INTERSTATE COUNCIL
ON STANDARDIZATION, METROLOGY AND CERTIFICATION
(ISC)

INTERSTATE STANDARD

GOST 33465-2015

Global navigation satellite system

ROAD ACCIDENT EMERGENCY RESPONSE SYSTEM

Protocol of data exchange between in-vehicle emergency call device/system and emergency response system infrastructure

Official Edition
English Version Approved by Interstandard

Moscow
Standartinform
2017

Foreword

The purposes, main principles and basic order of work on interstate standardization are established by GOST 1.0-2015 "Interstate system for standardization. Basic principles" and GOST 1.2-2015 "Interstate System for Standardization. Interstate standards. Rules for development, taking over, renovation and cancellation"

Details

1 DEVELOPED by Non-Commercial Partnership "For Promotion of Navigation Technologies Development and Application" and Joint Stock Company "Research and Technical Centre of Advanced Navigation Technologies" "Internavigation" (JSC "Internavigation RTC")

2 INTRODUCED by Federal Agency on Technical Regulating and Metrology

3 ADOPTED by Interstate council for standardization, metrology and certification by means of correspondence (protocol No. 82-П, dated 12.11.2015)

Votes in favour:

Short name of the country according to IC (ISO 3166) 004—97	Country code according to IC (ISO 3166) 004—97	Abbreviated name of national standards body
Armenia	AM	Ministry of Economics of Republic of Armenia
Belarus	BY	Gosstandart of Republic of Belarus
Kyrgyzstan	KG	Kyrgyzstandart
Russian Federation	RU	Rosstandart
Tajikistan	TJ	Tajikstandart

4 Interstate Standard GOST 33465-2015 is introduced as a national standard of the Russian Federation by Order No. 2035-ст, dated 15.12.2016, of Federal Agency on Technical Regulating and Metrology from 01.01.2017.

5 This Standard developed on based GOST R 54619-2011*

6 INTRODUCED FOR THE FIRST TIME

The information on the amendments to this Standard is published in the annually issued information index "National standards", and the text of the amendments and corrections is published in the monthly issued information indices "National standards". In case of revision (replacement) or cancellation of this Standard the appropriate notice will be published in the monthly issued information index "National standards". The appropriate information, notice and texts are also placed in the general-use information system — on official site of Federal Agency on Technical Regulating and Metrology in the Internet (www.gost.ru)

National standard GOST R 54619-2011 withdrawn from 01.06.2017 by Order No. 2035-ст, dated 15.12.2016, of Federal Agency on Technical Regulating.

Contents

Introduction

This Standard belongs to the set of standards entitled "Global Navigation Satellite System. Road Accident Emergency Response System," and is one of the base standards included in this set.

The Road Accident Emergency Response System is meant to mitigate the consequences of road accidents and other emergencies on the roads by reducing the time required to report such accidents to emergency services. This System is called "ERA-RB" in the Republic of Belarus, "EVAK" in the Republic of Kazakhstan and "ERA-GLONASS" in the Russian Federation. The System is analogous to the European eCall System currently under development, and is harmonised with it in regard to its main functional features (the use of in-band modem as the main data transmission tool; unified content and format of mandatory data transmitted in the minimum set of data pertaining to road accidents; uniform procedures for initiation and termination of duplex voice connection with the persons in the vehicle cabin, etc.).

This Standard describes the protocol used for data exchange between the in-vehicle emergency call system/device and the System Operator's infrastructure of the Road Accident Emergency Response System, and its related protocol used for support of services including the Base Service of this System.

The Standard provides all necessary information on the format and procedures of message exchange, and shall be used for development of data transmission subsystems intended for operation on the side of in-vehicle emergency call systems/devices as well as on the System Operator's side.

The basic provisions of this Standard are interrelated with the key standards from the set "Global Navigation Satellite System. Road Accident Emergency Response System."

This Standard takes into account the basic provisions of the relevant regional (European) and International standards as well as of other world-wide standardisation documents, in regard to:

- networking model of Open Systems Interconnection: in part of the protocols used at transport and network layers for data transfer between an in-vehicle emergency call device/system and System;
- European eCall Safety System: in part of the minimum set of data transmitted by an in-vehicle emergency call device/system;
- wireless mobile (cellular) communication: in part of SMS data transmission.

This Standard is intended for application by:

- manufacturers of in-vehicle emergency call systems/devices;
- manufacturers of motor vehicles;
- Operator of the Road Accident Emergency Response System;
- developers and vendors of services based on the navigation IT platform of the Road Accident Emergency Response System.

INTERSTATE STANDARD

Global navigation satellite system

ROAD ACCIDENT EMERGENCY RESPONSE SYSTEM

Protocol of data exchange between in-vehicle emergency call device/system and emergency response system infrastructure

Date of Introduction — 2017—01—01

1 Scope

This Standard applies to in-vehicle emergency call devices/systems intended for installation on wheeled vehicles of Categories M and N in accordance with the requirement of the Regulation [1].

The Standard sets out the requirements for protocols of data exchange between an in-vehicle emergency call device/system and the infrastructure of the Road Accident Emergency Response System (hereinafter referred to as the "System") including the exchange of those data that are related to the provision of the Base Service in accordance with the requirements of the Regulation [1] and GOST 33464.

2 Normative references

The following standards are referred to in this Standard:

GOST 33464-2015 Global navigation satellite system. Road accident emergency response system. In-vehicle emergency call device/system. General technical requirements

Note — When using this standard it is expedient to check the validation of the reference standards in the general-use information system — on official site of Federal Agency on Technical regulating and Metrology in Internet or according to the annual information index "National standards" which is published as of January, 1st, of current year, and according to releases of monthly issued information index "National standards" in the current year. If a reference standard which the dated reference is provided to is replaced, it is recommended to use a version of this standard with the above specified year of approval (acceptance). If after the approval of this standard an amendment is inserted in a reference standard which the dated reference is provided to, and this amendment regards the provision referred to, it is recommended to apply this provision without regard to this amendment. If a reference standard is cancelled without a replacement, it is recommended to apply the provision which refers to it to a part which does not engage this reference.

3 Terms, definitions and abbreviations

3.1 Terms and definitions

The terms defined in GOST 33464 as well as the following terms with their respective definitions are used for the purposes of this Standard:

3.1.1 **minimum set of data;** MSD: Set of data transmitted by the in-vehicle emergency call system/device in the case of a road traffic accident, including: location and movement parameters of the affected vehicle, accident time, vehicle VIN-code and other information necessary for emergency response.

Official Edition

3.1.2 **operator of Road Accident Emergency Response System (System Operator):** Legal entity involved in activities for System operation, in particular, processing of data stored in the System database.

3.1.3 **data transfer protocol:** Collection of rules and conventions governing the contents, format, timing, succession and error checks of messages transmitted between network devices.

3.1.4 **service:** Infrastructure element included in the telematic platform of the Road Accident Emergency Response System in order to complete functions of an algorithm behind a particular kind of servicing provided by the System with the help of the service support layer protocol.

3.1.5 **Road Accident Emergency Response System:** Automated geographically distributed Federal and State Information System that uses the signals of the GLONASS Global Navigation Satellite System and of other active GNSS to provide for prompt collection of data related to road accidents or other emergencies on motor roads as well as for processing, storage and transmission of such data to emergency services, and to enable access to the said data for the concerned governmental or local authorities, officials, legal and natural persons.

Note — The Road Accident Emergency Response System is called "ERA-RB" in the Republic of Belarus, "EVAK" in the Republic of Kazakhstan, and "ERA-GLONASS" in the Russian Federation. These systems are analogous to the European eCall System currently in development, and are harmonised with it in regard to the main functional features (the use of in-band modem as the main data transmission tool, unified content and format of mandatory data transmitted in the MSD for road accidents, uniform procedures for initiation and termination of duplex voice connection with the persons in the vehicle cabin, etc.).

3.1.6 **in-vehicle emergency call system;** IVS: System supporting the functions of an in-vehicle emergency call device and providing for automatic transmission of vehicle data messages when a road accident or an accident of other kind occurs.

Notes

1 In addition, an in-vehicle emergency call system may be used for manual transmission of vehicle data messages in the case of road accidents or accidents of other type.

2 Categories of vehicles that shall be equipped with in-vehicle emergency call systems are specified in [1].

3.1.7 **in-vehicle emergency call device;** IVD: Device used for measurement and evaluation of vehicle coordinates, speed and direction of movement based on the signals from at least two active Global Navigation Satellite Systems, for manual transmission of vehicle data messages when a road accident or an accident of other kind occurs, and for duplex voice communication with emergency services over wireless mobile communication networks.

Notes

1 In addition, an in-vehicle emergency call device may be used for automatic transmission of vehicle data messages in the case of road accidents or accidents of other type. The types of road accidents detected automatically and the time frames for implementation of the function for automatic transmission of vehicle data messages in the device are established in [1].

2 Categories of vehicles that shall be equipped with in-vehicle emergency call devices are specified in [1].

3.2 Abbreviations

The following abbreviations are used for the purpose of this Standard:

CP-1251	— Code page CP1251 (standard character set and 8-bit encoding used for all Russian versions of Microsoft Windows);
CRC-8(16)	— Cyclic Redundancy Code;
Digital	— Information in electronic form used to identify the sender of data;
DNS	— Domain Name System;
eCall	— Emergency Call (European emergency response system);
EGTS	— Telematic Standard of Road Accident Emergency Response System;
ERA	— Emergency Response to Accident;

FTP — File Transfer Protocol;
GSM — Global System for Mobile communications (global digital standard for cellular mobile communications);
HTTP — HyperText Transfer Protocol;
IMAP — Internet Message Access Protocol (application layer protocol for e-mail access);
IP — Internet Protocol;
ISDN — Integrated Services Digital Network;
IVDS — In-Vehicle Emergency Call Device/System;
Little-endian— Low byte first (byte order);
NGTP — Next Generation Telematics Protocol (next generation protocol defining new connectivity architecture and design concept);
NIS — Navigation and Information Systems;
OSI — Basic reference model of Open Systems Interconnection (abstract networking model used for communication and for development of network protocols);
PDU — Protocol Description Unit);
POP3 — Post Office Protocol Version 3;
RAM — Random Access Memory;
RAM — Random Access Memory;
SC — Service Centre (centre responsible for SMS message processing, storage and transmission to recipients);
signature
SIM — Subscriber Identification Module;
SME — Short Message Entity (entity capable of sending and receiving SMS messages);
SMS — Short Message Service;
SMSC — Short Message Service Centre;
SMTP — Simple Mail Transfer Protocol;
SSLP — Service Support Layer protocol;
SSLP — Service Support Layer protocol;
SW — Software;
TCP — Transmission Control Protocol;
telnet — TErminaL NETwork (network protocol enabling implementation of the network interface for text transmission);
TFTP — Trivial File Transfer Protocol;
TLP — Transport Layer Protocol;
TLP — Transport Layer Protocol;
TP — Telematic Platform;
TP — Telematic Platform;
UDP — User Datagram Protocol;
VH — Vehicle.

4 General

4.1 According to [2], the Open System Interconnection networking model defines the following data exchange layers:

- physical layer;
- data link layer;
- network layer;
- transport layer;
- session layer;
- data presentation and application layers.

4.2 In terms this model, the following protocols are used in the Road Accident Emergency Response System for data exchange between IVDS and System Operator:

- TCP — transport layer;
- IP — network layer.

The correspondence between the OSI networking model, TCP/IP protocol stack and data transfer protocols used in the Road Accident Emergency Response System is detailed in Table 1.

Table 1 — Correspondence between OSI model layers, TCP/IP protocol stack and System protocols

OSI Model		TCP/IP protocol stack		TCP/IP protocols	System protocols
Layer No.	Layer name	Layer No.	Layer name		
7	Application	4	Application	FTP, HTTP, POP3, IMAP, telnet, SMTP, DNS, TFTP	Service Support Layer
6	Presentation				
5	Session				Transport Layer
4	Transport	3	Transport	TCP, UDP	TCP
3	Network	2	Inter-network	IP	IP
2	Data link	1	Network access	—	—
1	Physical				—

4.3 This Standard establishes the requirements applicable to the following data exchange protocols used for communication between the components of the Road Accident Emergency Response System:

- Transport Layer Protocol;
- Service Support Layer Protocol for services including the Base Service of the Road Accident Emergency Response System.

4.4 In addition, this Standard establishes the requirements for the format of AL-ACK messages sent using in-band modems [3].

5 Transport Layer Protocol

5.1 Purpose of Transport Layer Protocol

5.1.1 The Transport Layer Protocol is intended for routing of Service Support Layer Protocol data between System infrastructure points and IVDS using this Protocol, for checks of data integrity and proper data transmission order, and for reliable data transfer to the destination point.

5.1.2 The system design principle based on the Transport Layer Protocol is described in Appendix A.

5.1.3 The analysis of the Transport Layer Protocol in terms of the NGTP concept is presented in Appendix B.

5.2 Routing requirements

The Transport Layer Protocol rests upon the principle of flexible packet routing between interrelated elements of the distributed network where telematic platforms that use this protocol reside. The telematic platform identifiers, which shall be unique within a single integrated network, are used as routing destination addresses.

5.3 Data integrity check mechanism

The integrity of the transmitted data is checked using checksums of Transport Layer headers and of Service Support Layer data. The recipient calculates the checksums and compares them with the respective values recorded in dedicated packet fields by the sending party. If the checksums do not match, the integrity is considered to be broken, and an error code is sent in the response describing processing results.

In order to minimise the utilisation of system resources during packet processing, the integrity control fields and algorithms used at the Transport Layer do not coincide with those used at the Service Support Layer. However, the mechanism is in both cases based on calculation of the checksum (Cyclic Redundancy Check code) for a byte sequence being transmitted.

For Transport Layer packet parts, the CRC-8 algorithm is used for calculation of CRC codes.

For Service Support Layer packet parts, the CRC-16 algorithm is used for calculation of CRC codes.

5.4 Ensuring reliable delivery of data packets

5.4.1 The mechanism used to ensure reliable delivery is based on the acknowledgement of packets sent before. After a packet is transmitted, the sender waits for its acknowledgement in the form of a packet of the specific type that includes the ID of the original packet and the result of its processing by the recipient. Such waiting is limited to a time interval that is governed by the Transport Layer Protocol and depends on the type of the lower level transport protocol being used (the TL_RESPONSE_TO parameter, see 5.8). Once the acknowledgement is received, the sender analyses the result code.

The result processing codes are also regulated by the Transport Layer Protocol; they are presented in Appendix C.

5.4.2 Depending on the results of this analysis, a packet is considered as either delivered or not. The packet is also considered undelivered if no acknowledgement is received before the TL_RESPONSE_TO time interval (see 5.8) expires. Any undelivered packets are sent again (the number of delivery attempts is controlled by this protocol and defined by TL_RESEND_ATTEMPTS specified in 5.8). After the maximum number of delivery attempts is reached, the data link is considered unreliable, the established session is closed (link is disconnected when TCP/IP is used as a transport protocol), and an attempt to create a new session (connection) is made after the TL_RECONNECT_TO time interval (see 5.8) expires.

