecordHead, T: SDrecordTail*

SD card is inserted and used for log: Using SD Card for Log and working OK. Size: sector_count x sector_size B, H: SDrecordHead, T: SDrecordTail

SD card is inserted and used for log, but some errors occurred: Using SD Card for Log but ERROR. Size: sector_count x sector_size B, H: SDrecordHead, T: SDrecordTail

Parameter	Description
sector_count	the number of sectors in SD card.
sector_size	the size (in bytes) of one sector in SD card.
SDrecordHead	SD card address of the last of record's end.
SdrecordTail	SD card address of beginning of the first record.

SD card is not inserted: *Using Internal Flash for Records*
Example: *pass getsd*
Response example: *Using SD Card. Size: 3911680 x 512 B, H: 6008, T: 5993*

4.1.29 gsminfo

This SMS command is used to get GSM and GPRS information. Response has 16 parameters.

Parameter	Title	Description
ST	Start Time	Date & time (GMT) from the last FM device reset/power ON.
	GSM network	
OP	Operator	GSM operator number
lvl	Level	GSM signal level.
LAC	Location Area Code	16 bit number thereby allowing 65536 location areas within one GSM PLMN.
CID	Cell ID	A GSM Cell ID (CID) is a generally unique number used to identify each Base Transceiver Station (BTS) or sector of a BTS
“”	Network mode	Current network mode. There are two possible values: GSM and 3G.
	FM device modem parameters (**M**)	
I	Initialization	The number of times when FM device has tried to initialize modem since Start Time (ST).
R	Reset	The number of resets of modem since Start Time (ST).
SP	Status Pin	The number of times when modem was turned ON and turned OFF unsuccessfully.
	GPRS service	
GPRS	General Packet Radio Service	Status of GPRS. There are two possible values: 0 – no GPRS/1 – attached to GPRS.
O	Opened	The number of opened GPRS sessions.
C	Closed	The number of closed GPRS sessions.
E	Error	The number of GPRS errors.
	Link with server (LK)	

O	Opened	The number of opened links.
E	Error	The number of link errors,
TMO	Timeout	The number of server response timeout.
Reset		
RS	Reset	Device last reset source. Possible causes: 04 - reset pin; 08 - power on/down reset; 10 - software reset; 20 - independent watchdog; 40 - window watchdog; 80 - Low power reset. 14 means 10 (software reset) + 04 (reset pin) In other words FM device resets itself, for example after FW update.
P	Protocol	GPRS protocol version: 0 – standard protocol; 1 – extended protocol.

Example: *pass gsminfo*

Response example: *ST:2013.04.20 23:26:33; OP 22210,lvl 15,LAC 20030, CID: 28289, GSM; M:I 126, R 125, SP: 0; GPRS 0:O 64,C 0, E 248; LK:O 575, E 1, TMO 126; RS: 04; P 0*

4.1.30 imei

This SMS command is used to get device IMEI number.

Example: *pass imei*

Response example: *IMEI: 863071016796615*

4.1.31 immobilizer

This SMS command allows to immobilize the vehicle. This functionality works in correlation with the “Immobilizer” functionality, described in the “DOUT activation with conditions” document. If the “Immobilizer” mode is not configured, the command will not work.

SMS command

SMS command structure: *pass immobilizer <speed>*

<speed> - describes the vehicle speed at which the progressive DOUTs must be toggled.

Responses

- If command was performed successfully, response will be: Immobilizer OK
- If no custom DOUTs are configured, the device will respond with the following message: *immobilizer is not configured.*
- If <speed> parameter was inputted incorrectly, only numbers from 1 to 250 can be sent, the response will be: *Invalid speed*.

SMS command to reset functionality

SMS command structure: *pass resetimmob*

Responses

- If the immobilizer is successfully reset, the response will be: *immobilizer reset*

4.1.32 info

This customizable SMS command with configurable set of parameters can be used to get some specific information from the FM device via SMS.

Parameter	Description
ain1	Is used to get the status/values of inputs/outputs. Values: 1 – high, 0 – low. Analog inputs – millivolts. Response will contain only values of requested parameters.
ain2	
ain3	
ain4	
din1	
din2	
din3	
din4	
dout1	
dout2	
dout3	
dou4	
alt	Current altitude
apn	APN name
apnp	APN password
apnu	APN username
bat	Returns battery voltage in millivolts.
bleapp	BLE App version
blebl	BLE BL version
blemac	BLE MAC address
blesd	BLE SD version
blest	BLE status
blver	Bootloader version
cid	GSM Cell ID
coords	Current latitude and longitude separated by comma (,). Device will reply “GNSS data not available” if there will be no GNSS fix
dir	Current angle
eng	Current engine state: 0 – OFF, 1 – ON.
extver	(For HCV5 Lite/Pro5 Lite) Extender FW version
extstat	(For HCV5 Lite/Pro5 Lite) Extender status
fpart	(For Trace5-2G, Trace5-LTE and Trace5-LTM (Rev.A)) Currently used partition number
fwver	Firmware version
get3g	Answer according to “get3g” SMS command description
gmap	Hyperlink to Google maps with coordinates: latitude and longitude.
gprs	Status of GPRS. There are two possible values: 0-no GPRS, 1-attached to GPRS.
gprsc	The number of closed GPRS sessions.
gprse	The number of GPRS errors.
gprso	The number of opened GPRS sessions.
gps	Current GPS/GLONASS state: 1-off, 2-on no fix, 3-on got fix, 4-not responding, 5-sleep, 6-disabled.
gsmlvl	GSM signal level
hdop	Current HDOP level

Parameter	Description
hwid	HW identifier
hwver	Hardware version
imei	Device IMEI number
io[xxxxx]	Used to get specific IO current value. [xxxxx] contains IO ID number. Only activated in configuration IO data can be retrieved. For multi-part IO elements, first IO ID should be requested. IO parameters that are being reset with record sending should not be available, response would be NA.
ip1	IP1 address and port number
ip2	IP2 address and port number
lac	Location Area Code
lke	The number of link errors.
lko	The number of opened links.
lktmo	The number of server response timeout.
modi	The number of times when FM device has tried to initialize modem since start time.
modr	The number of modem resets since start time.
modrev	Used to get modem revision information.
modsp	The number of times when modem was turned ON and turned OFF unsuccessfully.
op	GSM operator number
p	GPRS protocol version: 0 – standard protocol; 1 – extended protocol.
prot	Currently set protocol: 0 - TCP or 1 - UDP.
pwr	Returns power supply voltage in millivolts.
resc	Denotes, whether the device is in rescue mode: • 0 – the device is in regular mode • 1, 2 - the device is in rescue mode
rs	Device last reset source. Possible causes: 04 - reset pin; 08 - power on/down reset; 10 - software reset; 20 - independent watchdog; 40 - window watchdog; 80 - Low power reset. 14 means 10 (software reset) + 04 (reset pin) In other words FM device resets itself, for example after FW update.
sat	Currently visible satellites
speed	Current speed, km/h
st	Date & time (GMT) from the last FM device rest/power ON.
tacho	Current tachograph status information: 0 – Not available; 1 – everything is OK; 2 – tacho parameters not configured; 3 – extender not responding; 4 – tacho task in progress; 5 – physical connection OK, logical is not OK.
time	GNSS date and time

Password, command and command text are separated by space symbol. If there is no password, then you need to write just space symbol before the command. Command text parameters are separated by semicolon (;) symbol. Response command text parameters are separated by semicolon (;) symbol. All content of SMS command is not case sensitive.

x`

SMS structure: *pass info param1;param2;param3;[...];paramN*
Example 1: *pass info bat;alt;IO29;IO123;coords*
Response example: *4075;213.9;11659;0x4C545531323334353637383930313233340000000000000;54.7412916,25.2227750*

Example 2: *pass info time;imei;eng;IO29*
Answer: *2016-12-13 12:40:30;865789021340700;0;11691*

Maximum length of a response is 160 symbols. If response is longer, then the device will send maximum 157 characters of response plus “#” symbol and two digits number that shows, which tag was last in the reply counting from the beginning.
Example: *pass info gmap;coords;IO123;pwr;st;time*
Answer: *https://www.google.com/maps/?q=54.7413016,25.2227550;54.7413016,25.2227550;0x4C5455313233343536 373839303132333400000000000000;11659;2016-12-13 11:20:02;#05*

If some requested IO is not enabled in configuration or some tag is unknown, then the device will reply: *NA*

Example: *pass info time;imei;eng;IO29;lktma*

Answer: *2016-12-13 12:41:57;865789021340700;0;NA;NA*

4.1.33 iqfinfo

This SMS command is used to obtain the values of various iQFreeze parameters.

SMS command

SMS command structure: *pass iqfinfo*

The response includes the following parameters:

- Refrigerator serial number
- Trailer serial number
- Device serial number
- Device FW version
- Device BT name
- Current time and date (UTC)
- Vehicle registration number

If an error occured, the FM device will send the following response: *Communication error*

4.1.34 iqfset

This SMS command is used to manage refrigerator set-point temperature values and configuration.

SMS command

SMS command structure: *pass iqfset <setpoint>,<cycle mode>*

<setpoint> value range: from -99.9 to 99.9. A dot must be used as the delimiter symbol.

<cycle mode> can have the following values:

- 1 – the refrigerator is in continuous mode;
- 2 – the refrigerator is in start/stop mode.

If an incorrect symbol or value is included in the command, or no value is included, the FM device will send the following response: *Syntax error*

Any of the parameters can be omitted, only the provided parameters will be changed.

Examples:

pass iqfset ,2 – only the cycle mode will be changed;

pass iqfset 10.5, – only the set-point will be changed.

4.1.35 lastchange

This SMS command is used to get information about when the last time the device configuration or primary server IP was changed.

The date and time of the configuration change will be updated during any type of configuration modification.

When the device is connecting to GPRS and tries to open a link (with different IP settings), the date and time of the IP change will be updated. When the device reconnects to a server with the same IP but a different port, the date and time of the IP change will not be updated.

The latest changes are tracked when the internal device clock is synchronized (upon GNSS fix, synchronizing with an NTP server, or connecting to a mobile network).

Command Structure

SMS command structure: <password> *lastchange*

Responses

Response example: *lastchange ip: 2015-11-09, 09:02; cfg: 2015-11-09, 10:02*

If the device configuration and server settings were never changed: *lastchange ip: no time available; cfg: no time available*

4.1.36 lcvselect

This SMS command is used to automatically select a correct LCV group and LCV subgroup combination. It reads and analyses CANbus line data. After that it searches for a match within its LCV profile database and selects the best possible configuration.

With the SMS message user can also provide “distance” and/or “RPM” values. These values are used by the auto selection process to make a more accurate selection. Additionally the user can add a optional parameter to enable all possible IOs for the selected LCV group on the active profile after the process of LCV group selection is complete.

Note
Enable IO parameters function requires the use of Protocol v1.1, without it enabled, this function will not work. Additionally enough free IO slots must be available to accommodate the newly enable IO parameters.

Configuration compatible with “LCV auto select” feature

The LCV auto selection functionality will work only in CAN “silent” mode. This functionality will not work with Toyota1, Toyota2 groups, because CAN “active” mode for these groups is needed. Also, Tractor1, Tractor2, Truck1, Truck2, Claas1 and Komatsu1 groups will not work.

LCV selection can be performed, if vehicle engine is started, vehicle is not moving and if one of these four statements is true:

1. In both CAN1 and CAN2 interfaces LCV modes are selected.
2. In one CAN interface LCV mode is selected and another CAN interface is not enabled.
3. In one CAN interface LCV mode is selected and another CAN interface Tacho read, MobileEye or Trailers are selected.
4. Both CAN interfaces are not enabled.