5.5 Description of data types used in Transport Layer Protocol

5.5.1 The Transport Layer Protocol defines and employs several data types for fields and parameters. The content and description of data types used in the Transport Layer Protocol are presented in Table 2.

5.5.2 The USHORT, UINT, ULONG, FLOAT and DOUBLE multi-byte data types conform to the Little-endian byte order (low byte first). Bytes that make up STRING and BINARY data type sequences are interpreted as is, i.e., in order of their appearance.

5.5.3 The fields and parameters defined in the Transport Layer Protocol may be either:

- M (mandatory) — must always be transmitted, or
- O (optional) — not necessarily transmitted: presence of this field or parameter is governed by other parameters included in the packet.

Table 2 — Content and description of data types used in Transport Layer Protocol

Data type	Size in bytes	Range of values	Description
BOOLEAN	1	TRUE-1, FALSE-0	Logical data type that takes two values only, TRUE or FALSE
BYTE	1	0 ... 255	Unsigned integer
USHORT	2	0 ... 65535	Unsigned integer
UINT	4	0 ... 4294967295	Unsigned integer
ULONG	8	0 ... 18446744073709551615	Unsigned integer
SHORT	2	-32768 ... +32767	Signed integer

Table 2 (continued)

Data type	Size in bytes	Range of values	Description
INT	4	-2147483648 ... +2147483647	Signed integer
FLOAT	4	±1,2E — 38 ... 3,4E + 38	Signed fraction as per [4]
DOUBLE	8	±2,2E — 308 ... 1,7E + 308	Signed fraction as per [4]
STRING	Variable. Size defined by extra parameters or by special terminator (code 0x00).	—	Sequence of printable characters in CP-1251 encoding by default, unless a different encoding is explicitly specified (using an extra parameter)
BINARY	Variable. Size defined by extra parameters.	—	Sequence of elements of BYTE data type
ARRAYOFTYPE	Variable. Size defined by extra parameters.	—	Sequence of elements of any TYPE defined above other than BINARY. Byte order and size of each element is determined by the data type itself. Data type instances follow each other in sequence. E.g., ARRAY OF STRING may contain 10 instances of STRING data type where the size of each instance is determined by the delimiter (code 0x00) that must be present between those instances.

5.6 Description of data structures used in Transport Layer Protocol

5.6.1 The general structure of a Transport Layer Protocol packet is determined by packet contents and format.

5.6.1.1 Each packet used in the Transport Layer Protocol includes the header, Service Support Layer data fields, and a field containing the checksum of Service Support Layer data.

The packet structure defined in the Transport Layer Protocol is shown in Figure 1.

Transport Layer Protocol header	Service Support Layer data	Checksum of Service Support Layer data

Figure 1 — Structure of Transport Layer Protocol packets

5.6.1.2 The total length of a Transport Layer Protocol packet does not exceed 65535 bytes; the latter value corresponds to the maximum value of the Window Size parameter (maximum size of entire packet accepted by the recipient) of the TCP header. Such maximum packet size provides for a more efficient utilisation of data transmission links without the use of data control methods other than the standard one defined by the TCP/IP protocol [3].

The packet format used in the Transport Layer Protocol is described in Table 3.

Table 3 — Format of Transport Layer Protocol packets

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Type	Data type	Size in bytes
PRV (Protocol Version)								M	BYTE	1
SKID (Security Key ID)								M	BYTE	1
PRF (Prefix)		RTE	ENA		CMP	PR		M	BYTE	1
HL (Header Length)								M	BYTE	1
HE (Header Encoding)								M	BYTE	1
FDL (Frame Data Length)								M	USHORT	2
PID (Packet Identifier)								M	USHORT	2
PT (Packet Type)								M	BYTE	1
PRA (Peer Address)								O	USHORT	2
RCA (Recipient Address)								O	USHORT	2
TTL (Time To Live)								O	BYTE	1
HCS (Header Check Sum)								M	BYTE	1
SFRD (Services Frame Data)								O	BINARY	0 ... 65517
SFRCS (Services Frame Data Check Sum)								O	USHORT	0, 2

5.6.1.3 The Transport Layer Protocol header consists of the following parameters (fields): PRV, PRF, PR, CMP, ENA, RTE, HL, HE, FDL, PID, PT, PRA, RCA, TTL, and HCS. The Service Support Layer Protocol is accounted for in the SFRD field, and the Service Support Layer Protocol checksum is recorded in the SFRCS field.

The description of the above parameters (fields) is given in Table 4.

Table 4 — Description of parameters (fields) included in Transport Layer Protocol packets

Parameter (field) name	Parameter (field) name purpose
PRV	Header structure version. Must contain the value 0x01, to be incremented each time the header structure is modified.
SKID	Key ID used for encryption.
PRF	Prefix of the Transport Layer header; must be equal to 00 in the current version.
RTE (Route)	Bit field that defines whether this packet should be routed further to a remote telematic platform, and whether the optional PRA, RCA and TTL parameters are required for its routing. If equal to 1, routing is required, and PRA, RCA and TTL must be present. This field is set by the dispatcher of a telematic platform that has generated this packet, or by the IVDS that has generated a packet for that platform (in case that the HOME_DISPATCHER_ID parameter identifies the platform address where the IVDS is registered). If there is no HOME_DISPATCHER_ID parameter in the IVDS, the packet is routed based on the internal rules of the dispatcher processing the packet.
ENA (Encryption Algorithm)	Bit field that defines the algorithm code used for data encryption of the SFRD field. If set to 00, the SFRD field is not encrypted. The list of algorithms and their codes are not specified in this version of the Protocol.

Table 4 (continued)

Parameter (field) name	Parameter (field) name purpose
CMP (Compressed)	Bit field that defines if the SFRD field is compressed. When set to 1, the data in the SFRD field are considered compressed. The compression algorithm is beyond the scope of this version of the Protocol.

PR (Priority)	Bit field that defines the routing priority for this packet. It may take the following values:
	00 — top priority;
	01 — high priority;
	10 — medium priority;
	11 — low priority.
	Setting a higher priority allows transmitting packets with urgent data, e.g., those containing the minimum set of data for the Base Service of the Road Accident Emergency Response System, or data indicating that the alarm system has operated on the vehicle. When this field is analysed by the dispatcher, the received higher priority packets are routed faster than the low priority ones, thus, the processing speeds up in case of critical events.

HL	Length of the Transport Layer Protocol header in bytes, including the checksum byte (HCS field)
HE	Defines the method used to encode the Transport Layer Protocol header part that follows this parameter. Reserved
FDL	Number of bytes in the SFRD field with Service Support Layer Protocol data.
PID	ID of the Transport Layer Protocol packet; incremented by 1 after the transmission of each new packet by the sending party. The value in this field is cyclically changed in the range from 0 to 65535, i.e., is wrapped around to 0 when 65535 is reached.
PT	Type of Transport Protocol Layer packet. Field PT may take the following values: 0 — EGTS_PT_RESPONSE (acknowledgement for Transport Protocol Layer); 1 — EGTS_PT_APPDATA (packet including Service Support Layer Protocol data); 2 — EGTS_PT_SIGNED_APPDATA (packet including digitally signed Service Support Layer Protocol data)
PRA	Address of the telematic platform where this packet has been generated. This address is unique across the integrated network, and is used to create acknowledgement packets on the recipient side.
RCA	Address of the telematic platform this packet is intended for. It is used to identify what telematic platform the packet belongs to and how it should be routed via intermediate telematic platforms.
TTL	Time to Live for packet routing between the telematic platforms. Using this parameter prevents endless loops when packets are routed in the systems with complex addressing topology. The TTL is initialised by the telematic platform where the packet originates from, and is set to a maximum permitted number of telematic platforms between the originator and destination. Then, the TTL value is decremented by one each time the packet is relayed by an intermediate platform, and the checksum of the Transport Layer Protocol header is re-calculated accordingly. If the value becomes zero, but the packet still needs to be routed further, the packet is destroyed, and the response with the respective code (PC_TTLEXPIRED, see Appendix C) is returned.

Table 4 (continued)

Parameter (field) name	Parameter (field) name purpose
SFRCS	Checksum. To calculate the checksum for data sent in the SFRD field, the CRC-16—CCITT algorithm is used. This field is present only when the SFRD field is non-empty. An example code of CRC-16 calculation is given in Appendix D.
SFRD	Data structure that depends on the packet type and contains Service Support Layer Protocol data.
HCS	Checksum of Transport Layer Protocol header (from the PRV field to the HCS field, not including the latter field). To calculate the HCS field value, the CRC-8 algorithm is applied to all bytes of the said sequence. An example code of CRC-8 calculation is given in Appendix E.

5.6.1.4 The packet assembly flowchart for the Transport Layer Protocol is given in Figure 2.

5.6.2 Data structures of different packet types

The data format of the SFRD field depends on the type of the Transport Layer Protocol packet.

5.6.2.1 Data structure of EGTS_PT_APPDATA packet

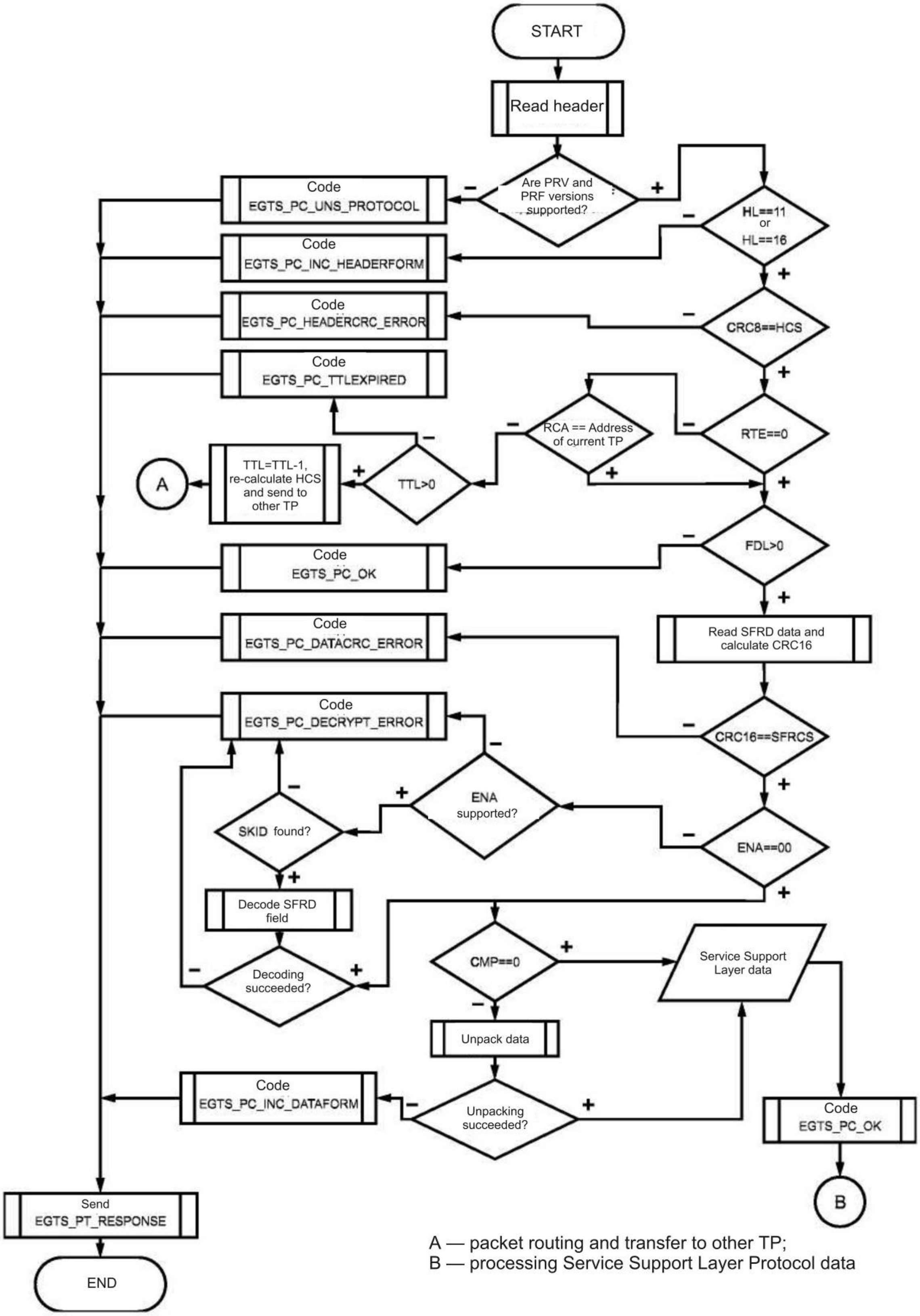
Packets of this type are intended for transfer of one or several structures that include the information of the Service Support Layer Protocol. The SFRD field format of the EGTS_PT_APPDATA packet is specified in Table 5.

Table 5 — SFRD field format for packets of EGTS_PT_APPDATA type

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Type	Data type	Size in bytes
SDR 1 (Service Data Record)								O	BINARY	9 ... 65517
SDR 2								O	BINARY	9 ... 65517
…								…	…	…
SDRn								O	BINARY	9 ... 65517
Note — The SDR 1, SDR 2, SDRn structures contain Service Support Layer Protocol data. One or several such structures may be included one after another in the SFRD field. The description of their internal format is given in Section 6.										

5.6.2.2 Data structure of EGTS_PT_RESPONSE packet

Packets of this type are used to acknowledge that the Transport Layer Protocol packet has been received. This packet type includes data processing results for the previously received Transport Layer Protocol packet. The SFRD field format of the EGTS_PT_RESPONSE packet is specified in Table 6.


START
Read header
Are PRV and PRF versions supported?
+
–
Code EGTS_PC_UNS_PROTOCOL
HL==11 or HL==16
Code EGTS_PC_INC_HEADERFORM
CRC8==HCS
Code EGTS_PC_HEADERCRC_ERROR
Code EGTS_PC_TTLEXPIRED
RTE==0
RCA == Address of current TP
TTL>0
TTL=TTL-1, re-calculate HCS and send to other TP
A
FDL>0
Code EGTS_PC_OK
Read SFRD data and calculate CRC16
Code EGTS_PC_DATACRC_ERROR
CRC16==SFRCS
Code EGTS_PC_DECRYPT_ERROR
ENA supported?
ENA==00
SKID found?
Decode SFRD field
Decoding succeeded?
CMP==0
Service Support Layer data
Unpack data
Unpacking succeeded?
Code EGTS_PC_INC_DATAFORM
Code EGTS_PC_OK
B
Send EGTS_PT_RESPONSE
END

A — packet routing and transfer to other TP;
B — processing Service Support Layer Protocol data

Figure 2 — Assembly flowchart of Transport Layer Protocol packet on reception

Table 6 — SFRD field format for packets of EGTS_PT_RESPONSE type

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Type	Data type	Size in bytes
RPID (Response Packet ID)								M	USHORT	2
PR (Processing Result)								M	BYTE	1
SDR 1 (Service Data Record)								O	BINARY	9...65514
SDR 2								O	BINARY	9...65514
…								…	…	…
SDRn								O	BINARY	9...65514

Notes

1 The RPID parameter is an ID of the Transport Layer packet being acknowledged.

2 The PR parameter is the processing result for a packet part that pertains to the Transport Layer (checksum calculation for the Transport Layer header and the Service Support Layer data, packet size verification, decision on whether the packet must be routed further, etc.). The list of possible result processing codes is given in Appendix C.

3 The SDR 1, SDR 2, SDRn structures contain Service Support Layer Protocol data. One or several such structures may be included one after another.

5.6.2.3 Data structure of EGTS_PT_SIGNED_APPDATA packet

Such packets are used where the transmitted Service Support Layer structures must be completed with digital signature data that identify the sender of a given packet. The SFRD field format of the EGTS_PT_SIGNED_APPDATA packet is specified in Table 7.

5.6.2.4 Each packet of the type EGTS_PT_APPDATA or EGTS_PT_SIGNED_APPDATA sent from the IVDS to the telematic platform, or in the opposite direction, shall be acknowledged with a packet of the type EGTS_PT_RESPONSE with its PID field containing an ID from the original EGTS_PT_APPDATA or EGTS_PT_SIGNED_APPDATA packet.

Figure 3 illustrates the sequence of packet exchange during interaction between the IVDS and the telematic platform.

Table 7 — SFRD field format for packets of EGTS_PT_SIGNED_APPDATA type

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Type	Data type	Size in bytes
SIGL(Signature Length)								M	USHORT	2
SIGD (Signature Data)								O	BINARY	0...512
SDR 1 (Service Data Record)								O	BINARY	9...65515
SDR 2								O	BINARY	9...65515
…								…	…	…
SDRn								O	BINARY	9...65515

Notes

1 The SIGL parameter is the data length of a digital signature included in the SIGD field.

2 The SIGD parameter contains digital signature data themselves.

3 The SDR 1, SDR 2, SDRn structures contain Service Support Layer Protocol data. One or several such structures may be included following one another.

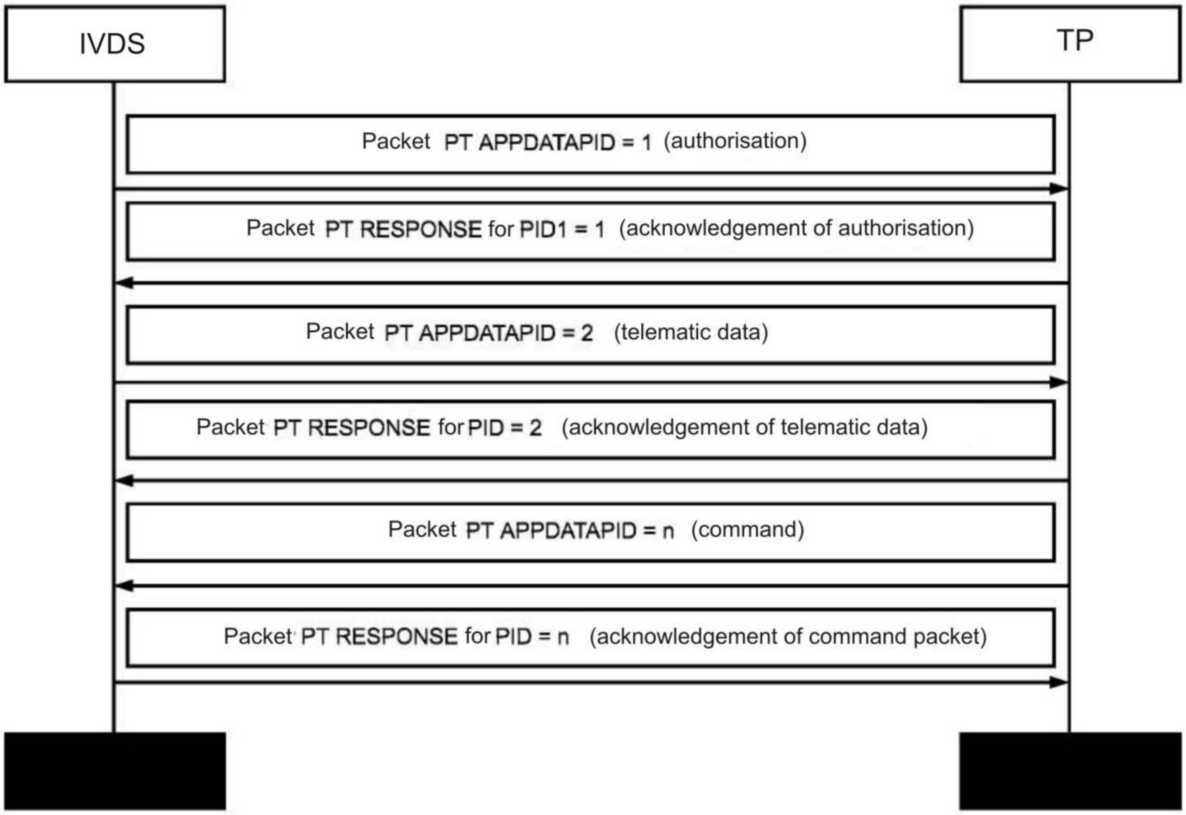

IVDS
TP
Packet PT APPDATAPID = 1 (authorisation)
Packet PT RESPONSE for PID1 = 1 (acknowledgement of authorisation)
Packet PT APPDATAPID = 2 (telematic data)
Packet PT RESPONSE for PID = 2 (acknowledgement of telematic data)
Packet PT APPDATAPID = n (command)
Packet PT RESPONSE for PID = n (acknowledgement of command packet)

Figure 3 — Interaction between IVDS and telematic platform at Transport Protocol Layer level

5.7 Description of data structure for case when SMS is used as redundant data link

5.7.1 SMS message structure

When the SMS service is used to send Transport Layer Protocol packets, the PDU mode [5], [6] is used. This mode enables the transmission of both text and binary data using SMS services of GSM operators. As the Transport Layer Protocol described this Standard operates with binary data, the PDU mode is most appropriate when the SMS service is used as a redundant data link for the Transport Layer.

5.7.1.1 An 8-bit encoding is used for SMS transmission. The format of SMS messages intended for transmission in the PDU mode is shown in Table 8, and is based on the structure detailed in [6], (section 9).

Table 8 — SMS format for use in PDU mode

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Type	Size in bytes
SMSC_AL (SMSC Address Length)								M	1
SMSC_AT (SMSC Address Type)								O	0 or 1
SMSC_A (SMSC Address)								O	0 or 6
TP_RP	TPJJDHI	TP_SRR	TP_VPF		TP_RD	TP_MTI		M	1
TP_MR (MessageReference)								M	1
TP_DA_L (Destination Address Length)								M	1
TP_DA_T (Destination Address Type)								M	1

Table 8 (continued)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Type	Size in bytes
TP_DA (Destination Address)								M	6
TP_PID (Protocolldentifier)								M	1
TP_DCS (Data Coding Schema)								M	1
TP_VP (ValidityPeriod)								O	0, 1 or 7
TP_UDL (User Data Length)								M	1
TP_UD (UserData)								O	0 …140

5.7.1.2 The parameters included in SMS messages sent in the PDU mode are described below.

- SMSC_AL — useful SMSC address data length, in octets;

- SMSC_AT — SMSC address format type.

The permitted values of the SMSC_AT parameter are given in Table 9.

Table 9 — Format of TP_DA_T and SMSC_AT (address type) fields

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Size in bytes
1	TON			NPI				1

The TP_DA_T and SMSC_AT parameters listed in Table 9 have the following meaning:

- TON – Type Of Number. This parameter may have the following values:

a) 000 — unknown;
b) 001 — International format;
c) 010 — National format;
d) 011 — network-specific number;
e) 100 — subscriber number;
f) 101 — alphanumeric code (as per [2], in 7-bit encoding by default);
g) 110 — shortcut;
h) 111 — reserved.

- NPI – NumberingPlanIdentification (applicable to TON values of 000, 001 and 010 only). This parameter may have the following values:

a) 0000 — unknown;
b) 0001 — numbering plan for ISDN telephony;
c) 0011 — numbering plan for data transmission;
d) 0100 — telegraph;
e) 1000 — National;
f) 1001 — private;
g) 1111 — reserved.

This field is optional, and present when the SMSC_AL parameter value is greater than 0;

- SMSC_A — SMSC address. Each decimal digit of the number is represented by four bits (lower 4 bits define the high order digit, and upper 4 bits, the low order one) and, if the number of digits in the address is odd, then bits 4 through 7 of the last byte are set to 0xF (1111b). This parameter is optional, and included depending on the SMSC_AL parameter value. If SMSC_A is missing, the SMSC from the SIM card is used;

- TP_MTI (Message Type Indicator) — must contain a binary value of 01;

- TP_RD (Reject Duplicates) — defines whether the SMSC must accept this message for processing if the previous message has been sent from the same number, has the same values of the TP_MR field and of destination number in the TP_DA field, but has not been processed yet;

- TP_VPF (Validity Period Format) — TP_VP parameter format; see Table 10 for possible values;

Table 10 — TP_VP field format depending on TP_VPF field value

Bit values		Description
0	0	Field TP_VP is not transmitted
1	0	Field TP_VP has "relative time" format, and size of 1 byte
0	1	Field TP_VP has "extended time" format, and size of 7 bytes
1	1	Field TP_VP has "absolute time" format, and size of 7 bytes

- TP_SRR (Status Report Request) — setting this bit to 1 means that this message must be acknowledged from the SMSC side;
- TP_UDHI (User Data Header Indicator) — setting this bit to 1 means that the TP_UD_HEADER (user data header) must be transmitted;
- TP_RP (Reply Path) — defines if the RP field is present in the message;
- TP_MR — message ID (must be incremented by 1 each time a new message is sent);
- TP_DA_L — useful data length of the destination address, in octets;
- TP_DA_T — format type of the destination address; the permitted values of the TP_DA_T and SMSC_AT parameters are listed in Table 9;
- TP_DA — destination address. The number is encoded as described for the SMSC_A parameter;
- TP_PID — protocol ID (must be set to 00);
- TP_DCS — data encoding type (must be set to 0x04; this corresponds to an 8-bit encoding with no compression);
- TP_VP — time of message validity. The format of this field is defined in Table 10. The parameter is optional; its presence and size depend on the TP_VPF field value;
- TP_UDL — length of message data in the TP_DL field, expressed in bytes as long as an 8-bit encoding is used;
- TP_UD — user data themselves. The format of this field depending on the TP_UDHI field value is shown in Table 11.

Table 11 — TP_UD field format

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Type	Size in bytes
LUDH (Length of User Data Header)								O	1
IEI "A"» (Information-Element-Identifier "A")								O	1
LIE "A" (Length of Information-Element "A")								O	1
IED "A" (Information-Element-Data of "A")								O	1...n
IEI "B" (Information-Element-Identifier "B")								O	1
LIE "B" (Length of Information-Element "B")								O	1
IED "B" (Information-Element-Data of "B")								O	1...n
IEI "N" (Information-Element-Identifier "N")								O	1
LIE "N" (Length of Information-Element "N")								O	1
IED "N" (Information-Element-Data of "N")								O	1...n
UD (User Data)								M	1...140

The TP_UD field parameters listed in Table 11 have the following meaning:

- LUDH — length of user data header in bytes, not counting this field itself;
- IEI "A", IEI "B", IEI "N" — identifiers of "A", "B" and "N" information elements, respectively, which determine the element types, and may have the following values (in hexadecimal notation):

a) 00 — part of SMS message being concatenated;
b) 01 — indicator of special-purpose SMS message;
c) 02 — reserved;
d) 03 — unused;
e) 04—7F — reserved;
f) 80—9F — SME used for special purpose;
g) A0—BF — reserved;
h) C0—DF — SC used for special purpose;
i) E0—FF — reserved.

- LIE "A", LIE "B", LIE "N" — parameters that define the data length, in bytes, of information elements "A", "B" and "N", respectively, not counting this field;
- IED "A", IED "B", IED "N" — parameters that define the data of information elements "A", "B" and "N", respectively;
- UD — user data. The size of this field depends on whether the TP_UD_HEADER user data header containing the LUDH, IEI, LIE and IED fields is present. If the header is not present, the size is equal to the TP_UDL value described in Table 8. Otherwise, it is calculated as a difference (TP_UDL — LUDH-1).

If the IEI element of TP_UD_HEADER is set to 00, the IED field shall be as specified in Table 12.

Table 12 — Data field format of information element describing SMS message part to be concatenated

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Type	Size in bytes
CSMRN (Concatenated Short Message Reference Number)								M	1
MNSM (Maximum Number of Short Messages)								M	1
SNCSM (Sequence Number of Current Short Message)								M	1

Notes

1 CSMRN — concatenated SMS message number; must be identical for all parts of long SMS messages.

2 MNSM — total number of message parts making up long SMS message; must be in a range from 1 to 255.

3 SNCSM — part number of long SMS message being transferred. This number is incremented by 1 each time a new message part is sent, and must be in a range from 1 to 255. If the value in this field is zero or is greater than the value in the MNSM field, the recipient must ignore the whole information element.

5.7.2 Description of data format used for transmission

5.7.2.1 When the SMS service is used for data exchange between the IVDS and the telematic platform, packets are encapsulated into the TP_UD field (see Table 8) after their packing according to Transport Layer Protocol and Service Support Layer Protocol rules, subject to the requirement that the total packet size must not exceed 140 bytes. The authorisation is not used in this case, and the transmitted packets need not be acknowledged using a packet of type EGTS_PT_RESPONSE as per Transport Layer Protocol and a sub-record EGTS_SR_RECORD_RESPONSE as per Service Support Layer Protocol. An SMS transfer notification is an indicator that the packet has been transferred to the IVDS successfully.

Each EGTS_SR_COMMAND_DATA sub-record of EGTS_COMMAND_SERVICE containing a command or message must be acknowledged by the EGTS_SR_COMMAND_DATA sub-record with the respective values set in the CT (CommandType) and CCT (CommandConfirmationType) fields. If a command is sent to the IVDS via SMS, the respective EGTS packet with a command reception acknowledgement in the form of the EGTS_SRCOMMAND_DATA sub-record must be sent from the IVDS by SMS.

5.7.2.2 If the SMS message includes a digital signature, the Transport Layer packet of the EGTS_PT_SIGNED_APPDATA type is used for its transfer.