SMS command

The structure of the command is:
password lcvselect distance,RPM,enableIO

<password> - String type, the password for authentication. The maximum size of the parameter is 50 bytes
<lcvselect> - Automatic select of correct LCV group and sub group combination command name.
<distance> - Distance travelled by a vehicle, which is currently displayed by the vehicle's odometer. Make sure to enter the same value, it has to be a natural number. Error cannot exceed ± 1 km. Maximum value, which could be sent via SMS is 21055406 km.
<RPM> - RPM value currently displayed in the vehicle's dashboard. Make sure to enter the same value, it has to be a natural number. Error cannot exceed ± 100 rev/min. Maximum value, which could be sent via SMS is 8031 rev/min.
<enableIO> - Command to automatically enable IOs for the automatically selected LCV group. If command is not included or a “0” is written, then the function will not automatically enable IO parameters. If “1” is selected, IOs will be enabled if the condition are met.

Examples of the command:
pass lcvselect 10200,900,1

Other examples:

- If you do not want to send distance and RPM values via SMS, the SMS command should look as follows:
  “pass lcvselect ,“
- If you want to send only RPM value, the SMS command should look as follows:
  “pass lcvselect ,RPM“
- If you want to send only vehicle distance value, the SMS command should look as follows:
  “pass lcvselect distance,“
- If you do not want to send distance and RPM values via SMS, but enable automatic IO selection, the SMS command should look as follows:
  “pass lcvselect ,,1”

Responses

If SMS structure is correct and LCV auto select process successful started, then the FM-device responds:
“LCV auto selection is in progress”

If SMS something went wrong and LCV auto select process cannot be started, the device responds with an error message. One of these error messages can be received:

- *“Unknown command”* - LCV auto select SMS structure is incorrect.
- *“Wrong CAN configuration”* - Current FM device configuration is incompatible with LCV auto select feature. Correct configuration options are described below in the “Configuration compatible with “LCV auto select” feature” section.
- *“LCV autoselection not started, try again”* - This message signals that “LCV auto select” should be restarted.
- *“Waiting for LCV autoselection start”* - Currently a critical process (i.e. CFG task) is running in the FM device. Within 2 minutes period device constantly checks process status. If it has concludes and there are no other critical processes, then the auto select will start. User will be notified with the *“LCV auto selection is in progress”* message. If critical processes are still running, then the auto select will not be started. User will get the “*LCV autoselection not started, try again”* message.

- *"Parameters are out of ranges"* - Distance and/or RPM value received via SMS are out of accepted range.
- *"No CAN data were received"* - No CAN data packets were received.
- *"No valid matches were found"* - LCV auto select task concluded, but it failed to find matches in our database.
- "*Configured group: [Group]/[Subgroup],can1 ch:[number], can2 ch:[number],ERR: Not enough I/O slots* " - group is selected, but there are not enough free IO slots to enable all the auto-select IOs, IOs are not enabled.
- "*Configured group: [Group]/[Subgroup], Error: old I/O protocol is used*" - protocol v1.1 was not enabled prior to the SMS command.

Selection logic

"LCV auto select" feature is designed to choose the best group / sub-group combination for your specific vehicle. In order to do so, it employs a rating system, where each CANbus parameter has its own weight coefficient. Coefficient values within each sub-group are added together. This allows to compare different group / sub-group combinations and select only the best ones.

CANbus parameters themselves also undergo a verification procedure. In order to add weight to a specific sub-group, they have to have valid values. Valid value could mean that it has to be equal to something or it has to behave in a certain faction, or it has to be in a valid range, etc. Rules for validation are defined internally.

Currently, "Fuel", "Vehicle distance", "RPM" and "Wheel based speed" parameters are assigned different coefficient values. Remaining CANbus parameters are rated lower.

After completing the selection process, the FM device will send a SMS message with the information about selected group and sub-group. For example:

"Configured group: Ford/Ford2"

LCV group and subgroup will be configured automatically.

"*Configured group: [Group]/[Subgroup],can1 ch:[number], can2 ch:[number],I/O elements enabled* "

LCV group, subgroup and IO parameters will be configured automatically

Notes

- LCV Auto select functionality does not interfere with IO list configuration unless *enableIO* command is used.
- LCV Auto select does not guarantee that all CAN IO parameters supported for currently selected LCV Group and Sub-group will be available for this vehicle.
- *EnableIO* parameters are enabled in the active profile.

4.1.37 modrev

SMS message is used to get modem revision information. Answer SMS format: *Modem revision: <info>*

Example: *pass modrev*

Response examples:

- M95 modem: *Modem revision: M95FAR02A06*
- UG96 modem: *Modem revision: UG96LNAR02A04E1G*
- MC60 modem: *Modem revision: MC60CAR01A07*
- For the BG96 modem device returns the full version: *Modem revision: BG96MAR02A07M1G_01.018.01.018*

4.1.38 neconnect

This SMS command is used to initiate the connection to a server once, using connection settings in the SMS command. The tracking device does not send any responses to this command.

SMS command:

SMS command structure: *pass neconnect <APN>,<APNLogin>,<APNPassword>,<IP>,<Port>,<Protocol>*

Where:

<APN> - Access Point Name (APN) is the name of a gateway between a GPRS (or 3G, etc.) mobile network and another computer network, frequently the public Internet.

<APNLogin> - User name for the APN settings.

<APNPassword> - Password for the APN settings.

<IP> - 32-bit number, commonly known as an Internet Protocol address (xxx.xxx.xxx.xxx).

<Port> - 16-bit number, commonly known as the port number (xxxxx).

<Protocol> - The principle of communication. There are two available protocols: TCP and UDP.

SMS command example

pass neconnect APN,APNlogin,APNpassword,92.62.134.34,9015,TCP

4.1.39 nreset

This SMS command is used to initiate the restart of the tracking device. The tracking device does not send any responses to this command.

SMS command

SMS command structure: *pass nreset*

4.1.40 optiver

This SMS command is used to request the version of the OptiTemp device.

SMS command

SMS command structure: *pass optiver*

Response examples:

- *Version: TMS X2 V2.100*
- *Device does not respond* – the version could not be retrieved

4.1.41 plock

This SMS command is used to lock/unlock possibility of changing IP and port parameters. Lock/unlock passwords are 32-byte long and are provided by Ruptela support. Be careful with this SMS command.

SMS format: *pass plock plockPassword*

Example: *pass plock aNmuxyBxxr83jumWuBkx1rxkq8eZeaeC*

Response example: *plock OK*

If the operation was unsuccessful then the answer is: *plock ERROR*

4.1.42 plockinfo

This SMS command is used to retrieve the status of currently locked parameters and passwords.
SMS format: *pass plockinfo*
When the command is received, the FM device checks if the following safety mechanisms are currently activated in the device:

- IP lock
- Port lock
- APN lock
- The FM device is locked to the SIM card
- SIM card PIN code
- Configuration password

Response structure: *plockinfo: <IP lock>,<port lock>,<APN lock>,<SIM lock to device>,<PIN code>,<CFG password>*
Each parameter can have 2 values:

- 0 – disabled
- 1 – enabled

Response example: *plockinfo: 1,1,1,0,0,1*
In the example above, IP lock, port lock,APN lock and configuration password are active.

4.1.43 reset

This SMS command is used to reset FM device. Device will reboot, its configuration parameters will not be lost.
Example: *pass reset*
Response example: *Resetting device*

4.1.44 set3g

The structure of the command would be:
password set3g 1/0/2
The field enable/disable could have three values: '1' means enabled, '0' means disabled and '2' means automatic. Only current profile would be changed.

Example: *password set3g 1*
In this example, 3G would be enabled.

Response from FM device:
Following responses are provided for the number, which sent the set3g by SMS.

- Response format:

3G automatic/disabled/enabled mode set OK/FAIL (1/2/3/4-profile)

- If device enabled 3G. For all examples current profile is 1:

"3G enabled mode set OK (1-profile)"

- If device could not enabled 3G:

"3G enabled mode set FAIL (1-profile)"

- If device disabled 3G:

"3G disabled mode set OK (1-profile)"

- If device could not disable 3G:

"3G disabled mode set FAIL (1-profile)"

- If device change to automatic 3G:

"3G automatic mode set OK (1-profile)"

- If device could not change to automatic 3G:

"3G automatic mode set FAIL (1-profile)"

- If field enabled/disabled is out of range:

"Set3G ERROR: Field is out of range"

- If HW version does not support 3G

"Set3G: hw does not support 3G"

4.1.45 setcfg

This SMS command is dedicated to change FM parameters. After SMS password, user inputs parameter ID and parameter value:

password **setcfg** ParamID1 Value1, ParamID2 Value2, ParamID3 Value3, …

Examples:

password setcfg 101 wave, 102 pioneer, 103 , 100 1
Sets APN name: wave, APN user: pioneer, no APN password, Protocol: UDP

password setcfg 4362 6, 4202 1, 4242 12, 4282 5, 4322 10
Sets IO in 4 profile, slot 3. IO modem temperature: enabled, level is 12, delta is 5, average is 10, event on - hysteresis on both

Only parameters provided with SMS are changed, others remain the same.

One SMS message can be 160 symbols long, so the maximum simultaneously transmitted parameters count in SMS message are limited to one message length.

Setcfg command, allows user to change all parameters of the configuration except timetable (because it will not fit into one SMS message), passwords (configuration and SMS) and "enable SMS configuration".

Following responses are provided for the number, which sent configuration by SMS:

- When successfully setting parameters for FM device: "Configuration parameter(s) was set!"
- Incorrect parameter setting: "*Configuration parameter(s) was NOT set! Parameter No. 1 is incorrect*".
- Unsuccessful set-up, when parameters are locked: "*Configuration parameter(s) was NOT set! Parameter No. 1 is*

locked“

- Unsuccessful set-up, when setting IO parameters, but not every setting was provided by SMS: “*Configuration parameter(s) was NOT set! IO slot No. 9 is not fully set*“.
- Unsuccessful set-up, when device is busy: “*Configuration parameter(s) was NOT set! Device is busy try again later*“
- If SMS configuration is disabled in configuration tool: “*You do not have permission to change the settings*“.

SMS configuration feature is enabled/disabled in configuration tool, in Authorized numbers section, by putting a checkbox by “Enable SMS configuration“.

4.1.46 setconnection

This SMS command is used to permanently change FM device configuration settings: APN, APN username, APN password, protocol, IP1, PORT1, IP2, and PORT2 (parameter description: 3.1.6).

SMS format: *pass setconnection apn,apnlogin,apnpassword,Protocol,IP1,Port1,IP2,Port2*

Example: *pass setconnection apn,apnlogin,apnpass,TCP,111.111.111.111,1111,222.222.222.222,2222*

Response example: *set connection data ok*

If one of the parameters should be preserved, then the specific location for the parameter should be filled with **old**. For example, ip1 and port1 should be preserved (old value should remain):

Example: *pass setconnection apn,apnlogin,apnpass,TCP,*old*,*old*,222.222.222.222,2222*

Response example: *set connection data ok*

If the message is not ending with port2, then those parameters, which are not mentioned in the message should not be changed. For example ip2 and port2 were not in the message:

Example: *pass setconnection apn,apnlogin,apnpass,TCP,111.111.111.111,1111*

Response example: *Set connection data ok*

If configuration failed to set, when FM device sends response: *Set connection data incorrect*

4.1.47 setdinmode

This SMS command allows to set the DIN mode (Positive mode or Negative mode) for configurable DINs.