5.7.2.3 If the size of the protocol data packet exceeds 140 bytes, the SMS message concatenation technique described in [6] (9.2.3.24.1) is used. In essence, this technique implies that all user data subject for transfer are divided into parts and sent as separate SMS messages. Upon that, each message contains a dedicated structure that defines the total number of parts as well as the procedure used to assemble them on the recipient side. Such structure is put into the TP_UD_HEADER field that includes the information element describing the SMS message part to be concatenated. Thus, given the user data header size specified above and the maximum number of message part equal to 255, the maximum possible packet size may be 255 times (140 – 6) = 34170 bytes when an 8-bit encoding is used.

When the SMS service is used as a channel for transmission of EGTS packets to the IVDS, the size of a single EGTS packet shall be limited to 10×(140 – 6) = 1360 bytes since larger packets may overflow the IVDS receive buffer. The maximum size of 1360 bytes will allow transmitting elementary EGTS messages using digital signatures (fields SIGL/SIGD) and authorisation codes (ACL/AC).

5.8 Timing and numerical parameters of Transport Layer Protocol for case when packet data transmission is used

The designation and description of timing and numerical parameters of the Transport layer Protocol are specified in Table 13.

Table 13 — Timing and numerical parameters of Transport Layer Protocol

Name	Data type	Range of values	Default value	Description
TL_RESPONSE_TO	BYTE	0...255	5	Wait time for packet acknowledgement at Transport Layer, seconds
TL_RESEND_ATTEMPTS	BYTE	0...255	3	Number of attempts to send unacknowledged packets
TL_RECONNECT_TO	BYTE	0...255	30	Time to expire before attempting to re-establish communication on a broken link, s

6 Service Support Layer Protocol (general part)

6.1 Purpose of Service Support Layer Protocol

6.1.1 The purpose of the Service Support Layer Protocol is to enable data exchange between the components of the Road Accident Emergency Response System so that a level sufficient for provision of IT services to the consumers is maintained. Each IT service corresponds to a separate system service which is a key element of the System framework constructed using the Service Support Layer Protocol.

6.1.2 Service Support Layer Protocol has the following basic functions:

- exchange of information messages that contain data to be processed by various services, and data queries to be responded by those services;
- notification on results of Service Support Layer data delivery and processing;
- identification of service type associated with given data;
- evaluation of data attributes (number, type, contents, size, encoding, etc.).

6.2 Message exchange

The primary structure of the Service Support Layer Protocol is a record. The record contains all data required to process information or to request particular services. Each record may include several sub-records containing necessary data and defining the actions to be taken by the service which is responsible for processing of a given sub-record.

6.3 Notifications on delivery and processing of Service Support Layer data

At the Service Support Layer, the sending party is notified on the results of data delivery and processing by means of the data record acknowledgement technique which makes use of special sub-records containing an ID of the received/processed record.

6.4 Service identification for data used in Service Support Layer Protocol

In order to identify that the record is associated with a particular service, a service type ID defining the features and properties of the processed data is used. The service type is used to identify the service when the inter-platform routing is performed, and is unique within the framework of the Service Layer Support Protocol.

6.5 Evaluating data properties in Service Support Layer Protocol

The data used in the Service Support Layer Protocol are stored in sub-records with an ID that is unique within the scope of a given service and with a strictly defined data layout depending on the sub-record type. Such data organisation in the Service Support Layer Protocol ensures that the type, physical meaning, size and packing method of data are defined unambiguously.

6.6 Data structures used in Service Support Layer Protocol

6.6.1 Overall structure

An overall structure defined in the Service Support Layer Protocol for inclusion in Transport Layer Protocol packets may contain one or multiple successive records of various data content intended for different services. The said data structure is illustrated in Figure 4.

Service Support Layer data		
Record RID = 1	Record RID = 2	Record RID = N

Figure 4 — Overall structure of Service Support Layer data

6.6.2 Structure of individual records

6.6.2.1 Record layout

Individual records defined in the Service Support Layer Protocol consist of a record header and record data. The layout of individual records is illustrated in Figure 5.

Record header	Record data		
	Sub-record 1	…	Sub-record N

Figure 5 — Layout of individual records at Service Support Layer

A record header includes the parameters that define the source and recipient service types, record identifier, object identifier (e.g., IVDS), length of transferred data, as well as various flags that govern the presence of optional parameters and define the processing method.

Record data may include one or multiple sub-records defining the data types and containing the data to be transferred.

6.6.2.2 Record format

The format of individual Service Support Layer records is described in Table 14.

T a b l e 14 — Format of individual records defined in Service Support Layer Protocol

<table>
<tr><th>Bit 7</th><th>Bit 6</th><th>Bit 5</th><th>Bit 4</th><th>Bit 3</th><th>Bit 2</th><th>Bit 1</th><th>Bit 0</th><th>Type</th><th>Data type</th><th>Size in bytes</th></tr>
<tr><td colspan="8">RL (Record Length)</td><td>M</td><td>USHORT</td><td>2</td></tr>
<tr><td colspan="8">RN (Record Number)</td><td>M</td><td>USHORT</td><td>2</td></tr>
<tr><td colspan="8">RFL (Record Flags)</td><td rowspan="2">M</td><td rowspan="2">BYTE</td><td rowspan="2">1</td></tr>
<tr><td>SSOD</td><td>RSOD</td><td colspan="3">RPP</td><td>TMFE</td><td>EVFE</td><td>OBFE</td></tr>
<tr><td colspan="8">OID (Object Identifier)</td><td>O</td><td>UINT</td><td>4</td></tr>
<tr><td colspan="8">EVID (Event Identifier)</td><td>O</td><td>UINT</td><td>4</td></tr>
<tr><td colspan="8">TM (Time)</td><td>O</td><td>UINT</td><td>4</td></tr>
<tr><td colspan="8">SST (Source Service Type)</td><td>M</td><td>BYTE</td><td>1</td></tr>
<tr><td colspan="8">RST (Recipient Service Type)</td><td>M</td><td>BYTE</td><td>1</td></tr>
<tr><td colspan="8">RD (Record Data)</td><td>M</td><td>BINARY</td><td>3...65498</td></tr>
</table>

The parameters of a Service Support Layer record listed in Table 14 have the following purpose:

- RL — Record Length. Defines the size of data sent in the RD field;

- RN — Record Number. The value in this field is incremented from 0 to 65535 with wraparound, i.e., reset to 0 when 65535 is reached. This value is used for acknowledgement of records;

- RFL — Record Flags. Contains the bit flags that indicate whether this packet includes the OID, EVID and TM fields that describe the data included in the record;

- SSOD — Source Service On Device. Bit flag that defines the source service location:

a) 1 — source service is on the IVDS side;

b) 0 — source service is on the telematic platform;

- RSOD – recipient service on device. Bit flag that defines the recipient service location:

a) 1 — recipient service is on the IVDS side;

b) 0 — recipient service is on the telematic platform;

- RPP — Record Processing Priority. Bit field that defines the service processing priority for this record. It may contain a digital value from 0 (highest priority) to 7 (lowest priority).

- TMFE — Time Field Exists. Bit field that defines if the TM field is present in this packet:

a) 1 — TM field present;

b) 0 — TM field is not present;

- EVFE — Event ID Field Exists. Bit field that defines if the EVID field is present in this packet:

a) 1 — EVID field present;

b) 0 — EVID field is not present;

- OBFE — Object ID Field Exists. Bit field that defines if the OID field is present in this packet:

a) 1 — OID field present;

b) 0 — OID field is not present;

- OID — identifier of the object this record has been generated by or is intended for (unique IVDS identifier). If the record is transmitted to the IVDS in response to a command from the TP, the same OID as the one received in that command must be specified in order to indicate that the data belong to a correct object and the request matches the response on the TP side. The algorithm of such OID use is shown in Figure 6.

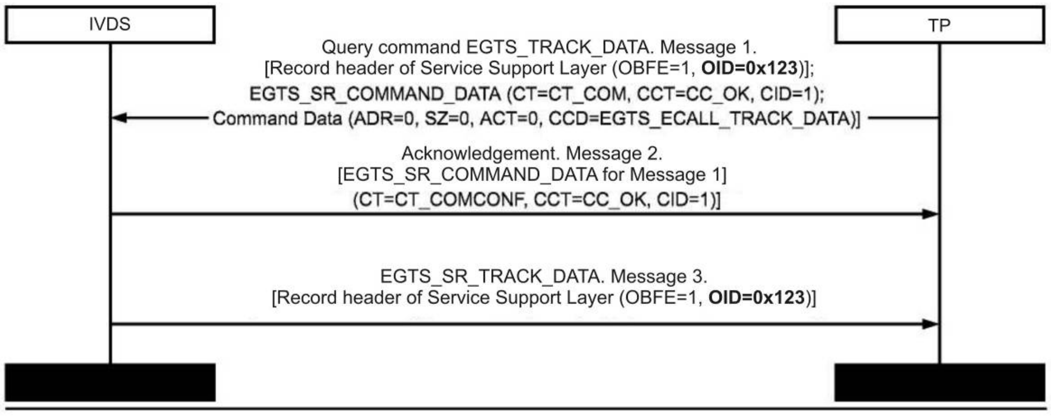

IVDS
TP
Query command EGTS_TRACK_DATA. Message 1.
[Record header of Service Support Layer (OBFE=1, OID=0x123)];
EGTS_SR_COMMAND_DATA (CT=CT_COM, CCT=CC_OK, CID=1);
Command Data (ADR=0, SZ=0, ACT=0, CCD=EGTS_ECALL_TRACK_DATA)]
Acknowledgement. Message 2.
[EGTS_SR_COMMAND_DATA for Message 1]
(CT=CT_COMCONF, CCT=CC_OK, CID=1)]
EGTS_SR_TRACK_DATA. Message 3.
[Record header of Service Support Layer (OBFE=1, OID=0x123)]

Figure 6 — Algorithm of OID use

- EVID — unique event identifier. The EVID field sets a global event identifier that is used when multiple information entities must be logically bound to a single event whereas the entities themselves may either belong to different data packets, or be separated in time. Using this ID, the application software may bind such entities together when it reports event data to the user. For example, if a series of snapshots is produced after the emergency button press, the EVID field will be specified in each service record associated with the button press event until all entities related to this event have been delivered, however long the transmission of the complete information pool may be;
- TM — time of record creation on the originator side (seconds since 00:00:00 01.01.2010 UTC). If multiple records bound to the same object and the same time moment are transmitted in a single packet, the TM (time mark) field may be sent in the first record only;
- SST — identifier of the source service that has generated this record. For example, a service processing navigation data on the IVDS side, a command service on the telematic platform side, etc.;
- RST— identifier of the recipient service for this record. For example, a service processing navigation data on the telematic platform side, a command service on the IVDS side, etc.;
- RD — field containing the data intrinsic to a particular service type (one or several sub-records of the service type indicated in the SST or RST field as appropriate for the type of data being transmitted).

6.6.3 Overall structure of sub-records

The format of individual Service Support Layer sub-records is described in Table 15.

Table 15 — Format of individual sub-records defined in Service Support Layer Protocol

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Type	Data type	Size in bytes
SRT (Subrecord Type)								M	BYTE	1
SRL (Subrecord Length)								M	USHORT	2
SRD (Subrecord Data)								O	BINARY	0...65495

Notes

1 SRT — sub-record type (the sub-type of transmitted data that belongs to the complete set of types of a single service). Type 0 is special, and is reserved for the sub-records used to acknowledge data for each service. The specific values of sub-record type numbers are defined by the service logics itself. The only Protocol requirements are that the number must be present and that zero ID is reserved.

2 SRL — length of sub-record data stored in the SRD field, in bytes.

3 SRD — sub-record data. Data in this field are specific to each combination of service ID and sub-record type.

6.6.4 Each data record at the Service Support Layer shall be responded with an acknowledgement containing the sub-record with the ID of the record being acknowledged and with the result of its processing. The flowchart illustrating the acknowledgement mechanism for message exchange at the Service Support Layer is shown in Figure 7.

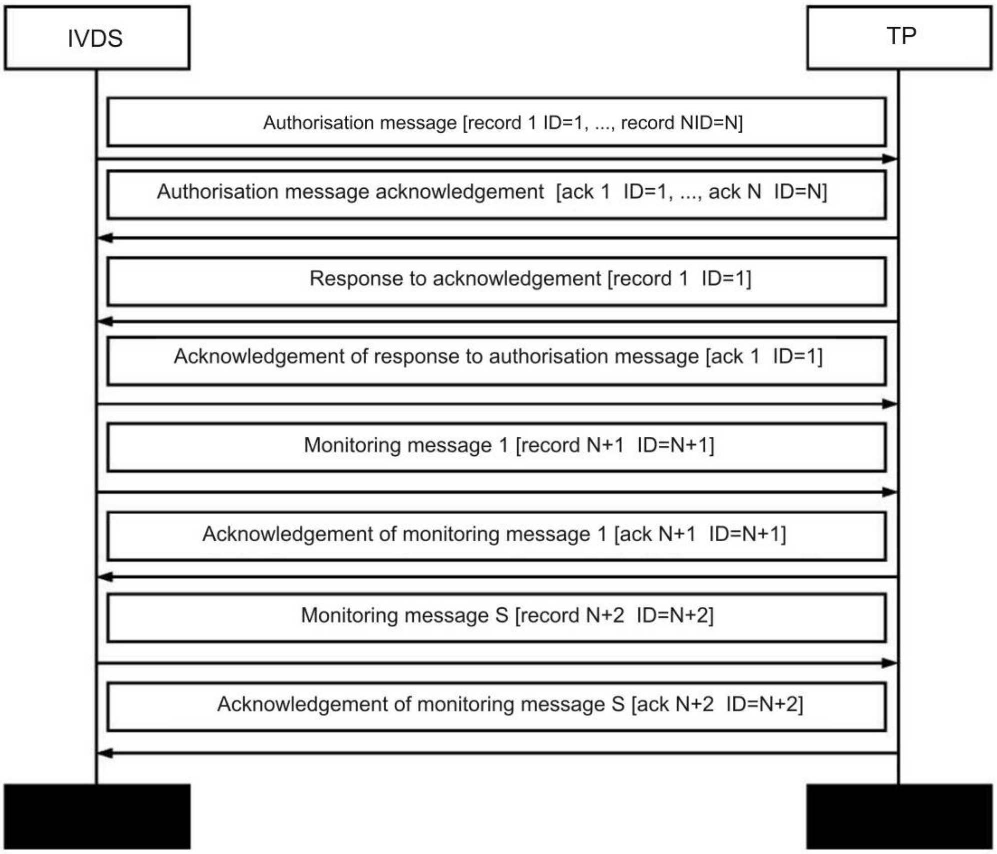

IVDS
TP
Authorisation message [record 1 ID=1, ..., record NID=N]
Authorisation message acknowledgement [ack 1 ID=1, ..., ack N ID=N]
Response to acknowledgement [record 1 ID=1]
Acknowledgement of response to authorisation message [ack 1 ID=1]
Monitoring message 1 [record N+1 ID=N+1]
Acknowledgement of monitoring message 1 [ack N+1 ID=N+1]
Monitoring message S [record N+2 ID=N+2]
Acknowledgement of monitoring message S [ack N+2 ID=N+2]

Figure 7 — Message exchange flow

Each message used in the Service Support Layer Protocol contains a header and a Transport Layer checksum as well as one or several Service Support Layer records. Moreover, each message may contain both data records and acknowledgements for the records received before.