SMS command

SMS command structure: *pass setdinmode <din#> <mode>*
<din#> – can have the following values:

- 1 – selects DIN1
- 2 – selects DIN2 (Trace5 only, excluding the pre-revision variants)
- 3 – selects DIN3
- 4 – selects DIN4

<mode> – can have the following values:

- 0 – positive trigger inputs (positive mode);
- 1 – ground trigger inputs (negative mode);

SMS command example

pass setdinmode 3 1 – this SMS command will set DIN3 to ground (negative) mode.

Responses

- If DIN# was set to ground (negative) mode, response will be: *DIN mode set: DIN# - ground mode;*
- If DIN# was set to positive mode, response will be: *DIN mode set: DIN# - positive mode;*

If the device does not support configuration of DINs or wrong DIN mode is specified, the following responses can be received:

- *DIN mode set: configuration of DIN# is not supported;*
- *DIN mode set: incorrect DIN specified;*
- *DIN mode set: incorrect DIN mode specified;*
- *Configurable DIN functionality is not available for your HW version.*

4.1.48 setio

This SMS command is used to set digital output levels. Values: 0 – low, 1 – high, 2 – do not change. Remember: the digital outputs must be connected to the electric circuit correctly.
SMS format for the 5th generation devices (except Trace5): *pass setio X1,X2,X3,X4*
SMS format for the 4th generation devices and Trace5 (except FM-Plug4): *pass setio X1,X2*

- *X1* – state of DOUT1
- *X2* – state of DOUT2
- *X3* – state of DOUT3
- *X4* – state of DOUT4

Example for the HCV5, LCV5, Pro5 devices: *pass setio 0,1,1,2*
Response example: *SETIO configuration data ok*

Example for the 4th gen devices (except FM-Plug4): *pass setio 0,1*
Response example: SETIO configuration data ok
If configuration SMS is incorrect, device will respond: *SETIO configuration data incorrect*

Note
The *setio* SMS command will not work with several FM device configurations. If a DOUT is configured to *Blocking* and driver registration is enabled and registration function is set to *Blocking*/or DOUT activation under conditions is enabled and DOUT is set as an output channel, then the *setio* SMS command will have no effect.

4.1.49 setioparam

Structure of setioparam SMS:

password **setioparam**
id=*id*,**profile**=*profile*,**enable**=*enable*,**level**=*level*,**delta**=*delta*,**average**=*average*,**eventon**=*eventon*,**include**=*include*,**prior**

```
ity=priority,switch=switch,edge=edge
```

Example:

```
01234567890123456 setioparam
id=256,profile=4,enable=1,level=3500,delta=250,average=2000,eventon=2,include=1,priority=1,switch=1,edge=3
```

This SMS can be used in any way, but it must at least contain *Id,profile* and *enable* parameters. Shortest possible message should be written this way:

```
password setioparam id=id,profile=profile,enable=enable
```

Example:

```
01234567890123456 setioparam id=256,profile=4,enable=1
```

If the short version is used the other parameters are set with old values. ***Id,profile*** and ***enable*** parameters are mandatory to make a valid SMS command.

Note that only two spaces between *password*[space]**setioparam**[space]***id*** are required. All other commands are separated by comma.

SMS can be written in lower case and upper case letters. Therefore, **setioparam** SMS is not case sensitive.

If SMS command doesn't define new values to **level***,* **delta***,* **average***,* **eventOn***,* **include***,* **priority***,* **switch** or **edge** parameters, it will not be changed. Previous value will remain.

When the FM-device receives the SMS where the parameter **enable** is set to 1, it searches in the current configuration if this parameter with the same ID is already enabled.

If parameter with the same ID was not enabled, FM-device will set this IO parameter to the first free slot that is available.

If IO parameter with the same ID was enabled, FM-device will overwrite the IO parameter with new values.

If FM-device finds more than one IO parameters with the same ID it will send an error message.

If newly created IO event is not set to specific parameters it will be set to default. Default values are listed below:

Level	Delta	Average	EventOn	Include	Priority	Switch	Edge
0	0	1000	2	0	0	0	0

For set enable, eventOn, include, priority, switch on edge it is necessary to indicate the state with a number:

Enable	EventOn	Include	Priority	Switch	Edge
0 – disable 1 – enable	0 – Hysteresis 1 – Change 2 – Monitoring	0 – not include data 1 – include data	0 – Low 1 – High	0 – no switch 1 – 1st profile 2 – 2nd profile 3 – 3rd profile 4 – 4th profile	0 – On Both 1 – On rising 2 – On falling

Following responses are provided for IO configuration by SMS:

- If the IO parameter was set correctly: *setioparam OK slot: XX*.

- Where *XX* – is the slot number, where the parameter is enabled.
- If the IO ID is not entered: *setioparam ERROR, I/O ID field is mandatory*.
- If the IO ID is too long: *setioparam ERROR, I/O ID field limits exceed.*
- If the non-existing IO ID is entered: *setioparam ERROR, this I/O ID does not exist.*
- If the IO profile is not entered: *setioparam ERROR, I/O profile field is mandatory*.
- If the IO profile is too long: *setioparam ERROR, I/O profile field limits exceed.*
- If the IO enable is not entered: *setioparam ERROR, I/O enable field is mandatory*.
- If the IO enable is too long: *setioparam ERROR, I/O enable field limits exceed.*
- If SMS command is too long: *setioparam ERROR, I/O parameter limits exceed.*
- If SMS command is not written correctly: *setioparam ERROR, wrong request syntax.*
- If all slots are full: *setioparam ERROR, no free slots for set the I/O.*
- If the tracking device finds more than one IO parameter with the same ID: *setioparam ERROR, more than one I/O with the same ID.*

- If a user sent the SMS to disable and the parameter is not enabled: *setioparam ERROR, parameter is already disabled*.
- If some other error occurs: *setioparam ERROR, other.*
- If the SMS configuration is disabled in the confiugration tool: *You do not have permission to change the settings*.

4.1.50 setiotime

Using this feature, FM is able to switch DOUTs for temporary period of time. User must provide pulse lengths for logical '1' & '0' values.

SMS format example for the HCV5, LCV5, Pro5 devices: *pass setiotime 1 500 0 500,0 200 1 300,0 500 1 600,1 700 0 800*

SMS format example for the 4th gen devices (except FM-Plug4): *pass setiotime 1 500 0 500,0 200 1 300,*

Description:

Parameter	Description
pass	SMS password
setiotime	Command Identifier
1	DOUT1 first logical state (1/0)
500	DOUT1 first logical state length (ms)
0	DOUT1 second logical state (1/0)
500	DOUT1 second logical state length (ms)
,	DOUT separator
0	DOUT2 first logical state (1/0)
200	DOUT2 first logical state length (ms)
1	DOUT2 second logical state (1/0)
300	DOUT2 second logical state length (ms)
,	DOUT separator
0	DOUT3 first logical state (1/0)
500	DOUT3 first logical state length (ms)
1	DOUT3 second logical state (1/0)
600	DOUT3 second logical state length (ms)
,	DOUT separator
1	DOUT4 first logical state (1/0)

700	DOUT4 first logical state length (ms)
0	DOUT4 second logical state (1/0)
800	DOUT4 second logical state length (ms)

Impulse resolution is 10 ms. Minimum impulse duration is 10 ms. If an Eco-panel is connected, minimum resolution and duration is 50 ms. When the FM device receives the *setiotime* command, it stores the status of DOUT's and after the sequence restores the previous status. If one *setiotime* command interrupts another, the state is **UNDEFINED** ('1' or '0') after both sequences are finished. Interrupt in this case means that several consecutive *setiotime* commands were sent, one overlapping the other. Sending several *setiotime* commands in a fast succession will make the the DOUT stuck, thus requiring a device reboot. It is highly recommended to at least wait until a response is received for the first sent *setiotime* command, before sending the next.

'0' means **GND** and '1' means **no GND**.

If one of the DOUTs is configured as *LED*, *Buzzer*, *Blocking* or *Jamming block*, *setiotime* cannot be used.

Additional feature. It is possible to set a repeat amount of a sequence.

SMS format example for the HCV5, LCV5, Pro5 devices: *pass setiotime 1 500 0 500 n=10,0 200 1 300 n=20,0 500 1 600 n=30,1 700 0 800 n=40*

SMS format example for the FM-Tco4 HCV, FM-Tco4 LCV, FM-Pro4 devices: *1 500 0 500 n=10,0 200 1 300 n=20*

Description:

Parameter	Description
pass	SMS password
setiotime	Command Identifier
1	DOUT1 first logical state (1/0)
500	DOUT1 first logical state length (ms)
0	DOUT1 second logical state (1/0)
500	DOUT1 second logical state length (ms)
n	Repetition identifier
=	Setter symbol
10	Amount of repetitions
,	DOUT separator
0	DOUT2 first logical state (1/0)
200	DOUT2 first logical state length (ms)
1	DOUT2 second logical state (1/0)
300	DOUT2 second logical state length (ms)
n	Repetition identifier
=	Setter symbol
20	Amount of repetitions
,	DOUT separator
0	DOUT3 first logical state (1/0)
500	DOUT3 first logical state length (ms)
1	DOUT3 second logical state (1/0)
600	DOUT3 second logical state length (ms)
n	Repetition identifier
=	Setter symbol
30	Amount of repetitions

,	DOUT separator
1	DOUT4 first logical state (1/0)
700	DOUT4 first logical state length (ms)
0	DOUT4 second logical state (1/0)
800	DOUT4 second logical state length (ms)
n	Repetition identifier
=	Setter symbol
40	Amount of repetitions

Each single DOUT has 10 slots for impulses levels (high / low). No more than 10 can be defined for one DOUT.

Longest time possible is 999 999 999 ms. Max number of repeats is 9999.

Max impulse count – as many as you can fit into 160 symbols of SMS.

It is possible to interrupt a sequence with “pass setiotime 0 10,0 10” followed by setio SMS command “pass setio 1,1”.

The sequence would be interrupted and the states of DOUT’s would be defined.

It is possible to set only one of the DOUTs.

SMS format example to set DOUT1: *pass setiotime 1 500 0 500*

SMS format example to set DOUT2: *pass setiotime ,0 200 1 300*

SMS format example for set DOUT3: *pass setiotime ,,1 500 0 500,*

SMS format example for set DOUT4: *pass setiotime ,,,1 500 0 500*

Response example: *setiotime set OK*

If configuration SMS is incorrect, device will response: *setiotime syntax error in DOUTX settings: error text*

Where „DOUTX“ is the specific DOUT. „Error text“ is described below in the table. It's the minimum diagnostic when something is wrong with the impulse set in SMS.

Error text	**Error description**
other process controls output	Means that other functionality is configured on DOUT's. For example: LED or buzzer
no comma symbol	No comma separator. Comma is used to separate DOUT's configuration
wrong format	When was received not enough symbols
wrong level	Level can be only logical '1' or '0'. Other values is unacceptable
no space symbol	After level symbols must to be space symbol
wrong ms number	Wrong time number
wrong repeat number	Wrong repeat count number
exceeded max slot	For one output is possible to set maximum 10 slot. If this count is exceeded this error is displayed
pulse is shorter than 10ms	Pulse length less than 10 milliseconds was set. It's not acceptable
other reason	Includes other reason not described in this table

4.1.51 setlcv

This SMS command is used to set a specific LCV group, subgroup and additionally enable IOs which are assigned to this group, thus reducing the amount of work needed for configuring the FM device.

The structure of the command is:
pass setlcv <Group_Name>,<Sub_Group_number>,<CAN_interface_number>,<Silent/Active>,
<Channel_X>,<Additional_CAN_interface>

Excluding the command itself, only the *<Group_name>* ,*<Sub_Group_number>* and *<CAN_interface_number>* fields are mandatory. All other fields are optional. Default values for the optional parameters:

- *<Silent/Active> = Silent;*
- *<Channel_X> = 1;*
- *<Additional_CAN_interface> - not active.*

Notes

- The SMS command is not case sensitive, so *uppercase letters won't have influence when inputting the Group name.*
- *All LCV Group names and subgroup numbers are identical to those in the configurator. The only exception is the Caterpillar subgroup, which does not have a number in the configurator. In this case the <Group_name> would be "tractor" and <Sub_Group_number> - "2". (Example: pass setlcv tractor,2,1)*

<pass> – the password for authentication. The maximum size of the parameter is 50 bytes.
<setlcv> – command to set lcv group, subgroup and other parameters to the ones specified.
<Group_name> – LCV group name, which the user wants to enabled (Example: vag).
<Sub_Group_number> – LCV subgroup number, which the user wants to enable. Note that only the **number** is used, not the whole subgroup name like in the configurator (Example: name in configurator is VAG17, then <Sub_Group_name> = 17)
<CAN_interface_number> – CAN interface on which the selected parameters will be configured. CAN1 = 1; CAN2 = 2.
<Silent/Active> – set the enabled CAN interface to either Active or Silent. Active = 1, Silent = 0.
<Channel_X> – set the Channel for the CAN interfaces. The Channel setting is used when two separate physical connections must be made to receive all parameters from a LCV group/subgroup. Channel1 = 1, Channel 2 = 2.
<Additional_CAN_interface> – the user can specify whether to enable the other CAN interface. Activate = 1, not activate = 0.

Notes
Enable IO parameters function requires the use of Protocol v1.1, without it enabled, this function will not work. Additionally enough free IO slots must be available to accommodate the newly enabled IO parameters. The LCV group, and IOs will be set only on the currently active profile, overwriting any previous CAN parameters on that profile.

Examples of the command:
pass setlcv vag,17,1,0,2, 1 – VAG17 subgroup will be enabled on CAN1 interface, silent mode on CAN1 and CAN2, CAN1 will be enabled on Channel 2, CAN2 will also be enabled on Channel 1.

pass setlcv VAG,17,2 – VAG17 subgroup will be enabled on CAN2 interface, all other parameters are set to default settings.

Responses
If SMS structure is correct, Protocol v1.1. was enabled prior to sending the "setlcv" command, LCV group and subgroup

is enabled, the response will be:
“Configured group: VAG17, I/O elements enabled”

If SMS something went wrong , the device responds with an error message. One of these error messages can be received:

- *“Configured group: VAG17, Error: Not enough I/O slots”* - the LCV subgroup was set, but there are not enough free IO slots to enable all the IO parameters for this subgroup.
- *“Configure group: VAG17, Error: old I/O protocol is used”* - the LCV subgroup was set, but Protocol v1.1 is not enabled, thus some parameters cannot enabled, so no parameters are enabled at all.
- *“Error: unknown parameter”* - Group name is misspelled, Subgroup number incorrect or invalid value was provided for one of the parameters.
- “*Error: LCV auto select still in progress*” - The device is currently in the progress of LCV auto select, during which the user cannot set a LCV group or enable IO parameters.
- “*Wrong CAN configuration*” - an incompatible LCV mode is configured on a CAN interface.
- **“*Configured group: Tractor2, Error: Not enough I/O slots*” - If a LCV group contains more than 80 IO elements, it will not be enabled .**

4.1.52 setlock

This SMS command is used for locking the vehicle. This SMS command uses the following structure:

pass setlock <lock> <ATH>

<*lock*> can have the following values:

- *lock* – locks the vehicle if the doors are closed
- *unlock* – unlocks the vehicle
- *mlock* – locks the vehicle regardless of the door state
- *skip* – does not modify the current lock status

<*ATH*> can have the following values:

- *ath-on* – blocks the engine
- *ath-off* – unblocks the engine
- *ath-skip* – does not modify the current engine blocking status

SMS command example if you want to block the engine:

pass setlock skip ath-on

SMS command example if you want to lock the vehicle if the doors are closed:

pass setlock lock ath-skip

Response structure:

- *setlock* was successful: *setlock <lock>-OK <ATH>-OK*
- *setlock* was unsuccessful: *setlock <lock>-FAIL:<parameters> <ATH>-FAIL*

<*parameters*> can have the following values:

- *IGN – ignition*

- *ODW – open door/window error*
- *DFD – door front driver*
- *DFP – door front passenger*
- *DRD – door rear driver*
- *DRP – door rear passenger*
- *DHT – door hatch*

<ATH>-FAIL may occur when:

- the ignition is on
- the ignition state is unknown

Response examples:

- *setlock* was successful: *setlock LOCK-OK ATH-ON-OK*
- *setlock was unsuccessful: setlock LOCK-FAIL:IGN,DFD,DFP,DRD,DRP,DHT ATH-ON-FAIL*

4.1.53 setnetw

This SMS command is used to set the required cellular network type. This command uses the following structure:

pass setnetw <network>

<network> can have the following values:

- *1* – automatic mode
- *2* – 2G mode (GSM)
- *3* – 3G mode (UMTS)
- *4* – LTE Cat M1 mode

Response structure can be the following: *setnetw <network mode> <status> mode <outcome> <profile>*

<network mode> can be one of the following:

- *automatic* – indicates the automatic network mode
- *2G* – indicates the 2G network mode
- *3G* – indicates the 3G network mode
- *LTE* – indicates the LTE Cat M1 network mode

<status> can be one of the following:

- *disabled* – indicates that the network is disabled
- enabled – indicates that the network is enabled.

<outcome> - can be one of the following:

- *set OK* – indicates that the network mode is set successfully;
- *set FAIL* – indicates that the network mode is not set.

<profile> - can be one of the following:

- *(1-profile)* – the network mode is set in Profile 1;
- *(2-profile)* - the network mode is set in Profile 2;

- *(3-profile)* - the network mode is set in Profile 3;
- *(4-profile)* - the network mode is set in Profile 4.

Possible responses from the tracking device:

- Response example if the HW version does not support 3G/LTE:

  setnetw: modem does not support 3g/lte
- Response format if the <*network mode*> value is out of range:

  setnetw error: field is out of range
- Response example if 4G mode was enabled successfully in Profile 1:

  lte mode set OK (1-profile)
- Response example if the device does not support 3G mode:

  setnetw: modem does not support 3g

4.1.54 setvalue

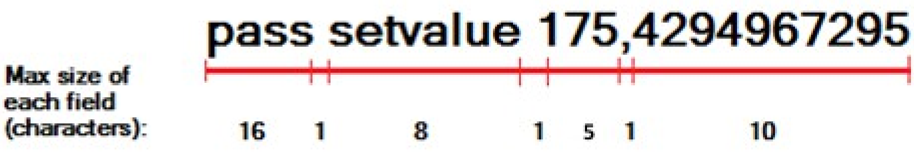
Purpose is to set specific IO values.

Parameter	**Trace5*****	**Advanced family**	**Eco family**	**Plug4**
Virtual odometer (ID:65)	●	●	●	●
ECO Absolute idling time (ID:175)	●	●****	●	●
CANBUS Distance (ID:114)*		●		
DIN1 hours accumulated (ID:577)	●	●	●	
DIN2 hours accumulated (ID:578)	●**	●	●	
DIN3 hours accumulated (ID:579)	●**	●	●	
DIN4 hours accumulated (ID:580)	●**	●****	●	

** – value will be overwritten by valid message from CAN-Bus (if available in specific vehicle).*
*** – not available for pre-revision Trace5 devices.*
**** – (for Trace5-2G, Trace5-LTE and Trace5-LTM (Rev.A)) implemented partially, initially will only support Virtual odometer (ID:65).*
***** – not available for HCV5 Lite/Pro5 Lite devices*

In general SMS structure is: "*(password) (command) (command text)*"

In "command text" field there will be ID and its value separated by comma (,).



```
pass setvalue 175,4294967295
```

The response format depends on these options:

1. If IO value was changed successfully, then format is: ID,value
2. If FM device failed to change IO value, then format is: ID,fail

3. If the value change for specified IO is not supported, then format is: ID,unsupp
4. If the FM device is even failed to parse the ID or data is incorrect: Set IO value data incorrect

Example: *pass setvalue 65,0*	Response example: *65,0*
Example: *pass setvalue 175,0*	Response example: *175,0*
Example: *pass setvalue 155,15*	Response example: *155,unsupp*

4.1.55 snapshot

This SMS command allows taking pictures on demand via connected peripherals with cameras.

SMS command

SMS command structure: *pass snapshot <peripheral number>*

<peripheral number> can have the following values:

- *0* – the command applies to all configured peripherals with cameras;
- *1* – the command applies to the RS232 camera configured on PortA;
- *2* – the command applies to the RS232 camera configured on PortB;
- 3 – the command applies to the fatigue sensor.

Response structure: *snapshot <peripheral number> <answer> , <other peripheral number> <answer>*

<answer> can be one of the following:

- *ok* – indicates that the peripheral took a picture successfully;
- *busy* – indicates that the peripheral is busy with a previously taken picture at the moment;
- *n/a* – indicates that the peripheral is either not configured, not connected or not responding;
- *full* – indicates that the memory of the SD card is full;
- *fullsd* – indicates that the memory of the SD card is full and the picture will be discarded, as there are no pictures to be overwritten;
- *err* – indicates that there was an error.

Example 1: *pass snapshot 1*
Response: *snapshot 1 ok*

Example 2: *pass snapshot 0*
Response: *snapshot 1 ok, 2 n/a*

4.1.56 ssl status

This SMS command allows receiving information about current SSL authentication status.

SMS command

SMS command structure: *pass ssl status*

Response structure: *SSL status server1 <status>,server2 <status>*

<status> can have the following values:

- *0* – SSL authentication is disabled on this server;
- *1* – SSL authentication is enabled on this server.

If OCSP validation is enabled, *<status>* can have additional values:

- *2* – the certificate is valid;
- *3* – OCSP lookup failed;
- *4* – the certificate is revoked;
- *5* – OCSP server URL not found;
- *6* – unknown certificate;
- *7* – validation request timeout;
- *8* – modem firmware does not support OCSP validation.

4.1.57 switchip

SMS message is used to change current primary IP and port (IP1, Port1 or IP2, Port2). This change is temporary and applies only until device reset. After the reset IP and port configuration will be restored to the previous state.
SMS format: *pass switchip X*
X – which IP and port should be primary
IP1 primary set example: *pass switchip 1*
Response example: *Setting primary IP OK*
IP2 primary set example: *pass switchip 2*
Response example: *Setting primary IP OK*
If the operation was unsuccessful, then the answer is: *Setting primary IP FAIL*

4.1.58 tacho

SMS message is used get tachograph status information. Answer SMS format: *TACHO status:X*

Status (*X*)	**Description**
0	Not available.
1	Everything is OK.
2	Tacho parameters not configured.
3	Extender not responding.
4	Tacho task is in progress.
5	Physical communication OK, logical is not OK.

Example: *pass tacho*
Response example: *TACHO status:1*

4.1.59 tachostatus

This SMS command is used to get more detailed information about the tachograph status.

SMS command

SMS command structure: *<password> tachostatus*

Possible responses:

Tacho status 1. Ready for remote tacho download. – If a successful service response and successful routing control response are received

Tacho status 2. Tacho parameters are not configured. – If the tachograph solution is not enabled in the device configuration

The device is busy with critical process. Please try again later. – If the device is downloading a tachograph file at the moment when SMS command is received

Errors can be the following:

Tacho status 0. Error (10 00) – When no response received to service enable request (response timeout). In this case, please check the physical connection.