6.7 Description of services

6.7.1 The list of services supported by the Service Support Layer Protocol with their decimal identifiers and descriptions is presented in Table 16.

Table 16 — List of services supported by Protocol

Code	Name	Description	AE[1)]	BSE[2)]	ASE[3)]
1	EGTS_AUTH_SERVICE	This service type is used for IVDS authentication on the telematic platform. When the TCP/IP protocol is used, the IVDS must pass this procedure since no for further interaction is possible without it.	+	-	+

Table 16 (continued)

Code	Name	Description	AE[1)]	BSE[2)]	ASE[3)]
2	EGTS_TELEDATA_SERVICE	This service is intended for processing of telematic data (coordinate data, sensor operation data, etc.) received from the IVDS.	+	-	+
4	EGTS_COMMANDS_SERVICE	This service type is intended for processing of control and configuration commands as well as data messages and status messages transmitted between the IVDS, telematic platform and operators.	+	+	+
9	EGTS_FIRMWARE_SERVICE	This service is intended for downloading data to the IVDS, including configuration data, firmware of the IVDS itself as well as of its connected peripheral equipment that supports remote software updates.	+	+	+
10	EGTS_ECALL_SERVICE	This service ensures proper ERA functionality. It is described in Section 7.	+	+	+
Note — IVDS configuration options: 1 IVDS manufactured in auxiliary equipment configuration. 2 IVDS manufactured in standard equipment configuration and intended for the Base Service of the System only. 3 IVDS manufactured in standard equipment configuration and intended for other System services in addition to the Base Service.					

6.7.2 EGTS_AUTH_SERVICE

This service is used to perform IVDS authentication on the telematic platform side and to acquire IVDS accounting data as well as IVDS infrastructure information (the list and software versions of modules, units, peripheral equipment, and the data regarding the vehicle). This service may be used by the IVDS only after a new TCP/IP connection with the telematic platform is established.

The requirements established in this subsection of the Standard solely apply to IVDS manufactured in auxiliary equipment configuration, and do not apply to standard IVDS that support the Base Service only.

The list of sub-records used by EGTS_AUTH_SERVICE is given in Table 17.

Table 17 — List of EGTSAUTHSERVICE sub-records

Code	Name	Description
0	EGTS_SR_RECORD_RESPONSE	Used for acknowledgement of processing completed for Service Support Layer Protocol records. This sub-record type shall be supported by all services.
1	EGTS_SR_TERM_IDENTITY	Used by the IVDS when it sends an authorisation request to the telematic platform. This sub-record contains IVDS accounting data.
2	EGTS_SR_MODULE_DATA	Intended for providing the telematic platform with the information related to the infrastructure on the IVDS side, to available IVDS units and modules, their status and parameters. This sub-record is optional, and the decision whether its fields must be filled and the sub-record delivered is left to the IVDS developer. Each sub-record describes a single module. One record may contain several such sub-records in series, so that the data on multiple components of IVDS hardware and its peripheral equipment may be transferred.

Table 17 (continued)

Code	Name	Description
3	EGTS_SR_VEHICLE_DATA	Used by the IVDS for transmission of vehicle data to the telematic platform.
6	EGTS_SR_AUTH_PARAMS	Used by the telematic platform to provide the IVDS with the data related to encryption technique and parameters required for further interaction.
7	EGTS_SR_AUTH_INFO	Intended for providing the telematic platform with IVDS authentication data using the data encryption parameters previously received from the platform side.
8	EGTS_SR_SERVICE_INFO	Used to inform the recipient (either the IVDS or telematic platform as appropriate for the transfer direction) on the supported services, and to request a particular set of necessary services (from the IVDS to the TP).
9	EGTS_SR_RESULT_CODE	Used by the telematic platform to inform the IVDS on results of its authentication.

6.7.2.1 EGTS_SR_RECORD_RESPONSE sub-record.

The sub-record format is specified in Table 18.

Table 18— EGTS_SR_RECORD_RESPONSE sub-record format

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Type	Data type	Size in bytes
CRN (Confirmed Record Number)								M	USHORT	2
RST (Record Status)								M	BYTE	1

The fields of the EGTS_SR_RECORD_RESPONSE sub-record have the following purpose:

- CRN — number of a record being acknowledged (RN field value from the record being processed);
- RST — record processing status. Processing result codes are given in Appendix C.

After the acknowledgement is received by the sender, the sender analyses the RST field of the EGTS_SR_RECORD_RESPONSE sub-record, and either deletes the record in the internal storage in case of successful processing, or takes the required actions in case of an error, as appropriate for its reason.

Combining the Transport Layer acknowledgement (EGTS_PT_RESPONSE packet type) with the Service Support Layer acknowledgement sub-records EGTS_SR_RECORD_RESPONSE is recommended.

6.7.2.2 EGTS_SR_TERM_IDENTITY sub-record

The sub-record format is specified in Table 19.

Table 19 — Format of EGTS_SR_TERM_IDENTITY sub-record used for EGTS_AUTH_SERVICE

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Type	Data type	Size in bytes
TID (TerminalIdentifier)								M	UINT	4
Flags								M	BYTE	1
MNE	BSE	NIDE	SSRA	LNGCE	IMSIE	IMEIE	HDIDE			
HDID (Home Dispatcher Identifier)								O	USHORT	2
IMEI (International Mobile Equipment Identity)								O	STRING	15
IMSI (International Mobile Subscriber Identity)								O	STRING	16
LNGC (Language Code)								O	STRING	3

Table 19 (continued)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Type	Data type	Size in bytes
NID (Network Identifier)								O	BINARY	3
BS (Buffer Size)								O	USHORT	2
MSISDN (Mobile Station Integrated Services Digital Network Number)								O	STRING	15

The fields of the EGTS_SR_TERM_IDENTITY sub-record have the following purpose:

- TID — unique identifier assigned when the IVDS is programmed. Zero value in this field means that the IVDS has not passed the configuration procedure, either completely or partially. This ID is assigned by System Operator; it uniquely identifies the set of IVDS accounting data. TID is assigned when the IVDS is installed as auxiliary equipment, and the IVDS accounting data (IMSI, IMEI, serial_id) are handed over to System Operator. If the IVDS is used in part of standard equipment, the TID is reported to Operator by the vehicle manufacturer when he sends the accounting data (VIN, IMSI and IMEI);
- HDIDE — bit flag indicating whether the HDID field is present in the sub-record (set if present, cleared otherwise);
- IMEIE — bit flag indicating whether the IMEI field is present in the sub-record (set if present, cleared otherwise);
- IMSIE — bit flag indicating whether the IMSI field is present in the sub-record (set if present, cleared otherwise);
- LNGCE — bit flag indicating whether the LNGC field is present in the sub-record (set if present, cleared otherwise);
- SSRA — bit flag identifying the service algorithm in use (set if the simple algorithm is used, cleared if the algorithm based on requests for services is used);
- NIDE — bit flag indicating whether the NID field is present in the sub-record (set if present, cleared otherwise);
- BSE — bit flag indicating whether the BS field is present in the sub-record (set if present, cleared otherwise);
- MNE — bit flag indicating whether the MSISDN field is present in the sub-record (set if present, cleared otherwise);
- HDID — ID of "home" telematic platform (where the detailed accounting data for the IVDS are stored);
- IMEI — mobile equipment (modem) identity. If the IMEI can not be determined, the IVDS shall fill all 15 character positions of this field with zeroes;
- IMSI — mobile subscriber identity. If the IMSI can not be determined, the IVDS shall fill all 16 character positions of this field with zeroes;
- LNGC — language code preferable for use on the IVDS side, e.g., "rus" means Russian;
- NID — identifier of the operator network where the IVDS is registered. Only 20 lower bits are used. It represents a pair of MCC-MNC codes, and has the format specified in Table 20;
- BS — maximum IVDS input buffer size in bytes. The size of each data packet transmitted to the IVDS shall not exceed this value. The BS field may accept different values (e.g., 1024, 2048, or 4096) depending on the hardware and software implementation of a particular IVDS;
- MSISDN — mobile subscriber telephone number. If this parameter can not be determined, the device shall fill all 15 character positions of this field with zeroes (the field format is described in National numbering plans approved by respective regulatory legal acts[1)]).

Whether the HDID field must be transmitted is defined by IVDS settings and is reasonable when the IVDS may be connected to a telematic platform other than its home platform, e.g., when the geographically distributed network of platforms is used. If the home platform is a single platform in use, the HDID transmission is not required.

[1)] In the Russian Federation, the "Russian numbering system and plan" has been approved by Order No. 142 dated November 17, 2006, of the RF Ministry for Communication and Information Technologies.

The "simple" algorithm of service allocation implies that the IVDS may access all services, and is permitted to send the data to the required service immediately as long as the IVDS remains in this mode. Depending on permissions currently enabled for a given IVDS on the telematic platform, the data packet for the service may be responded with an acknowledgement record containing the relevant error indicator. Such algorithm is recommended for systems where allocation of rights for service use is simple. This decreases the traffic and accelerates the IVDS authorisation.

The algorithm based on requests for services implies that the IVDS must obtain information on the services available for use from the telematic platform prior to making use of particular services (prior to sending data). The request for services may be sent either during authorisation or after it. In the first case, the sub-records of the SR_SERVICE_INFO type are added, and Bit 7 of the SRVP field is set to 1. In the second case, such request may also be sent using the SR_SERVICE_INFO sub-records.

Table 20 — Format of NID field of EGTS_SR_TERM_IDENTITY sub-record used for EGTS_AUTH_SERVICE

Bits 20...23	Bits 10...19	Bits 0...9	Type	Data type	Size in bytes
—	MCC (Mobile Country Code)	MNC (Mobile Network Code)	M	BINARY	3

The MCC/MNC pair provides for unambiguous identification of GSM, CDMA, TETRA and UMTS cellular operators as well as of certain satellite operators.

The NID field parameters included in the EGTS_SR_TERM_IDENTITY sub-record have the following meaning:

- MCC — country code;
- MNC — mobile network code within the country.

6.7.2.3 EGTS_SR_MODULE_DATA sub-record

The sub-record format is specified in Table 21.

The fields of the SR_MODULE_DATA sub-record have the following meaning:

- MT — module type indicating the functional nature of the module (1 — base module; 2 — I/O module; 3 — navigation receiver module; 4 — wireless communication module). The above numbering of module types is the recommended one, but individual implementations of the authorisation service are free to extend it by their own numbering that may include external peripheral controllers;
- VID — vendor identifier;
- FWV — module firmware version (its high byte defines the major version specified before a decimal point, and low byte defines the minor version specified after it, e.g., 2.34 is represented by 0x0222);

Table 21 — Format of EGTS_SR_MODULE_DATA sub-record used for EGTS_AUTH_SERVICE

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Type	Data type	Size in bytes
MT (Module Type)								M	SHORT	1
VID (Vendor Identifier)								M	UINT	4
FWV (Firmware Version)								M	USHORT	2
SWV (Software Version)								M	USHORT	2
MD (Modification)								M	BYTE	1
ST (State)								M	BYTE	1
SRN (Serial Number)								0	STRING	0...32
D (Delimiter)								M	BYTE	1
DSCR (Description)								O	STRING	0...32
D (Delimiter)								M	BYTE	1

- SWV— module software version (high byte is a number before a decimal point, low byte, after it);
- MD — modification code of module software;
- ST — state [1 — ON, 0 — OFF, above 127 — fault (see Appendix C)];
- SRN — module serial number;
- D — delimiter of string parameters (always 0);
- DSCR — short description of the module.

6.7.2.4 EGTS_SR_VEHICLE_DATA sub-record

The sub-record format is specified in Table 22. If the IVDS is used in standard equipment configuration as per VIN field data, this sub-record shall be sent along with EGTS_SR_TERM_IDENTITY.

Table 22 — Format of EGTS_SR_VEHICLE_DATA sub-record used for EGTS_AUTH_SERVICE

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Type	Data type	Size in bytes
VIN (Vehicle Identification Number)								M	STRING	17
VHT (Vehicle Type)								M	UINT	4
VPST (Vehicle Propulsion Storage Type)								M	UINT	4

The fields of the EGTS_SR_VEHICLE_DATA sub-record have the following meaning:

- VIN — vehicle identification number;
- VHT — vehicle type:

a) Bits 31—5: unused;
b) Bits 4—0;
c) 0001 — passenger (Class M1);
d) 0010 — bus (Class M2);
e) 0011 — bus (Class M3);
f) 0100 — light cargo vehicle (Class N1);
g) 0101 — heavy cargo vehicle (Class N2);
h) 0110 — heavy cargo vehicle (Class N3);
i) 0111 — motorcycle (Class L1e);
j) 1000 — motorcycle (Class L2e);
k) 1001 — motorcycle (Class L3e);
l) 1010 — motorcycle (Class L4e);
m) 1011 — motorcycle (Class L5e);
n) 1100 — motorcycle (Class L6e);
o) 1101 — motorcycle (Class L7e);

- VPST — vehicle propulsion storage (energy source) type. More than one bit may be set if multiple source types are used. If all bits are 0, the type is not given.

a) Bits 31—6: unused;
b) Bit 5:1 — hydrogen;
c) Bit 4:1 — electricity (above 42 V and 100 A/h);
d) Bit 3:1 — liquid propane (LPG);
e) Bit 2:1 — condensed natural gas (CNG);
f) Bit 1:1 — diesel;
g) Bit 0:1 — gasoline.

6.7.2.5 EGTS_SR_AUTH_PARAMS sub-record

The sub-record format is specified in Table 23.

Table 23 — Format of EGTS_SR_AUTH_PARAMS sub-record used for EGTS_AUTH_SERVICE

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Type	Data type	Size in bytes
FLG (Flags)								M	BYTE	1
—	EXE	SSE	MSE	ISLE	PKE	ENA				
PKL (Public Key Length)								O	USHORT	2
PBK (Public Key)								O	BINARY	0...512
ISL (Identity String Length)								O	USHORT	2
MSZ (Mod Size)								O	USHORT	2
SS (Server Sequence)								O	STRING	0...255
D (Delimiter)								O	BYTE	1
EXP (Exp)								O	STRING	0...255
D (Delimiter)								O	BYTE	1

The fields of the EGTS_SR_AUTH_PARAMS sub-record have the following meaning:

- EXE — bit flag indicating whether the EXP field and the delimiter after it are present in the sub-record (present if set to 1);
- SSE — bit flag indicating whether the SS field and the delimiter after it are present in the sub-record (present if set to 1);
- MSE — bit flag indicating whether the MSZ field is present (present if set to 1);
- ISLE — bit flag indicating whether the ISL field is present (present if set to 1);
- PKE — bit flag indicating whether the PKL and PBK fields are present (present if set to 1);
- ENA — bit field defining the required packet encryption algorithm. If set to 00, encryption is not used and the EGTS_SR_AUTH_PARAMS sub-record contains a single byte, otherwise additional parameters are included depending on the algorithm type and on the values of other FLG field bits;
- PKL — public key length in bytes;
- PBK — public key data;
- ISL — resulting length of identification data;
- MSZ — parameter used in the encryption process;
- SS — special server sequence of bytes used in the encryption process;
- D — delimiter of string-type parameters (always 0);
- EXP — special sequence used in the encryption process.

If encryption is required and the requested encryption algorithm is supported, the party being authorised generates and sends the EGTS_SR_AUTH_INFO record encrypted using the said algorithm. In this case, bits 11 and 12 of the KEYS field in the Transport Layer header are set accordingly, and all data transmitted between the parties since then are sent in encrypted form.

If the requested encryption algorithm is not supported, the initiator sends the EGTS_SR_RECORD_RESPONSE sub-record indicating an error.

Depending on the service request algorithm currently in use, the record may also contain the EGTS_SR_SERVICE_INFO sub-records which define the number and parameters of supported services and of those requested by the initiator.

6.7.2.6 EGTS_SR_AUTH_INFO sub-record

The sub-record format is specified in Table 24.

The fields of the EGTS_SR_AUTH_INFO sub-record have the following meaning:

- UNM — user name;
- D — delimiter of string parameters (always 0);
- UPSW— user password;
- SS — special server sequence of bytes transferred in the EGTS_SR_AUTH_PARAMS sub-record (optional field that is present depending on the encryption algorithm being used).

T a b l e 24 — Format of EGTS_SR_AUTH_INFO record used for EGTS_AUTH_SERVICE

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Type	Data type	Size in bytes
UNM (User Name)								M	STRING	0...32
D (Delimiter)								M	BYTE	1
UPSW (User Password)								M	STRING	0...32
D (Delimiter)								M	BYTE	1
SS (Server Sequence)								O	STRING	0...255
D (Delimiter)								O	BYTE	1

6.7.2.7 Sub-record EGTS_SR_SERVICE_INFO.

The sub-record format is specified in Table 25.

T a b l e 25 — Format of EGTSSRSERVICEJNFO sub-record used for EGTSAUTHSERVICE

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Type	Data type	Size in bytes
ST (Service Type)								M	BYTE	1
SST (Service Statement)								M	BYTE	1
SRVP (Service Parameters)								M	BYTE	1
SRVA	—					SRVRP				

The fields of the EGTS_SR_SERVICE_INFO sub-record have the following meaning:

- ST — service type indicating the functional nature of the service (for example, EGTS_TELEDATA_SERVICE, EGTS_ECALL_SERVICE, etc.);
- SST — current service state (see Table 26);
- SRVP — service parameters;
- SRVA — service attribute (bit flag):

a) 0 — supported service;

b) 1 — requested service;

- SRVRP — service routing priority. Bit field indicating the priority as regards the data transfer to this service (in case of system scaling and use of several application instances for the same service type) is defined by bits 0 and 1:

a) 00 — top priority;

b) 01 — high priority;

c) 10 — medium priority;

d) 11 — low priority.

T a b l e 26 — List of possible service states

Code	Name	Description
0	EGTS_SST_IN_SERVICE	Service is operational and permitted for use
128	EGTS_SST_OUT_OF_SERVICE	Service is not operational (switched off)
129	EGTS_SST_DENIED	Service is not permitted for use
130	EGTS_SST_NO_CONF	Service is not configured
131	EGTS_SST_TEMP_UNAVAIL	Service temporarily unavailable

6.7.2.8 EGTS_SR_RESULT_CODE sub-record

The sub-record format is specified in Table 27.

The fields of the EGTS_SR_SERVICE_CODE sub-record have the following meaning:

- RCD — code indicating the result of authorisation procedure.

Table 27 — Format of EGTS_SR_RESULT_CODE sub-record used for EGTS_AUTH_SERVICE

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Type	Data type	Size in bytes
RCD (Result Code)								M	BYTE	1

6.7.2.9 Description of authorisation procedure

Prior to operation in the infrastructure of System Operator, each IVDS shall be assigned a unique identifier UNIT_ID corresponding to specific IMEI and IMSI values and to other IVDS accounting data required for interaction with System Operator.

The requirement of this clause does not apply to standard systems solely supporting the Base Service. In standard equipment configuration, the EGTS_AUTH_SERVICE is not used. The EGTS_ECALL_SERVICE messages in this case may be sent at once. EGTS_AUTH_SERVICE is employed if GPRS is in use and the server is connected via TCP/IP.

The IVDS may be configured using one of the following methods.

1) After the "Additional functions" button is pressed in the IVDS standby mode and the IVDS completes its registration in a GSM or UMTS network, the cellular operator infrastructure detects a new device and sends it an encrypted SMS message with the accounting data. To transmit these data, the IVDS parameters must be set in the EGTS_SR_COMMAND_DATA sub-record of the EGTS_COMMANDS_SERVICE.

The following IVDS parameters shall be specified: EGTS_GPRS_APN (the access point used to establish the GPRS session), the EGTS_SERVER_ADDRESS parameter that defines the address and port of the server to be connected to via TCP/IP, and the IVDS unique identifier UNIT_ID.

Then, the IVDS parses the SMS message, checks the validity of the data structures, calculates the checksums and compares them to the ones received in the message. If such parsing and checking succeed, the IVDS will start a GPRS session and connect to the specified server via TCP/IP.

The algorithm of such IVDS configuration method is shown in Figure 8.

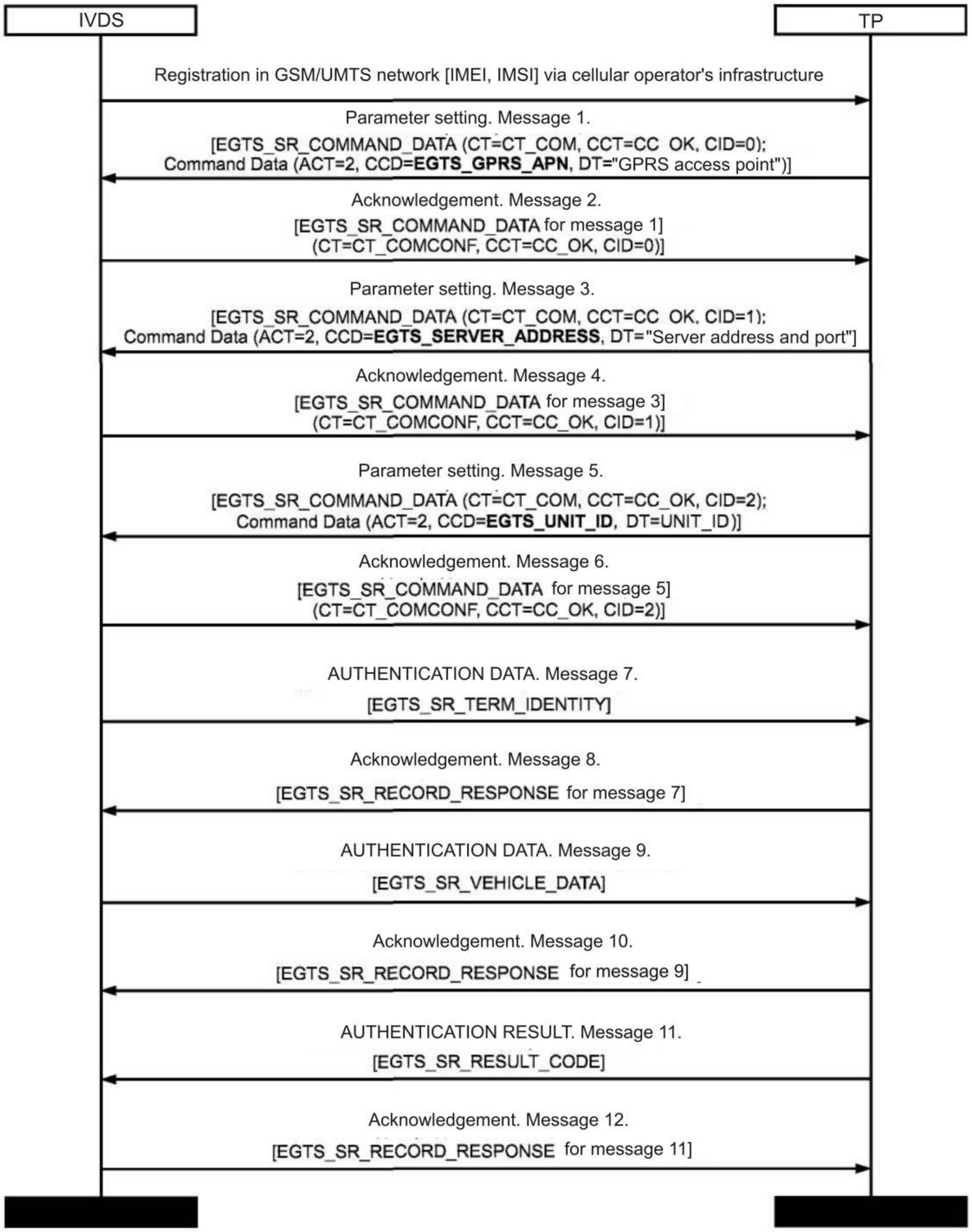

IVDS
TP
Registration in GSM/UMTS network [IMEI, IMSI] via cellular operator's infrastructure
Parameter setting. Message 1.
[EGTS_SR_COMMAND_DATA (CT=CT_COM, CCT=CC OK, CID=0);
Command Data (ACT=2, CCD=EGTS_GPRS_APN, DT="GPRS access point")]
Acknowledgement. Message 2.
[EGTS_SR_COMMAND_DATA for message 1]
(CT=CT_COMCONF, CCT=CC_OK, CID=0)]
Parameter setting. Message 3.
[EGTS_SR_COMMAND_DATA (CT=CT_COM, CCT=CC OK. CID=1);
Command Data (ACT=2, CCD=EGTS_SERVER_ADDRESS, DT="Server address and port"]
Acknowledgement. Message 4.
[EGTS_SR_COMMAND_DATA for message 3]
(CT=CT_COMCONF, CCT=CC_OK, CID=1)]
Parameter setting. Message 5.
[EGTS_SR_COMMAND_DATA (CT=CT_COM, CCT=CC_OK, CID=2);
Command Data (ACT=2, CCD=EGTS_UNIT_ID, DT=UNIT_ID)]
Acknowledgement. Message 6.
[EGTS_SR_COMMAND_DATA for message 5]
(CT=CT_COMCONF, CCT=CC_OK, CID=2)]
AUTHENTICATION DATA. Message 7.
[EGTS_SR_TERM_IDENTITY]
Acknowledgement. Message 8.
[EGTS_SR_RECORD_RESPONSE for message 7]
AUTHENTICATION DATA. Message 9.
[EGTS_SR_VEHICLE_DATA]
Acknowledgement. Message 10.
[EGTS_SR_RECORD_RESPONSE for message 9]
AUTHENTICATION RESULT. Message 11.
[EGTS_SR_RESULT_CODE]
Acknowledgement. Message 12.
[EGTS_SR_RECORD_RESPONSE for message 11]

Figure 8 — Algorithm of IVDS configuration using SMS

2) After the IVDS registration in the GSM or UMTS network is complete, the GPRS session is started and the TCP/IP connection is established with the server which address data are already stored in IVDS memory. During the authentication procedure that follows, the System Operator's infrastructure examines the TID parameter in the EGTS_SR_TERM_IDENTITY sub-record (Table 17). If the TID value is zero, the configuration proceeds by IVDS parameter settings using the EGTS_SR_COMAND_DATA parameter of EGTS_COMMANDS_SERVICE via SMS, as described for the method above.

After the AC EGTS_UNIT_ID parameter setting is complete, an authorisation result is sent to the IVDS with the code EGTS_PC_ID_NFOUND indicating that TID = 0 is not found in the system. The server does not break the connection with the IVDS waiting for the IVDS to perform authorisation again, now with the correct TID parameter. The algorithm of such configuration method is illustrated in Figure 9.