Tacho status 0. Error (31 00) – When no response received to routing control request (response timeout). In this case, please try again.

Tacho status 0. Error (10 12) – When remote download is not activated in the tachograph. In this case, please contact a tachograph workshop for activation

Tacho status 0. Error (10 22) – When remote download is not possible. In this case, please check the connection to the correct tachograph socket (C port)

If you get other error messages, please contact out technical support.

4.1.60 uptime

This SMS command is used to know how much time in seconds has passed since the last device start-up.

SMS command

SMS command structure: *pass uptime*

Response example: *Uptime 1230*

4.1.61 ussd

GSM network operators are providing various services via USSD communication. Support of the USSD commands allows users to send various USSD request to the GSM operator over the FM device and receive answers. In this case the device operates in the transparent channel principle.

Operation principles

- USSD command is received through the SMS and then passed to the modem;
- USSD command is not affected or ignored by ongoing firmware, configuration updates or tacho and smart card

reading.

- After getting the response, the device transmits the answer to the sender of the SMS;
- Timeout of 120 seconds is used when waiting for the response from the GSM operator.

SMS command

Request SMS structure: *pass ussd *command#*

Where command is a set of digits and asterisk symbols (*). Command always begins with an asterisk symbol ***** and is terminated by **#**. Inside, the command can have several asterisk symbols.

Correct command examples:

**485#*

**52*452*4#*

Incorrect command examples:

52#	*- missing* *****
**45*	*- missing* **#**
**5 65#*	*- blank space in the command*

Responses

- When the operator responds to the ussd command within 120 seconds, the answer is sent in the following format: *ussd response*
- If * or # symbols are missing or command is blank in the request message: *ussd bad format*
- If the response is not received from the operator within 120 seconds: *ussd timeout*
- If there is an error with the GSM network operator: *ussd modem error*

4.1.62 version

This SMS command is used to get current FM device version. Response has 5 parameters.

Parameter	Description
1	Bootloader version.
2	Firmware version.
3	Hardware version.
4	GSM signal level. The signal level ranges from 0 (no signal) to 31 (strong signal).
5	Voltage status: 0 – lower than 8 Volts (bad), 1- higher than 8 Volts (OK). Exception for Eco4 S and Eco4 T: 0 – lower then 6 Volts (bad), 1- higher then 6 Volts (OK).

Example: *pass version*

Response example: *5402,00.02.15,1089,5,1*

4.1.63 webcoords

This SMS command is used to get hyperlink to Google maps with coordinates, current vehicle speed and ignition status.

Parameter	Description
Time	GMT date & time of coordinates
Hyperlink	Hyperlink to Google maps with coordinates: latitude and longitude
Speed	Current vehicle speed, km/h
Ignition	Ignition status

Syntax: *pass webcoords*

Can be 3 different SMS message responses.
1. When GPS data is available.
Response example: *2015-05-11 13:01, https://www.google.com/maps/?q=54.7404933,25.2222366, speed: 94, ignition: ON*
2. When GPS not available.
Response example: *GPS data not available. Ignition: ON*
3. When GPS not available but was available then device can return last known coordinates and the time shows when the coordinates were taken.
Response example: *No GPS. Last entry: 2015-05-16 17:32, https://www.google.com/maps/?q=54.7404933,25.2222366, speed: 65, ignition: ON, current ignition: OFF*

4.2 Informational messages, alerts

4.2.1 SMS alerts with date & time

SMS message can be sent if one of the following SMS alerts is configured: **overspeeding, power supply disconnecting, DIN1, DIN2, DIN3, DIN4 (ignition)**. Purpose of this message is to inform when these events were triggered. Date & time is GMT. To receive SMS alerts, "SMS Alert Number” must be configured.
“SMS Alert Number” can be configured via FM configurator. In the main configurator window, locate the “Authorized numbers” section and click “Options. This will open the “Numbers window”. The second input field in this window is the “SMS alert number”, the user must enter a number which will receive the SMS alerts.

Notes
When inputting the number, the “+” symbol should be replaced with “00”.

IO parameters must be properly configured to be able to send SMS alerts, this means one of the previously mentioned IO parameters must be enabled as shown:

- DIN1, DIN2, DIN3 or Ignition (DIN4) – “Event on” set to **On Change**;
- Power supply voltage - “Event on” set to **Hysteresis** and hysteresis must be set to **On Falling** or **On Both** ;
- GPS speed - “Event on” set to **Hysteresis** and hysteresis must be set to **On Rising** or **On Both**

After the “Event on” parameter is properly set, a new checkbox will appear at the top of the IO properties section called “SMS alert”. Put a tick in this checkbox to receive SMS alerts for the selected I/O parameter.

Afterwards, when a I/O is triggered, the device will send an SMS alert to the number previously specified in the “Authorized number”.

Notes

If the device was out of the GSM coverage, the FM device will accumulate events that will be sent as soon as GSM connectivity is restored. The FM device can store up to 16 events. If the 16 event threshold is reached, the device will start to overwrite oldest events. The device will send a report of all events in **ONE** message (as many, that can fit in 160 characters). Multiple events in one SMS message will be delimited by a semicolon symbol “;”.

Message format:

date1&time1-event1; date2&time2-event2; …

Examples:

2015.05.01 16:24:01-device disconnected;

2015.06.02 22:05:16-ignition: ON; 2015.06.02 22:05:35-DIN1: OFF;

2015.06.14 08:30:45-overspeed;

Additionally the SMS alert function now supports custom SMS text. Custom text fields appear after the “SMS alert” checkbox is clicked. The maximum custom text length is 50 symbols, all previous rules apply.

4.3 SMS during critical process

There are 4 critical processes:

- Firmware update
- Configuration update
- Tacho read
- Smart Card read

During these processes the following SMS commands will be ignored:

- reset
- connect
- econnect
- switchip
- setconnection
- delrecords
- setcfg
- getcfg
- setioparam
- getioparam
- clear obd
- tacho

Response *example: The device is busy with critical process. Please try again later.*

4.4 Supported SMS commands table

	5th generation devices					4th generation devices						
	HCV5	LCV5	PRO5	HCV5 Lite Pro5 Lite	Trace5	Eco4	Eco4 S	Eco4 T	Pro4	Tco4 LCV	Tco4 HCV	Plug4
accinfo	●	●	●	●	●	●	●	●	●	●	●	●
accreset	●	●	●	●	●	●	●	●	●	●	●	●
ahj-on/ahj-off	●	●	●		●							
authid	●	●	●		●**		●	●	●	●	●	
banned	●	●	●	●	●	●	●	●	●	●	●	●
caninfo	●	●	●	●					●	●	●	
cansinfo	●	●	●	●					●	●	●	
clear dtc	●										●	
clear obd	●	●								●	●	●
connect	●	●	●	●	●	●	●	●	●	●	●	●
coords	●	●	●	●	●	●	●	●	●	●	●	●
delrecords	●	●	●	●	●	●	●	●	●	●	●	●
dfota	●	●	●	●	●	●	●	●	●	●	●	
dmpfconnect	●	●	●		●**							
doutreset	●	●	●	●	●	●	●	●	●	●	●	
econnect	●	●	●	●	●	●	●	●	●	●	●	●
enginevolt												●
forward	●	●	●	●	●	●	●	●	●	●	●	●
get3g	●	●	●			●	●	●	●	●	●	
getapn	●	●	●	●	●	●	●	●	●	●	●	●
getcfg	●	●	●	●	●	●	●	●	●	●	●	●
getdinmode	●	●	●	●	●		●	●	●	●	●	
getecu	●	●								●	●	●
getgfwver							●					
getio	●	●	●	●	●	●	●	●	●	●	●	
getioparam	●	●	●	●	●	●	●	●	●	●	●	●
getlog/stoplog	●	●	●	●	●	●	●	●	●	●	●	●
getsd	●	●	●						●	●	●	
gsminfo	●	●	●	●	●	●	●	●	●	●	●	●
getnetw	●	●	●	●	●							
imei	●	●	●	●	●	●	●	●	●	●	●	●
immobilizer	●	●	●		●*	●	●	●	●	●	●	
info	●	●	●	●	●	●	●	●	●	●	●	●
iqfinfo			●						●			
iqfset			●						●			
lastchange	●	●	●	●	●	●	●	●	●	●	●	●
lcvselect	●	●								●		
modrev	●	●	●	●	●	●	●	●	●	●	●	●
neconnect	●	●	●	●	●			●	●	●	●	
nreset	●	●	●	●	●			●	●	●	●	
optiver	●										●	
reset	●	●	●	●	●	●	●	●	●	●	●	●
resetimmob	●	●	●		●	●	●	●	●	●	●	
set3g	●	●	●			●	●	●	●	●	●	
setcfg	●	●	●	●	●	●	●	●	●	●	●	●
setconnection	●	●	●	●	●	●	●	●	●	●	●	●
setdinmode	●	●	●	●	●		●	●	●	●	●	
setio	●	●	●	●	●	●	●	●	●	●	●	
setioparam	●	●	●	●	●	●	●	●	●	●	●	●
setiotime	●	●	●		●	●	●	●	●	●	●	
setlcv	●	●								●	●	
setlock		●			●*					●		

	5th generation devices					4th generation devices						
	HCV5	LCV5	PRO5	HCV5 Lite Pro5 Lite	Trace5	Eco4	Eco4 S	Eco4 T	Pro4	Tco4 LCV	Tco4 HCV	Plug4
setnetw	●	●	●	●	●							
setvalue	●	●	●	●	●				●	●	●	●
snapshot	●	●	●						●	●	●	
ssl status	●	●	●		●*			●	●	●	●	
switchip	●	●	●	●	●	●	●	●	●	●	●	●
tacho	●			●							●	
tachostatus	●			●							●	
USSD	●	●	●		●	●	●	●	●	●	●	
version	●	●	●	●	●	●	●	●	●	●	●	●
webcoords	●	●	●	●	●	●	●	●	●	●	●	●

* – not available with initial firmware release for Trace5-2G, Trace5-LTE and Trace5-LTM (Rev.A).

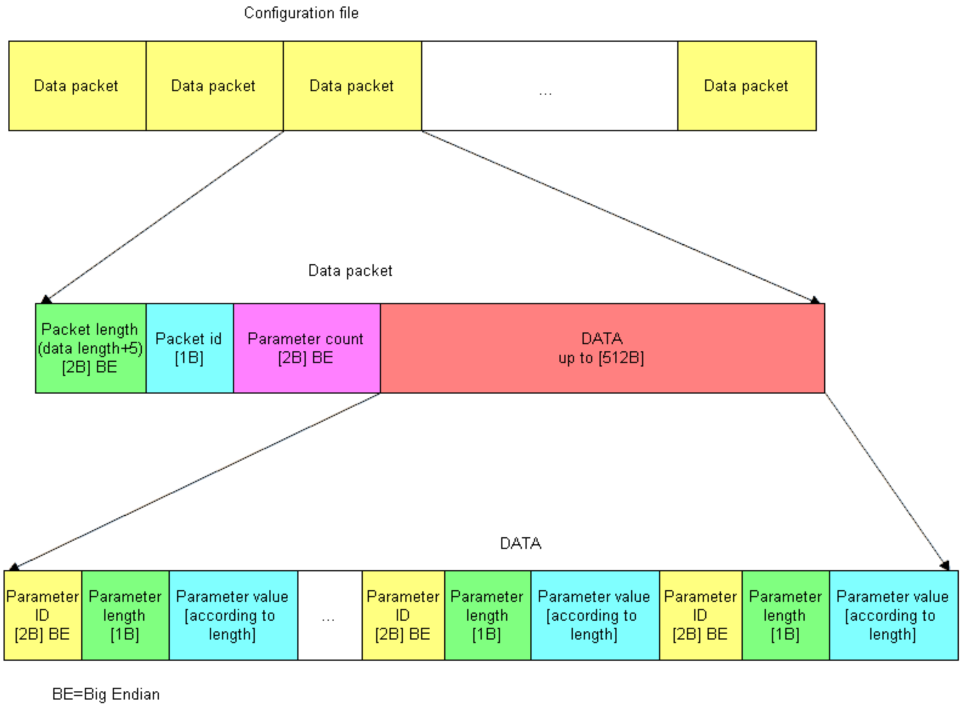
** – not available for pre-revision Trace5 devices.