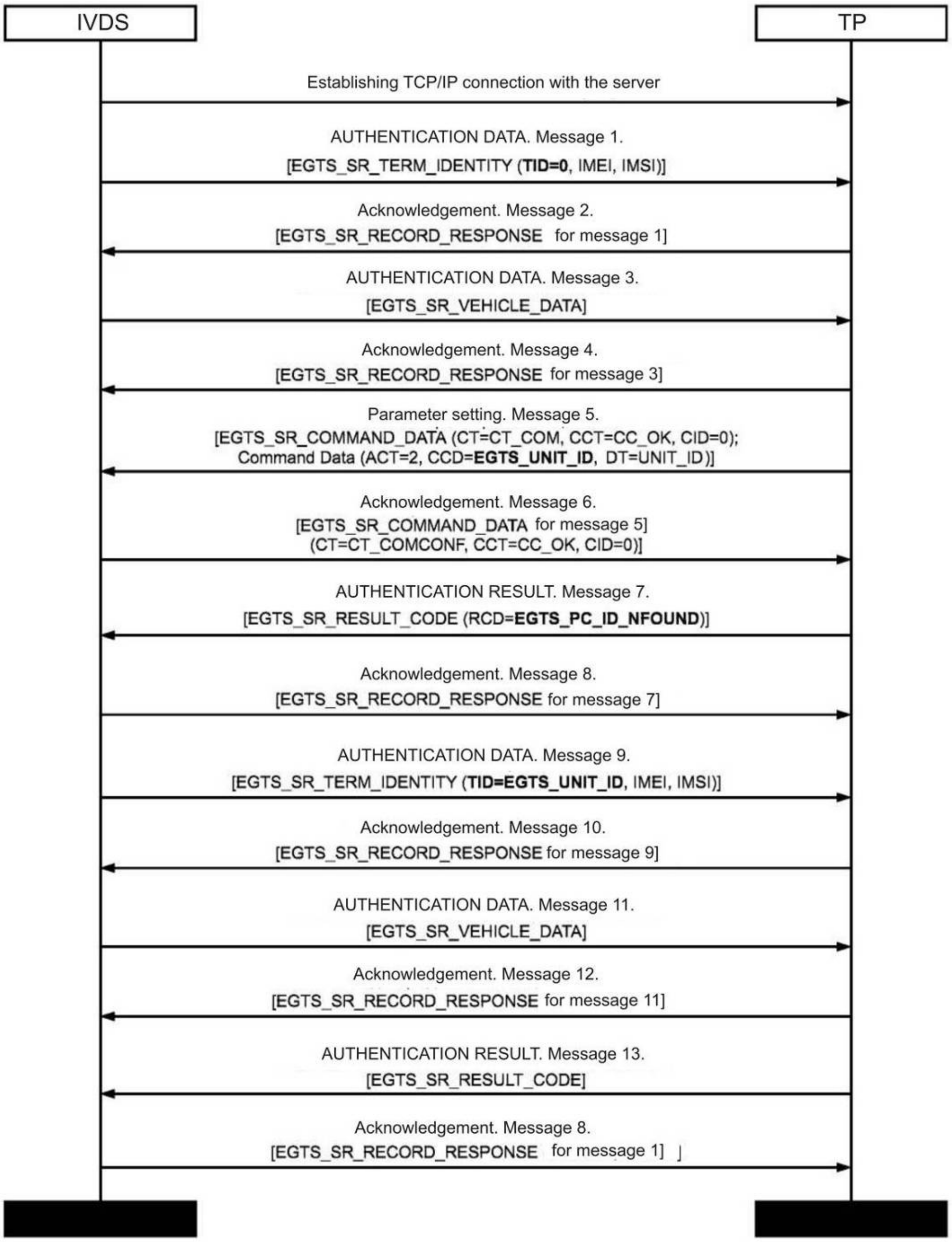

IVDS
TP
Establishing TCP/IP connection with the server
AUTHENTICATION DATA. Message 1.
[EGTS_SR_TERM_IDENTITY (TID=0, IMEI, IMSI)]
Acknowledgement. Message 2.
[EGTS_SR_RECORD_RESPONSE for message 1]
AUTHENTICATION DATA. Message 3.
[EGTS_SR_VEHICLE_DATA]
Acknowledgement. Message 4.
[EGTS_SR_RECORD_RESPONSE for message 3]
Parameter setting. Message 5.
[EGTS_SR_COMMAND_DATA (CT=CT_COM, CCT=CC_OK, CID=0);
Command Data (ACT=2, CCD=EGTS_UNIT_ID, DT=UNIT_ID)]
Acknowledgement. Message 6.
[EGTS_SR_COMMAND_DATA for message 5]
(CT=CT_COMCONF, CCT=CC_OK, CID=0)]
AUTHENTICATION RESULT. Message 7.
[EGTS_SR_RESULT_CODE (RCD=EGTS_PC_ID_NFOUND)]
Acknowledgement. Message 8.
[EGTS_SR_RECORD_RESPONSE for message 7]
AUTHENTICATION DATA. Message 9.
[EGTS_SR_TERM_IDENTITY (TID=EGTS_UNIT_ID, IMEI, IMSI)]
Acknowledgement. Message 10.
[EGTS_SR_RECORD_RESPONSE for message 9]
AUTHENTICATION DATA. Message 11.
[EGTS_SR_VEHICLE_DATA]
Acknowledgement. Message 12.
[EGTS_SR_RECORD_RESPONSE for message 11]
AUTHENTICATION RESULT. Message 13.
[EGTS_SR_RESULT_CODE]
Acknowledgement. Message 8.
[EGTS_SR_RECORD_RESPONSE for message 1]]

Fig. 9 — Algorithm of IVDS configuration using GPRS

If the authorisation succeeds, the telematic platform may precede the EGTS_SR_RESULT_CODE sub-record by the EGTS_SR_SERVICE_INFO sub-records indicating the services accessible for the IVS and supported by the platform, as appropriate for the algorithm based on requests for services.

This means that the listed services are the only ones that may be used by the IVDS right after authorisation, even if the use of the simple algorithm for service access is presumed.

If the algorithm based on requests for services is used, the IVDS may not employ any services if the permission for their use has not been received from the telematic platform. In fact, the permission for some requested services may be issued later, for example, when the services are on the remote telematic platforms that respond the queries asynchronously. In such case, the telematic platform will use the available routing data to send an asynchronous request for services to the remote platform provided that the HDID parameter was included in the EGTS_SR_TERM_IDENTITY sub-record in the process of IVS authorisation.

The message exchange algorithm used at the stage of IVDS authorisation on the telematic platform is illustrated on the flowchart presented in Figure 10.

The IVDS shall be authorised after its successful TCP/IP connection to the telematic platform, and shall send a message with the EGTS_SR_TERM_IDENTITY sub-record (Message 1) before the time period EGTS_SL_NOT_AUTH_TO expires, in order to transmit its initial authentication data.

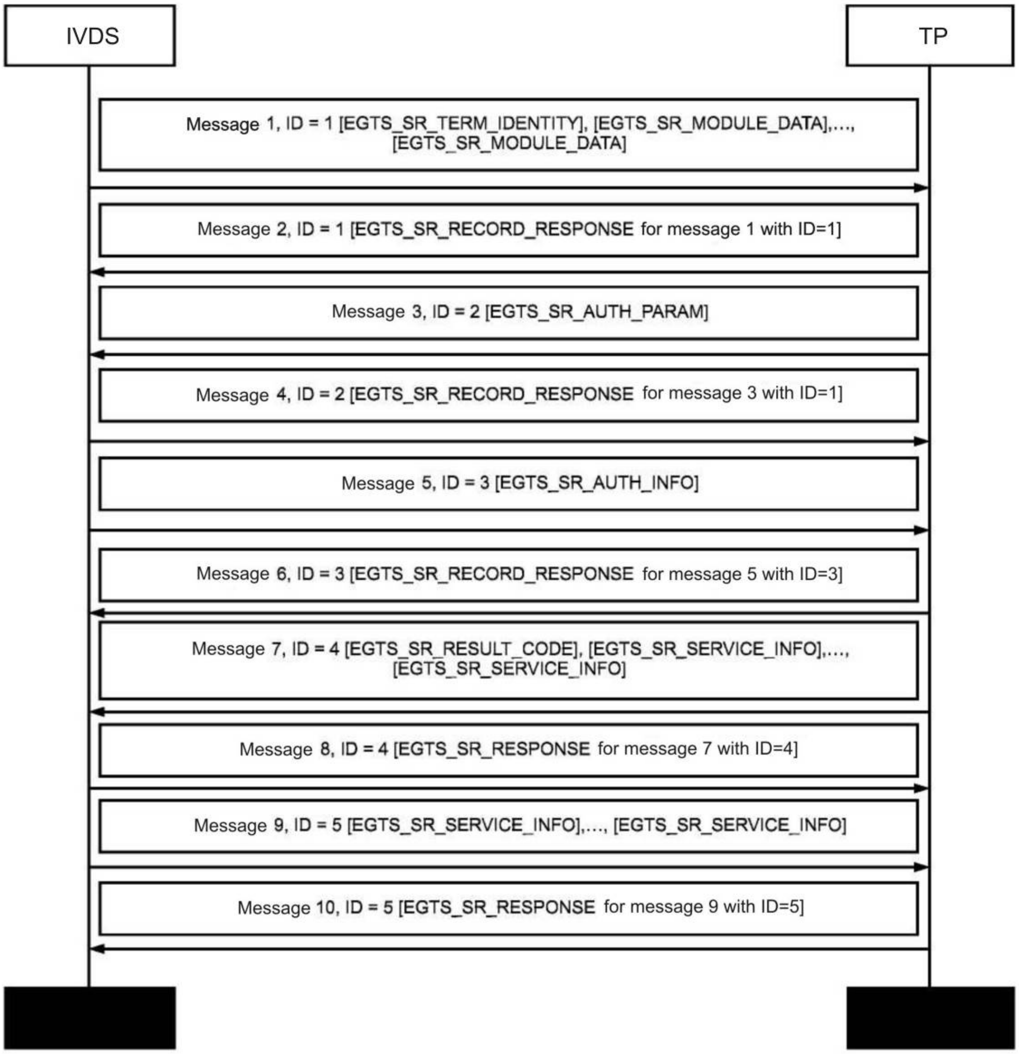

IVDS
TP
Message 1, ID = 1 [EGTS_SR_TERM_IDENTITY], [EGTS_SR_MODULE_DATA],..., [EGTS_SR_MODULE_DATA]
Message 2, ID = 1 [EGTS_SR_RECORD_RESPONSE for message 1 with ID=1]
Message 3, ID = 2 [EGTS_SR_AUTH_PARAM]
Message 4, ID = 2 [EGTS_SR_RECORD_RESPONSE for message 3 with ID=1]
Message 5, ID = 3 [EGTS_SR_AUTH_INFO]
Message 6, ID = 3 [EGTS_SR_RECORD_RESPONSE for message 5 with ID=3]
Message 7, ID = 4 [EGTS_SR_RESULT_CODE], [EGTS_SR_SERVICE_INFO],..., [EGTS_SR_SERVICE_INFO]
Message 8, ID = 4 [EGTS_SR_RESPONSE for message 7 with ID=4]
Message 9, ID = 5 [EGTS_SR_SERVICE_INFO],..., [EGTS_SR_SERVICE_INFO]
Message 10, ID = 5 [EGTS_SR_RESPONSE for message 9 with ID=5]

Fig. 10 — Message exchange during IVDS authorisation on telematic platform

After the EGTS_SR_TERM_IDENTITY sub-record is received by the telematic platform, the platform replies with Message 2 containing the acknowledgement EGTS_SR_RECORD_RESPONSE for the record with ID=1. Then, depending on the telematic platform settings (regarding the encryption or additional authorisation algorithm) it sends a packet (Message 3) with the EGTS_SR_AUTH_PARAM sub-record containing the parameters required for encryption/extended authorisation. If encryption and extended authorisation are not used, then the telematic platform may send the EGTS_SR_RESULT_CODE sub-record with the IVDS authorisation result instead of the EGTS_SR_AUTH_PARAM sub-record.

Then, the IVDS sends Message 4 with EGTS_SR_RECORD_RESPONSE acknowledgement for Message 3 with ID=2. If encryption and/or extended authorisation are in use, the IVDS replies with Message 5 which is encrypted as specified in Message 3 from the telematic platform and which contains the EGTS_SR_AUTH_INFO sub-record with the data required for extended authorisation.

After the EGTS_SR_AUTH_INFO sub-record is received by the telematic platform, the platform sends Message 6 to acknowledge Message 5 with ID=3, and carries out the authorisation procedure. The platform generates Message 7 with authorisation results in its EGTS_SR_RESULT_CODE sub-record which, in case of successful authorisation, may be completed with EGTS_SR_SERVICE_INFO sub-records describing the services permitted for use by this IVDS.

Then, the IVDS generates Message 8 to acknowledge Message 7 with ID=4. The IVDS may generate Message 9 and add the EGTS_SR_SERVICE_INFO sub-records with information on any requested services (if the procedure based on requests for services is used) and/or on the services supported on the IVS side.

The telematic platform then acknowledges Message 9 with ID=5 by sending Message 10.

This completes the authorisation stage, and the IVDS switches to message exchange with the platform in accordance with the operating mode active in the IVDS.

Should the authorisation procedure fail (due to incorrect IVDS authentication data, denied access of this IVDS to the telematic platform, etc.), the telematic platform shall break the TCP/IP connection with the IVDS after sending the message with the EGTS_SR_RESULT_CODE sub-record indicating the relevant code.

6.7.3 EGTS_COMMANDS_SERVICE

This service type is intended for processing of commands and acknowledgements transferred between the IVDS, telematic platform, and client applications.

The only sub-record required for interaction using this service is EGTS_SR_COMMAND_DATA described in Table 28.

Table 28 — Description of sub-records attributed to EGTS_COMMAND_SERVICE

Code	Name	Description
0	EGTS_SR_RECORD_RESPONSE	Used to acknowledge the processing of Service Support Layer Protocol records. This sub-record type shall be supported by all services.
51	EGTS_SR_COMMAND_DATA	Used by the IVDS and telematic platform to transfer commands, information messages, acknowledgements of command delivery and completion, and acknowledgements of message processing.

6.7.3.1 EGTS_SR_COMMAND_DATA sub-record.

The sub-record format is specified in Table 29.

Table 29 — Format of EGTS_SR_COMMAND_DATA sub-record for EGTS_COMMANDS_SERVICE

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Type	Data type	Size in bytes
CT (Command Type)				CCT (Command Confirmation Type)				M	BYTE	1
CID (Command Identifier)								M	UINT	4
SID (Source Identifier)								M	UINT	4
—						ACFE	CHSFE	M	BYTE	1
CHS (Charset)								O	BYTE	1
ACL (Authorisation Code Length)								O	BYTE	1
AC (Authorisation Code)								O	BINARY	0...255
CD (Command Data)								O	BINARY	0...65205

The parameters (fields) of the EGTS_SR_COMMAND_DATA sub-record listed in Table 29 have the following meaning:

- CT — command type:

a) 0001 — CT_COMCONF — acknowledgement of command reception or processing, or command execution result;

b) 0010 — CT_MSGCONF — acknowledgement of message reception, presentation and/or processing;

c) 0011 — CT_MSGFROM — information message from IVDS;

d) 0100 — CT_MSGTO — information message for output to vehicle display device;

e) 0101 — CT_COM — command to be executed on the vehicle;

f) 0110 — CT_DELCOM — deleting the previous command from the execution queue;

g) 0111 — CT_SUBREQ — additional sub-request for execution (for a command sent before);

h) 1000 — CT_DELIV — acknowledgement of command/message delivery;

- CCT — type of acknowledgement (makes sense for CT_COMCONF, CT_MSGCONF, or CT_DELIV commands):

a) 0000 — CC_OK — successful completion, positive response;

b) 0001 — CC_ERROR — processing failed;

c) 0010 — CC_ILL — command can not be executed because it is missing in the list of permitted commands (defined by the protocol), or because of insufficient rights to execute it;

d) 0011 — CC_DEL — command deleted successfully;

e) 0100 — CC_NFOUND — command to be deleted is not found;

f) 0101 — CC_NCONF — successful execution, negative response;

g) 0110 — CC_INPROG — command dispatched for processing but will require a long time to execute (execution result is not known yet);

- CID — command/message identifier. The value of this field shall be used by the side processing or executing the command or message in order to generate an acknowledgement with the field CID containing the same value as the one included in the command or message itself upon its delivery;

- SID — command/acknowledgement sender identifier. If an acknowledgement is sent from the IVDS for a command or command execution result (CT_COMCONF, CT_MSGCONF or CT_DELIV command types), the value of this field must be copied from the command earlier transmitted to the IVDS. When the transmission of the EGTS_SR_COMMAN_DDATA sub-record is initiated on the IVDS side, this field is set to zero;

- ACFE — authorisation code field exists. Bit flag indicating whether the ACL and IVDS fields are present in the sub-record:

a) 1 — ACL and IVDS fields are present in the sub-record;

b) 0 — ACL and IVDS fields are not present in the sub-record;

- CHSFE — Charset Field Exists. Bit flag indicating if the CHS field present in the sub-record:

a) 1 — CHS field is present in the sub-record;

b) 0 — CHS field is not present in the sub-record;

- CHS — character encoding used in the CD field that contains the command body. If this field is missing, the CP-1251 shall be used by default. The following CHS field values are defined (decimal):

a) 0 —CP-1251;

b) 1 — IA5 (CCITT T.50)/ASCII (ANSI X3.4);

c) 2 — binary data;

d) 3 — Latin 1 (Figure F.1, Appendix F);

e) 4 — binary data;

f) 5 — JIS(X 0208-1990);

g) 6 — Cyrillic (Figure F.2, Appendix F);

h) 7 — Latin/Hebrew (Figure F.3, Appendix F);

i) 8 — UCS2.