5 Configuration

There are two possible ways to read/write configuration from/to the tracking device. One option is to use USB connection with PC (personal computer). Another option is to use GPRS service (air). There are specific rules how to download/upload configuration. Configuration file extensions:

- *.ft5c* (HCV5)
- *.fl5c* (LCV5)
- *.fp5c* (Pro5)
- *.fg5c* (HCV5 Lite/Pro5 Lite)
- *.fa5c* (Trace5-LTM (pre-revision))
- *.fb5c* (Trace5 NA)
- *.fd5c* (all other Trace5 variants)
- *.ft4c* (FM-Tco4 HCV)
- *.fl4c* (FM-Tco4 LCV)
- *fp4c* (FM-Pro4)
- *.fe4c* (FM-Eco4)
- *.fk4c* (FM-Eco4 S and FM-Eco4 T)
- *.fo4c* (FM-Plug4)

When a configuration file is created by the configurator, it is divided into data packets:



5.1 Configuration data packet

Configuration data packet has a defined structure:

Configuration packet length [2B] Little Endian	Packet ID [1B]	Parameters count [2B] Little Endian	Configuration data [4-512 B]			
			Parameter ID [2B] Little Endian	Parameter length [1B]	Parameter value [1-128 B]	...

Parameters description:

Parameter	Description
Configuration packet length	The length (in bytes) of the current configuration data packet (with unique packet ID). 2 bytes long.
Packet ID	Unique packet identifier of the current configuration file. 1 byte long.
Parameter count	The number of parameters in configuration data sector. 2 bytes long.
Configuration data	
Parameter ID	Unique parameter identifier. 2 bytes long.
Parameter length	The length of parameter's (defined in *Parameter ID* field) value. 1 byte long.
Parameter value	Actual parameter value. Length is defined in *Parameter length* field.

Usually there are several dozens of parameters (configuration data) in every configuration packet. In order to investigate the configuration file use a simple tool *"configParser.jar"*. It can parse raw configuration file hex string (instructions are in *"explain.txt"* file). Open the configuration file with any hex editor software, save it as a text file *"one.txt"* without spaces and run *"configParser.jar"*.

Example of raw configuration data from a configuration file:

```
E601011F00640001016500200000000000000000000000000000000000000000000000000000000000000000660020000000000000000000000000
0000000000000000000000000000000000000000670020000000000000000000000000000000000000000000000000000000000000000068000100
690001006A0001006E002800000000000000000000000000000000000000000000000000000000000000000000000000000000006F00280000000000
000000000000000000000000000000000000000000000000000000000000000000000078000200007900020000 7A00023C007B000100820010000
00000000000000000000000000000830010000000000000000000000000000000000084001000000000000000000000000000000000008500100000000
00000000000000000000000008600100000000000000000000000000000000087001000000000000000000000000000000000088001000000000000
00000000000000000000089001000000000000000000000000000000000008A00100000000000000000000000000000000008B001000000000000000
0000000000000000008C001000000000000000000000000000000000008D0010000000000000000000000000000000000008E0001008F00010090000400
000000910001009200010193000100FA01025800940028000000000000000000000000000000000000000000000000000000000000000000000000
0000000000...
```

Packet length (1st packet) = 0xE601 (LE) = 0x01E6 (Endian conversion) = 486 Bytes

Packet ID = 0x01 = 1st packet

Parameter count = 0x1F00 (LE) = 0x001F (Endian Conversion) = 31 parameters

Configuration data (1st packet) = 0x6400…0100

Packet length (2nd packet) = 0xFA01 (LE) = 0x01FA (Endian conversion) = 506 Bytes

Packet ID = 0x02 = 2nd packet

Parameter count = 0x5800 (LE) = 0x0058 (Endian Conversion) = 88 parameters

Configuration data (2nd packet) = 0x9400…

Here is the same configuration packet parsed with *"configParser.jar"* (byte order of Little Endian parameters is already swapped for easier interpretation):

```
---------------------------------------
 Packet length: 1e6
 Packet ID: 1
 Parameter count: 1f
 Parameter ID: 64
 Parameter Length: 1
 Parameter value (hex): 01
 ………
 Parameter ID: 92
 Parameter Length: 1
 Parameter value (hex): 01
 Parameter ID: 93
 Parameter Length: 1
 Parameter value (hex): 00
---------------------------------------
 Packet length: 1fa
 Packet ID: 2
 Parameter count: 58
 Parameter ID: 94
 Parameter Length: 28
 Parameter value (hex): 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

5.2 Configuration upload process

Configuration upload process is controlled by 6 commands:

Command (ASCII)	Description
	Host commands (server of PC)
#cfg_start@	Enter into configuration mode (tracking device). Response example from the tracking device with the command *@cfg_sts#<status>,* where *<status>*: *3130* - OK
#cfg_send@*<configuration data packet>*	Send configuration data packet to the tracking device. After *#cfg_send@* command a raw configuration, the data packet should be inserted. Response examples from the tracking device with the command *@cfg_sts#<status>,* where *<status>*: *31xx* – packet *xx* OK; *3031* – packet receiving error. Where *xx* – the packet ID.
#cfg_write@	Start writing the recently sent configuration to the tracking device memory. Response examples from the tracking device with the command *@cfg_sts#<status>,* where *<status>*: *3130* – OK; *3030* – error.
#cfg_end@	Exit configuration mode (tracking device). Response example from the tracking device with the command *@cfg_sts#<status>,* where *<status>*: *3130* – OK.
#cfg_reset@	Clear recently sent configuration data packets. Response example from the tracking device with the command *@cfg_sts#<status>,* where *<status>*: *3130* – OK.

Configuration operations via GPRS are done using commands 2/102 (see paragraph 3.2.3). Configuration upload commands explained above are expressed in ASCII format. They should be converted to HEX format and inserted into the payload field of GPRS commands 2/102. Data in payload field must be terminated with *<0x0D><0x0A>*.

In order to ensure reliable OTA transfer it is strongly recommended to carry out the procedure using TCP. If UDP is used as the primary communication protocol, command 105 can be used to temporarily establish the connection via TCP prior to carrying out the configuration upload process. Refer to section 3.2.27 Command 105 – Set Connection Parameters for the complete description of the command.

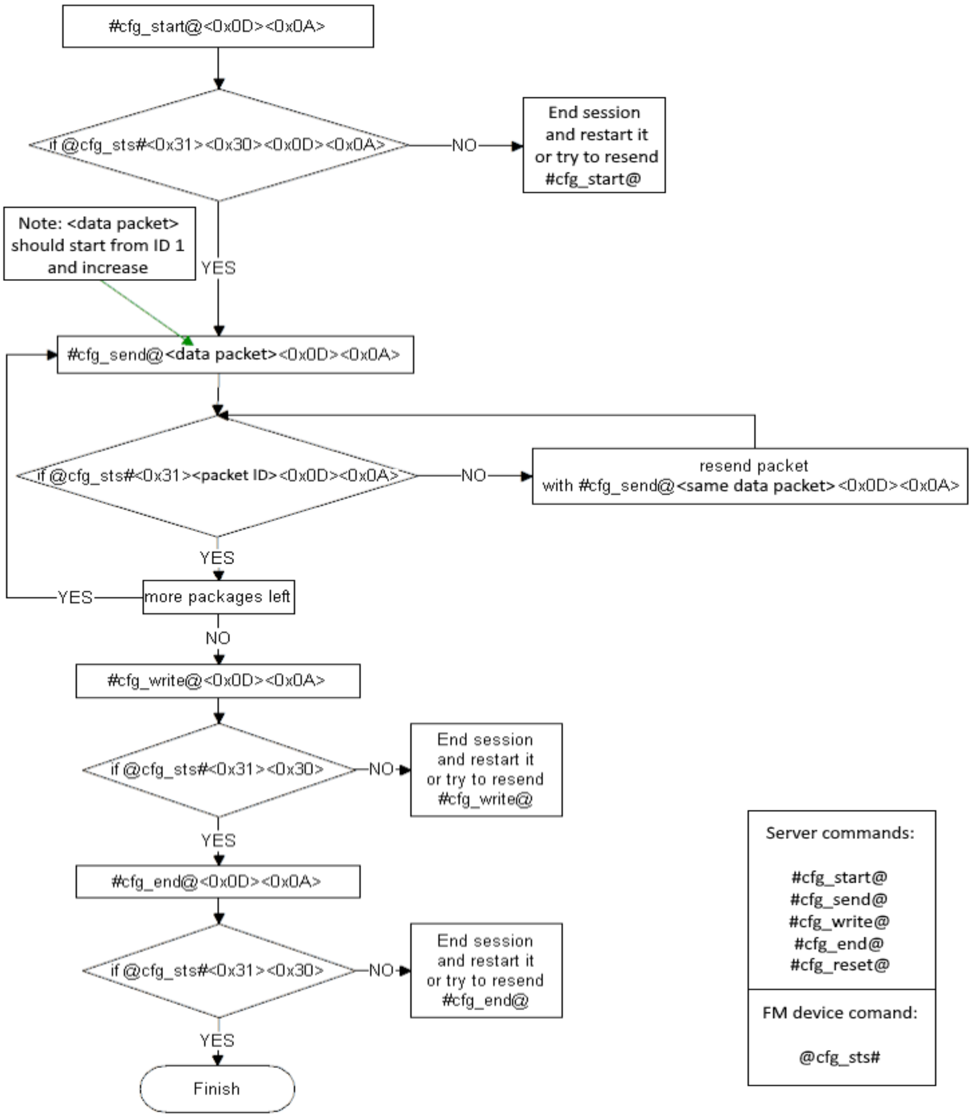
Server uses command 102 (0x66):

Field	Packet length	Command ID	Payload	CRC16
Size (bytes)	2	1	Not fixed	2
Example hex	000E	66	236366675F7374617274400D0A	0947
Example dec	14	102	-	-
Example ASCII	-	-	#cfg_start@	-

FM device uses command 2 (0x02) to send a response:

Field	Packet length	IMEI	Command ID	Payload	CRC16
Size (bytes)	2	8	1	Not fixed	1
Example hex	0016	0003124D0AC0BB1C	02	406366675F7374732331300D0A	4B58
Example dec	22	864547032316700	2	-	-
Example ASCII	-	-	-	@cfg_sts#10	-

Configuration upload flowchart


#cfg_start@<0x0D><0x0A>
if @cfg_sts#<0x31><0x30><0x0D><0x0A>
NO
End session and restart it or try to resend #cfg_start@
YES
Note: <data packet> should start from ID 1 and increase
#cfg_send@<data packet><0x0D><0x0A>
if @cfg_sts#<0x31><packet ID><0x0D><0x0A>
NO
resend packet with #cfg_send@<same data packet><0x0D><0x0A>
YES
YES
more packages left
NO
#cfg_write@<0x0D><0x0A>
if @cfg_sts#<0x31><0x30>
NO
End session and restart it or try to resend #cfg_write@
YES
#cfg_end@<0x0D><0x0A>
if @cfg_sts#<0x31><0x30>
NO
End session and restart it or try to resend #cfg_end@
YES
Finish
Server commands:
#cfg_start@
#cfg_send@
#cfg_write@
#cfg_end@
#cfg_reset@
FM device comand:
@cfg_sts#

1. Start configuration upload process by sending command: *#cfg_start@*.

2. Device should answer with a positive response: *@cfg_sts#<0x31><0x30>.*

3. Send configuration data packet: *#cfg_send@<data packet>**.

4. Device should answer with a positive response with the last received packet number: *@cfg_sts#<0x31><0x01>.*

5. Repeat step 3 until all configuration packets are sent.

6. After all configuration packets are sent to the device server should send a command to initiate writing the configuration to the memory of the FM device: *#cfg_write@*.

7. Device should answer with a positive response: *@cfg_sts#<0x31><0x30>.*

8. After successful write operation server should send exit from configuration mode command: *#cfg_end@*.

9. Device should answer with a positive response: *@cfg_sts#<0x31><0x30>.*

*How to make the *#cfg_send@* data packets:

The first 2 Bytes of the raw configuration file represent the length of the first configuration data packet. They should be interpreted as Little Endians and converted to decimal. This value shows how many Bytes of the raw configuration file should be added to the *#cfg_send@* packet (including the 2 first Bytes). When this packet is sent device should respond with *@cfg_sts#<0x31><0x01>* packet. Now the server should send the next configuration packet. Again the first 2 Bytes of the unsent raw configuration data represent the length of the next configuration data packet. This amount of Bytes again should be added to the *#cfg_send@* packet. When it is sent device should respond with *@cfg_sts#<0x31><0x02>.* These steps should be repeated until all configuration packets are sent to the device. At this point configuration upload procedure should be continued from step 6.

In the example provided in 5.1 paragraph the first 2 Bytes of raw configuration data contain *0xE601* which after conversion to Little Endian and decimal equal 486 Bytes. These bytes should be added to the first *#cfg_send@* packet and sent to the server. The next 2 Bytes of the unsent data contain *0xFA01* which after conversion to Little Endian and DEC equal 506 Bytes. These bytes should be added to the first *#cfg_send@* packet and sent to the server.

Example of uploading configuration file from server to the FM device. Protocol which is described in paragraph 3.2.3 is omitted, only payload data of commands 2/102 is shown.

Remarks:

<> - hex value
Data send format: *#cfg_send@<config_data_packet><0x0D><0x0A>*
Acknowledgement format: *@cfg_sts#<0x31><config_packet_number>*

Source	Command	Hex string	Description
Server	#cfg_start@<0x0D><0x0A>	23 63 66 67 5F 73 74 61 72 74 40 0D 0A	Start configuration mode.
Device	@cfg_sts#<0x31><0x30><0x0D><0x0A>	40 63 66 67 5F 73 74 73 23 **31 30** 0D 0A	Mode changed successfully. Status: **<0x31><0x30>**.
Server	#cfg_send@<data packet><0x0D><0x0A>	23 63 66 67 5F 73 65 6E 64 40 **00 02** 01 24 00 64 00 01 00 65 00 07 76 ... 0D 0A	Send packet 1 data to FM device. Packet size **512** bytes.
Device	@cfg_sts#<0x31><0x01><0x0D><0x0A>	40 63 66 67 5F 73 74 73 23 31 01 0D 0A	Confirm that packet 1 is received.
Server	#cfg_send@<data packet><0x0D><0x0A>	23 63 66 67 5F 73 65 6E 64 40 **00 02** 02 4b 00 97 01 01 00 98 ... 0D 0A	Send packet 2 data to FM device. Packet size **512** bytes.
Device	@cfg_sts#<0x31><0x02><0x0D><0x0A>	40 63 66 67 5F 73 74 73 23 31 02 0D 0A	Confirm that packet 2 is received.
...	...		
Server	#cfg_send@<data packet><0x0D><0x0A>	23 63 66 67 5F 73 65 6E 64 40 **FD 01** 12 48 a8 10 04 00 00 00 00 ... 0D 0A	Send packet 18 data to FM device. Packet size **509** bytes. The last packet was sent.
Device	@cfg_sts#<0x31><0x12><0x0D><0x0A>	40 63 66 67 5F 73 74 73 23 31 12 0D 0A	The last packet (18) was received.
Server	#cfg_write@<0x0D><0x0A>	23 63 66 67 5f 77 72 69 74 65 40 0D 0A	Write configuration to FM device memory.
Device	@cfg_sts#<0x31><0x30><0x0D><0x0A>	40 63 66 67 5F 73 74 73 23 **31 30** 0D 0A	Config was written successfully. Status:**<0x31><0x30>**
Server	#cfg_end@<0x0D><0x0A>	23 63 66 67 5F 65 6E 64 40 0D 0A	Exit configuration mode.
Device	@cfg_sts#<0x31><0x30><0x0D><0x0A>	40 63 66 67 5F 73 74 73 23 **31 30** 0D 0A	Mode changed successfully. Status: **<0x31><0x30>**

Please see a full example of configuraton upload process in “*Server communication examples*” file.

5.3 Configuration download from tracking device process

Configuration download process is quite similar to configuration upload. It is controlled by 5 commands:

Command (ASCII)	Description
Host commands (server of PC)	
#cfg_start@	Enter into configuration mode (tracking device). Response example from the tracking device with the command *@cfg_sts#<status>,* where *<status>*: *3130* - OK
#cfg_get@*<packet ID>*	Get the specified configuration data packet from the tracking device. The tracking devices sends the data packet with following command: @cfg_get#*<data packet>*
#cfg_end@	Exit configuration mode (tracking device). Response example from the tracking device with the command *@cfg_sts#<status>,* where *<status>*: *3130* - OK

In order to ensure reliable OTA transfer it is strongly recommended to carry out the procedure using TCP. If UDP is used as the primary communication protocol, command 105 can be used to temporarily establish the connection via TCP prior to carrying out the configuration download from tracking device process. Refer to section 3.2.27 Command 105 – Set Connection Parameters for the complete description of the command.

Configuration download flowchart

1. Start configuration download process by sending command: *#cfg_start@*.

2. Device should answer with a positive response: *@cfg_sts#<0x31><0x30>.*

3. Request the 1st configuration data packet from the FM device: *#cfg_get@<packet ID>*.

4. Device should answer with the first configuration packet: *@cfg#<data packet>.*

5. Repeat step 3 until the device sends an empty configuration data packet: @cfg#*<0x00><0x00><packet ID>*.

6. After all configuration packets are received the server should send a command to end the configuration download process: *#cfg_end@*.

7. Device should answer with a positive response: *@cfg_sts#<0x31><0x30>.*

Bytes received in *<data packet>* field of *@cfg#* packets should be joined and the configuration file should be made.

Example of uploading configuration file from server to the FM device. Protocol which is described in paragraph 3.2.3 is omitted, only payload data of commands 2/102 is shown.

Remarks:

<> - hex value

Format data request: *#cfg_get@<packet number><0x0D><0x0A>*

Format data send: *@cfg#<config_data_packet>*

Source	Command	Hex string	Description
Server	#cfg_start@<0x0D><0x0A>	23 63 66 67 5F 73 74 61 72 74 40 0D 0A	Start configuration session.
Device	@cfg_sts#<0x31><0x30><0x0D><0x0A>	40 63 66 67 5F 73 74 73 23 **31 30** 0D 0A	Response with status <0x31><0x30>**.**
Server	#cfg_get@<**0x01**><0x0D><0x0A>	23 63 66 67 5F 67 65 74 40 **01** 0D 0A	Try to get packet with number **1.**
Device	@cfg#<data packet>	40 63 66 67 23 **00 02** 01 24 00 64 00 01 00 65 00 07 76 ...	Send packet 1. Packet size **512** bytes.
Server	#cfg_get@<0x02><0x0D><0x0A>	23 63 66 67 5F 67 65 74 40 **02** 0D 0A	Try to get packet with number **2.**
Device	@cfg#<data packet>	40 63 66 67 23 **00 02** 02 4b 00 97 01 01 00 98 ..	Send packet 2. Packet size **512** bytes.
...	...		
Server	#cfg_get@<0x15><0x0D><0x0A>	23 63 66 67 5F 67 65 74 40 **15** 0D 0A	Try to get packet with number **15.**
Device	@cfg#<0x00><0x00><0x15><0x0D><0x0A>	40 63 66 67 23 **00 00** 15 0D 0A	Packet length = **0**, last packet = 15. This means the end of get configuration operation.
Server	#cfg_end@<0x0D><0x0A>	23 63 66 67 5F 65 6E 64 40 0D 0A	End configuration session.
Device	@cfg_sts#<0x31><0x30><0x0D><0x0A>	40 63 66 67 5F 73 74 73 23 **31 30** 0D 0A	Response with status **<0x31><0x30>**.

Please see a full example of configuration download process in “*Server communication examples*” file.

6 Firmware

There are two possible ways to write firmware (FW) to tracking device. One option is to use USB connection with PC (personal computer). Another option is to use GPRS service (air). There are specific rules how to upload new firmware. Firmware file extensions:

- *.efwt5* (HCV5)
- *.efwl5* (LCV5)
- *.efwp5* (Pro5)
- *.efwg5* (HCV5 Lite/Pro5 Lite)
- *.efwa5* (Trace5-LTM (pre-revision))
- *.efwb5* (Trace5 NA)
- *.efwd5* (all other Trace5 variants)
- *.efwt4* (FM-Tco4 HCV)
- *.efwl4* (FM-Tco4 LCV)
- *.efwp4* (FM-Pro4)
- *.efwe4* (FM-Eco4/4+; FM-Eco4 light/light+)
- *.efwk4* (FM-Eco4 S and FM-Eco4 T)
- *.efwo4* (FM-Plug4)

General firmware file pack for all of the tracking devices: **ReleasePack_XXXX-XX-XX.fwp**. Where **XXXX-XX-XX** is the release of the firmware update.

You cannot write different type of firmware to a specific device, for ex.: HCV5 firmware to a Pro5 device. Downgrade for some older versions of firmware is prohibited. Usage of newest firmware is advised. For FW update via GPRS commands 4/104 (see 3.2.5) are used.

6.1 Firmware .fwp packet

The combined firmware update .fwp packet contains the following files:

- Manifest file (describes contents of the .fwp pack)
- Firmware files for all devices
- nRF module APP, Bootloader and SoftDevice files.

6.2 Firmware data packet

Firmware file should be sent to FM device in data packets. At first you should divide FW file (raw data) into pieces of maximum 512 bytes. After this FU (Firmware Update) packets can be formed.

FU packet:

FU packet			
Whole FU packet length [2B] Little Endian.	**FU packet ID [2B]** Little Endian. Starts from no. 1.	**FU raw data [max 512B]**. From *.efw* file.	**FU data CRC [2B]** Little Endian. Only from FU raw data.

Parameters description:

Parameter	Description
Whole FU packet length	Length of the current FU packet size. Whole FU packet length = length + ID + raw FW data + CRC = 6 + raw FW data. Format: Little Endian. Parameter size is 2 bytes.
FU packet ID	Unique FU packet ID. This parameter identifies packets. Format: Little Endian. Parameter size is 2 bytes.
FU raw data	Raw data of part of firmware file. Max size is 512 bytes.
FU data CRC	CRC16 kermit calculation. It is calculated only from FU raw data. Format: Little Endian. Parameter size is 2 bytes.

Example of FU data packet (raw data 518 bytes):

06020100D7323566071624373301306FE0656754220E073311333801515F5140687A4040784E00647200245E79370175666772457713003501777A
01D5507803712D522A422F357673352637E1FF306F941A645446AB0733D1E338012D885140D5CE404067FB00641163245E04DA01756066724549
13003562147A0143517803532D522A042A3576291624370101306F4AF36754A8730433518F380169A451400B19404081F00264A5D3245E7E59017
560667245491300358C187A0143517803532D522A042A3576291624370101306F6DF065AC779D81CB4CF0A8E90830D304EB3EEAB17949DA2173
D124AE02CFAE870A0BC8AD257013C5007862BEDB283BF3314D4A6D9BCD367676F0273768267F9F6D0C77AC765614C3415430BE18C4500B756B4
8FF68B6013F161E218E69CF001E671B73BD2B1BF9E41287720E3E8D68FB3105FF366B523CA7276B7C88639A31A497D562B46395065847A8396A6
522A895F9389693080900647223247A79120153133F5AFA52B1FBED537052BE10F330BC3B2E226D01600286E4F5275F796818D02CF0E755134435
C3BDD54D283ACCA9C5105372B013C8E44D40F008D81A1E3285FB86016C1980FDB601570A463F874EF3EFB161DA00B531764A55241657923A902
D1C0734774D071273A0D6FFE85C4EFD78CF50FD4BBE7C996346DBA99EC80185BDFC41B5808D0381FE8088FE238E4BF39BB3522A1985

Whole FU packet length = 0x0602 (raw) = 0x0206 (Endian conversation) = 518 bytes.

FU packet ID = 0x0100 (raw) = 0x0001 (Endian conversation) = 1

FU data CRC = 0x1985 (raw) = 0x8519 (Endian conversion) = 34073

6.3 Firmware upload process

Special commands are used for the FW update. All commands are terminated with *<0x0D> <0x0A>* symbols.

Command	Description
Host commands (server of PC)	
|FU_STRT*	Enter into the firmware update mode (tracking device).
|FU_PCK*<FU_packet>	Send firmware data packet.
|FU_WRITE*	Start writing firmware (recently sent) to the tracking device memory.
|FU_END*	Terminate firmware update. Only used to quit FW update process.
FM device commands	
*FU_OK|	Acknowledgment from the tracking device.
*FU_EER|	Error from the tracking device. Wrong CRC result is shown if necessary (last 2 bytes).

Note

Different firmwares can be updated via the FW update command. For each different FW, only the |FU_WRITE* command differs. If the combined firmware file is sent, the |FU_STRT* command will also be different. List of possible FW updates are provided in the table below:

Command	Description
|FU_STRT_M*	Enter into the combined firmware update mode.
|FU_WRITE_G*	FW write command for MC60 modem GNSS FW (FM-Eco4 S only).
|FU_WRITE_BA*	FW write command for BLE APP, BLE SoftDevice, BLE Bootloader. (valid for HCV5, LCV5, Pro5, except for versions without Bluetooth)
|FU_WRITE_M*	FW write command for modem FW. (valid for HCV5, LCV5, Pro5, FM-Tco4 HCV, FM-Tco4 LCV, FM-Pro4)
|FU_WRITE_S1*	FW write command for modem SSL server 1 certificate. (valid for HCV5, LCV5, Pro5, Trace5, FM-Tco4 HCV, FM-Tco4 LCV, FM-Pro4, FM-Eco4 T)
|FU_WRITE_S2*	FW write command for modem SSL server 2 certificate. (valid for HCV5, LCV5, Pro5, Trace5, FM-Tco4 HCV, FM-Tco4 LCV, FM-Pro4, FM-Eco4 T)
|FU_WRITE_PK*	FW write command for modem SSL private key. (valid for HCV5, LCV5, Pro5, Trace5, FM-Tco4 HCV, FM-Tco4 LCV, FM-Pro4, FM-Eco4 T)
|FU_WRITE_FB*	FW write command for the combined firmware pack. (valid for HCV5, LCV5, Pro5)
|FU_WRITE_EXT*	FW write command for the extender firmware (valid for HCV5 Lite/Pro5 Lite)

In order to ensure reliable OTA transfer it is strongly recommended to carry out the procedure using TCP. If UDP is used as the primary communication protocol, command 105 can be used to temporarily establish the connection via TCP prior to carrying out the firmware upload process. Refer to section 3.2.27 Command 105 – Set Connection Parameters for the complete description of the command.

Firmware update flowchart

1. Enter to FW update mode by sending command: *|FU_STRT**.

2. Device should answer with a positive response: **FU_OK|*.

3. Send a FW packet to the FM device: *|FU_PCK*<FU_packet>**.

4. Device should answer with a positive response: **FU_OK|<FU packet ID>.*

5. Repeat step 3 until all FU packets are sent.

6. If **FU_OK|* isn't received check for the GPRS link availability and if the link is available repeat the last packet, if not available – reconnect and start from beginning (*|FU_STRT**).

7. After all FW packets are sent to the device server should send a command to initiate writing the FW to the memory of the FM device: *|FU_WRITE**.

8. Device should answer with a positive response: **FU_OK|*. After that FM device automatically ends FW update

mode and restarts.

*How to make the *|FU_PCK** data packets:

Raw FW file should be divided to 512 Byte parts, the last packet would be of shorter length. First, length of the whole FU packet should be calculated: FU packet length = length + packet ID + raw FW data + CRC = 6 Bytes + raw FW data. If it is not the last packet, length of the FU packet is 518 Bytes = 0x0602 (Little Endian). Then CRC of the raw FW part that will be sent should be calculated. Now the FU packet can be formed: length (2 Bytes, Little Endian) + packet ID (2 Bytes, Little Endian) + raw FW data (up to 512 Bytes) + CRC (2 Bytes, Little Endian).

Example of the whole server command 104 (see 3.2.5) packet is shown below:

```
0211687c46555f50434b2a06020100d09f03008fb335565748273f475f33674a3d645c4e50043b7d6d3b0935015248687a4040784e00647200245e79
3701754266724d6b4d033d01777a01f151780b757351223c74367e1148273f395f3367363d645c2c50043b1d6d3b09458251483347434823730362c2
93d2756220a027d5838714ddfd2003d34d47a097b0f7b0b6b7351223c74367e1148273f395f3367363d645c2c50043b1d6d3b0953015248cde540482
373036c293d2756220a027d5838714d714d033d5a4a79092d92780b6b7351223c74367e1148273f395f3367363d645c2c50043b8ffb380953015248b
dd14048b5d1006c293d2756220a027d030572452a70003501777a0120327803304e522a674935764a752437dfc73067363d645c96c6073b1d6d3b09
f99f514815df40482373036c293d2756220a027d5838714d714d033d5a4a79097b0f7b0b6b7351223c74367e1148273f395f336768c0675c779d05cb4
6a08af9029cc1a86876c204fb0aaa957307fe1b78e60185a4fdddb7237ebadd0e7769f1213d60bccb5411da664a2d3122ee2737eaf9336f674428a4776
117cb476b2bf10f3859ff7882410b655f08db62f825051d2904a513fd732e4e6e01cd001c83d033c2700c2ef142d26602986a466e2de60f7c68d06cf866
9f8db802d352a8396a09c4502b0564b995e90cd6b70247245e79140151032072633a4a288a79b681d9723550bcb6310d0a4d7d
```

Packet length = 0x0211 = 529 bytes (itself and CRC16 field are excluded)

Server protocol Command ID = 0x68 = 104

Firmware packet send command = 7c46555f50434b2a = |FU_PCK*

FU packet length = 0x0602 (Little Endian) = 0x0206 (Endian conversion) = 518 bytes

FU packet ID = 0x0100 (Little Endian) = 0x0001 (Endian conversion) = 1st packet

Firmware raw data 512 Bytes

FU packet CRC = 0xb631 (Little Endian) = 0x31b6 (Endian conversion) = 12726

Symbols to terminate payload packet - 0d0a

CRC16 Kermit answer = 0x4d7d = 19837

Firmware packet for Trace-2G, Trace5-LTE and Trace5-LTM (Rev.A)

For Trace5-2G, Trace5-LTE and Trace5-LTM (Rev.A) devices the FW file has the following structure:

<FW for Partition 0>

<FW for Partition 1>

<Metadata> – 16B

In order to perform the FW update, there are two options:

1. Upload the complete file as per the standard sequence;
2. Only upload the FW for a specific partition.

With the second option, the amount of data that needs to be sent to the device in order to perform the FW update is halved. To identify which of the two partitions must be used, a special parameter for the 'info' SMS command ('fpart') is introduced (see corresponding section). If the command response is "0", FW for partition 1 must be uploaded; if the response is "1", FW for partition 0 must be uploaded.

The FW size for Partition 0 and Partition 1 is completely identical. In order to obtain the byte array for the FW update, omit the last 16 Bytes of the original FW file byte array and divide the remaining byte array in half. The first half will be the FW for Partition 0, the second – FW for Partition 1.

Example of uploading FW file from server to the FM device. Protocol which is described in paragraph 3.2.5 is omitted, only payload data of commands 4/104 is shown:

Source	Command	Hex string	Description
Server	|FU_STRT*<0x0D><0x0A>	7C 46 55 5F 53 54 52 54 2A 0D 0A	Enter FW update mode.
Device	*FU_OK|<0x0D><0x0A>	2A 46 55 5F 4F 4B 7C 0D 0A	Mode changed successfully.
Server	|FU_PCK*<FW data packet><0x0D><0x0A>	7C 46 55 5F 50 43 4B 2A **06 02** 01 00 D7 … 19 85 0D 0A	Send FW data packet 1 to FM device. Packet size **518** bytes.
Device	*FU_OK|<0x0D><0x0A>	2A 46 55 5F 4F 4B 7C 01 00 0D 0A	Confirm that packet 1 is received.
Server	|FU_PCK*<FW data packet><0x0D><0x0A>	7C 46 55 5F 50 43 4B 2A **06 02** 02 00 E7 … 2E FF 0D 0A	Send FW data packet 2 to FM device. Packet size **518** bytes.
Device	*FU_OK|<0x0D><0x0A>	2A 46 55 5F 4F 4B 7C 02 00 0D 0A	Confirm that packet 2 is received.
…	…		
Server	|FU_PCK*<FW data packet><0x0D><0x0A>	7C 46 55 5F 50 43 4B 2A **06 02** C5 01 E7 … 69 74 0D 0A	Send packet 453 data to FM device. Packet size **518** bytes. The last packet was sent.
Device	*FU_OK|<0x0D><0x0A>	2A 46 55 5F 4F 4B 7C C5 01 0D 0A	The last packet (453) was received.
Server	|FU_WRITE*<0x0D><0x0A>	7C 46 55 5F 57 52 49 54 45 2A 0D 0A	Write firmware to FM device memory.
Device	*FU_OK|<0x0D><0x0A>	2A 46 55 5F 4F 4B 7C 0D 0A	Firmware was updated successfully.

7 Abbreviations

<0x00> – value in hex format

<0x0D> – (CR) Carriage Return symbol

<0x0A> – (LF) Line Feed symbol. Also known as New Line (NL) / End of Line (EOL)

B – bytes/bytes

BT - Bluetooth

kB – kilobyte

CFG – Configuration

Config – Configuration

FW – firmware

FU – Firmware Update