- ACL — number of bytes in the AC field that contains the authorisation code on the recipient side;
- AC — authorisation code used on the recipient side (IVDS) to restrict the rights for execution of individual commands. If the code in this field and the expected value do not match, the IVDS shall send CC_ILL in response to such command or message. The EGTS_SET_AUTH_CODE command is used on the IVDS side to set the authorization code;
- CD — command body, parameters and data returned in reply to the query command, their encoding being either defined by the CHS field value, or set to the default one.

The size of this field is determined by the total length of the Service Support Layer Protocol record and of the fields preceding it in this sub-record. The command format is specified in Table 30. This field may have zero length (not present) when no data for the IVDS are transmitted in reply to a command or a message.

Table 30 — IVDS command format

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Type	Data type	Size in bytes
ADR (Address)								M	USHORT	2
SZ (Size)				ACT (Action)				M	BYTE	1
CCD (Command Code)								M	USHORT	2
DT (Data)								O	BINARY	0...65200

The parameters listed in Table 30 have the following meaning:

- ADR — address of the module this command is intended for. The address is determined by the initial IVDS configuration, or from a module list which may be obtained in the EGTS_SR_MODULE_DATA sub-records transmitted when the IVDS registers through the EGTS_AUTH_SERVICE;
- SZ — memory size for the parameter (used in combination with ACT-2). When a new parameter is added to the IVDS, this field informs that 2^{SZ} bytes of IVDS memory are required for such parameter;
- ACT — description of action; used when the CT field of the EGTS_SR_COMMAND_DATA sub-record indicates the command type CT_COM. The following values of ACT are permitted:

a) 0 — command parameters. Used to send parameters for a command defined by the CCD field;

b) 1 — query value. Used to query data stored in the IVDS. The requested parameter is defined by the code in the CCD field;

c) 2 — set value. Used to set a new value for a certain parameter in the IVDS. The parameter to be set is defined by the code in the CCD field, and its value, by the DT field;

d) 3 — add new parameter to IVDS. The new parameter code is specified in the CCD field, and its value, in the DT field;

e) 4 — remove existing parameter from IVDS. The parameter code is specified in the CCD field;

- CCD — command code for ACT-0, or parameter code for ACT-1...4;
- DT — requested data or parameters required to execute the command. The format of data written to this field depends on the command type.

If the IVDS provides the data associated with the previous command containing CT_COMCONF in the CT field, the format of their acknowledgement shall correspond to Table 31. The structure described therein shall be included in the CD field (Table 29).

Table 31 — Format of IVDS command acknowledgement

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Type	Data type	Size in bytes
ADR (Address)								M	USHORT	2
CCD (Command Code)								M	USHORT	2
DT (Data)								O	BINARY	0...65200

The parameters listed in Table 31 have the following meaning:

- ADR — address of the target module for this command. The address is determined by the initial IVDS configuration or from a module list which may be obtained in the EGTS_SR_MODULE_DATA subrecords transmitted when the IVDS registers through the EGTS_AUTH_SERVICE. This field shall be always zero in commands EGTS_ECALL_REQ and EGTS_ECALL_MSD_REQ from System Operator;
- CCD — command/message code as per Table 32 or parameter code as per Table 34, defining any associated data to be transmitted in the DT field;
- DT — associated data of the type and contents defined by the CCD field value. The list and contents of associated data sent in acknowledgements to certain commands are specified in Table 33.

6.7.3.2 Description of commands, parameters and acknowledgements

The list and description of IVDS commands are given in Table 32, the list of acknowledgements for IVDS commands and messages, in Table 33, and the list of IVDS parameters, in Table 34.

The values of the following IVDS parameters may be queried but can not be changed or removed using the command service: EGTS_UNIT_SERIAL_NUMBER, EGTS_UNIT_HW_VERSION, EGTS_UNIT_SW_VERSION, EGTS_UNIT_VENDOR_ID, and EGTS_UNIT_IMEI.

The values of the above parameters are set by the manufacturers of individual IVDS modules and units, as well as by the developers of the respective software.

Table 32 — List of commands sent to IVDS

Command name	Code	Type, number and limiting parameter values	Description
EGTS_RAW_DATA	0x0000	BINARY (up to 65200 bytes)	Used to send arbitrary data, e.g., when any commands, messages or data are transmitted to peripheral devices or modules connected to the main IVDS unit in a module-specific format. In this case, the IVDS shall not parse the DT field data and shall pass them as is to the address specified in the ADR field.
EGTS_TEST_MODE	0x0001	BYTE	Command used to start/stop IVDS test: 1 — stat test, 0 — stop test
EGTS_CONFIG_RESET	0x0006		Restore factory settings. All parameters set by users are removed and reset to factory settings. To process this command, the operator shall write proper values to the ACL and AC fields described in Table 29.
EGTS_SET_AUTH_CODE	0x0007	BINARY	Set authorisation code on the IVDS side. To process this command, the operator shall write proper values to the ACL and AC fields described in Table 29. After this command is acknowledged, the IVDS will use the updated data to compare certain command it receives with the AC field value.
EGTS_RESTART	0x0008		Command used to restart the main IVDS software. To process this command, the operator shall write proper values to the ACL and AC fields described in Table 29.

Table 33 — List of acknowledgements for commands and messages from IVS

Command name	Code	Type and number of parameters	Description
EGTS_RAW_DATA	0x0000	BINARY (up to 65200 bytes)	Data received from peripheral devices and modules connected to the main IVS unit, in module-specific format

Table 34 — List of IVDS parameters

Parameter name	Code	Parameter type	Default value	Description	Applicability[1)]	Possible change[2}]
Radio mute						
EGTS_RADIO_MUTE_DELAY	0x0201	INT	0	Delay from activation of radio mute signal to start of sound playback, ms	AUX	Yes
EGTS_RADIO_UNMUTE_DELAY	0x0202	INT	0	Delay from deactivation of radio mute signal to termination of sound playback, ms	AUX	Yes
General-purpose settings						
EGTS_GPRS_APN	0x0203	STRING	" "	Specification of GPRS access point	AUX, STD+	Yes
EGTS_SERVER_ADDRESS	0x0204	STRING	" "	Server address and port for TCP/IP communication	AUX, STD+	Yes
EGTS_SIM_PIN	0x0205	INT	0	PIN code of SIM card	AUX, STD,	Yes
EGTS_INT_MEM_TRANSMIT_INTERVAL	0x0206	INT	60	Time interval between the repeated attempts to send a message if its packet or SMS transmission failed, min	AUX, STD, STD+	Yes
EGTS_INT_MEM_TRANSMIT_ATTEMPTS	0x0207	INT	10	Maximum number of retries of packet or SMS transmission in case of failures	AUX, STD, STD+	Yes
Test mode						
EGTS_TEST_REGISTRATION_PERIOD	0x0242	INT	5	If the IVDS has been registered in a network using the button for activation of additional services, then any later network registration of the IVDS using this button is not possible until this interval expires. If set to zero, no restrictions on later network registration of the IVDS are imposed. Set in minutes.	AUX, STD, STD+	Yes

Table 34 (continued)

Parameter name	Code	Parameter type	Default value	Description	Applicability[1)]	Possible change[2}]
EGTS_TEST_MODE_END_DISTANCE	0x020A	INT	300	Distance where Test mode is switched off automatically, m	AUX, STD, STD+	Yes
Service Station mode						
EGTS_GARAGE_MODE_END_DISTANCE	0x020B	INT	300	Distance where Service Station mode is switched off automatically, m	AUX	Yes
EGTS_GARAGE_MODE_PIN	0x020C	INT/0...8	0	Line signalling that the IVDS is in Service Station mode. NONE — Service Station mode is not active; X — PIN_X line is active when IVDS is in this mode	AUX	Yes
Miscellaneous parameters						
EGTS_GNSS_POWER_OFF_TIME	0x0301	INT	500	Time interval to expire from ignition turning-off to power disconnection of the GNSS receiver, in milliseconds	AUX	Yes
EGTS_GNSS_DATA_RATE	0x0302	INT/1,2,5, 10	Defined by IVDS manufacturer	Data output rate of GNSS receiver, Hz	AUX, STD, STD+	No
EGTS_GNSS_MIN_ELEVATION	0x0303	INT/5...15	15	Minimum elevation (cut-off angle) of navigation spacecrafts, deg	AUX, STD, STD+	No
Device parameters						
EGTS_UNIT_ID	0x0404	INT	0	Unique IVDS identifier assigned by System Operator at first authorisation	AUX, STD, STD+	Yes
EGTS_UNIT_IMEI	0x0405	STRING	,,,,	IMEI number	AUX, STD,	No
EGTS_UNIT_RS485_BAUD_RATE	0x0406	INT	19200	RS485 port baud rate, bit/s	AUX, STD,	Yes
EGTS_UNIT_RS485_STOP_BITS	0x0407	INT	1	Number of stop bits for data transmission via RS485 port	AUX, STD, STD+	Yes

Table 34 (continued)

Parameter name	Code	Parameter type	Default value	Description	Applicability[1)]	Possible change[2}]

EGTS_UNIT_RS485_PARITY	0x0408	INT/0,1,2	0	Parity check for transmission via RS485 port:
				0 — no parity check;
				1 — ODD parity;
				2 — EVEN parity
					AUX, STD, STD+	Yes

EGTS_UNIT_HOME_DISPATCHER_ID	0x0411	INT	0	ID of telematic platform where device accounting data, list of provided services and their statuses are stored	AUX, STD, STD+	Yes
EGTS_SERVICE_AUTH_METHOD	0x0412	INT	1	Service usage method: 1 — simple (IVDS may access all services by default); 0 — with acknowledgement (IVDS may use only those services that were reported by the telematic platform as permitted for use)	AUX, STD, STD+	Yes
EGTS_SERVER_CHECK_IN_PERIOD	0x0413	INT	30	Time between attempts to re-establish TCP/IP connection with server, s	AUX, STD+	Yes
EGTS_SERVER_CHECK_IN_ATTEMPTS	0x0414	INT	5	Number of attempts to establish TCP/IP connection with the server; when exceeded, an attempt to re-establish a high level (GPRS) session will be made	AUX, STD+	Yes
EGTS_SERVER_PACKET_TOUT	0x0415	INT	5	Time IVDS is waiting for server's acknowledgement of a previously sent packet, s	AUX, STD+	Yes
EGTS SERVER PACKET RETRANSMIT ATTEMPTS	0x0416	INT	3	Number of attempts to re-send an unacknowledged packet; when exceeded, the IVDS re-initiates the session at the TCP/IP level	AUX, STD+	Yes
EGTS_UNIT_MIC_LEVEL	0x0417	INT/0... 10	8	Microphone sensitivity level	AUX, STD,	Yes
EGTS_UNIT_SPK_LEVEL	0x0418	INT/0... 10	6	Speaker volume level	AUX, STD,	Yes

Table 34 (continued)

[1] "AUX" — for IVDS in auxiliary equipment configuration; "STD" — for IVDS in standard equipment configuration that is only intended for the Base Service of the System; "STD+" for IVDS in standard equipment configuration that is also intended for other System services in addition to the Base Service.

[2] "Yes" — means that the specified initial value of the IVDS parameter may change after the initial IVDS setup, "No" that the initial settings are not subject to changes while the IVDS is used.

IVDS systems installed in standard equipment configuration shall support the following parameters:
- EGTS_GPRS_APN;
- EGTS_SERVER_ADDRESS;
- EGTS_SIM_PIN;
- EGTS_AUTOMATIC_REGISTRATION;
- EGTS_TEST_MODE_END_DISTANCE;
- EGTS_GARAGE_MODE_END_DISTANCE;
- EGTS_GPRS_WHITE_LIST;
- EGTS_TEST_REGISTRATION_PERIOD;
- EGTS_GNSS_POWER_OFF_TIME;
- EGTS_GNSS_DATA_RATE;
- EGTS_GNSS_MIN_ELEVATION;
- EGTS_UNIT_ID;
- EGTS_UNIT_LANGUAGE_ID;
- EGTS_UNIT_IMEI;
- EGTS_UNIT_HOME_DISPATCHER_ID;
- EGTS_INT_MEM_TRANSMIT_INTERVAL;
- EGTS_INT_MEM_TRANSMIT_ATTEMPTS

6.7.3.3 Examples of command transmission procedures are given in Figures 11 and 12.

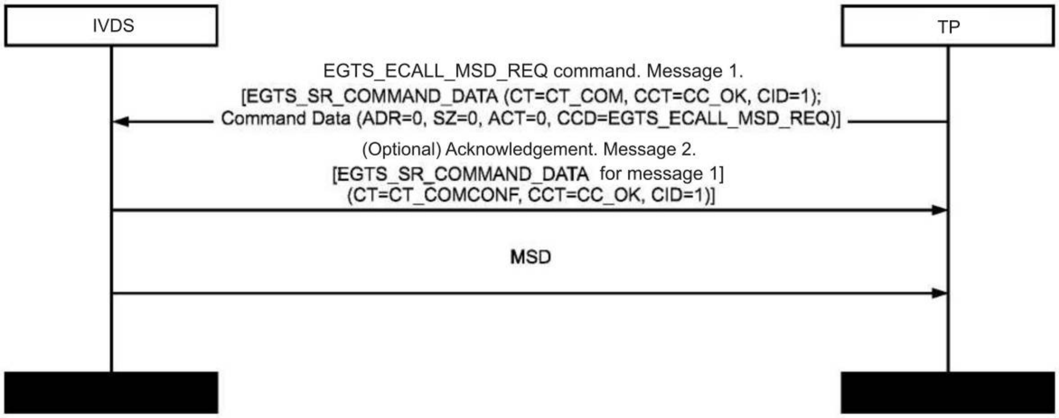

IVDS
TP
EGTS_ECALL_MSD_REQ command. Message 1.
[EGTS_SR_COMMAND_DATA (CT=CT_COM, CCT=CC_OK, CID=1);
Command Data (ADR=0, SZ=0, ACT=0, CCD=EGTS_ECALL_MSD_REQ)]
(Optional) Acknowledgement. Message 2.
[EGTS_SR_COMMAND_DATA for message 1]
(CT=CT_COMCONF, CCT=CC_OK, CID=1)]
MSD

Fig. 11 — Sending EGTS_ECALL_MSD_REQ command via SMS

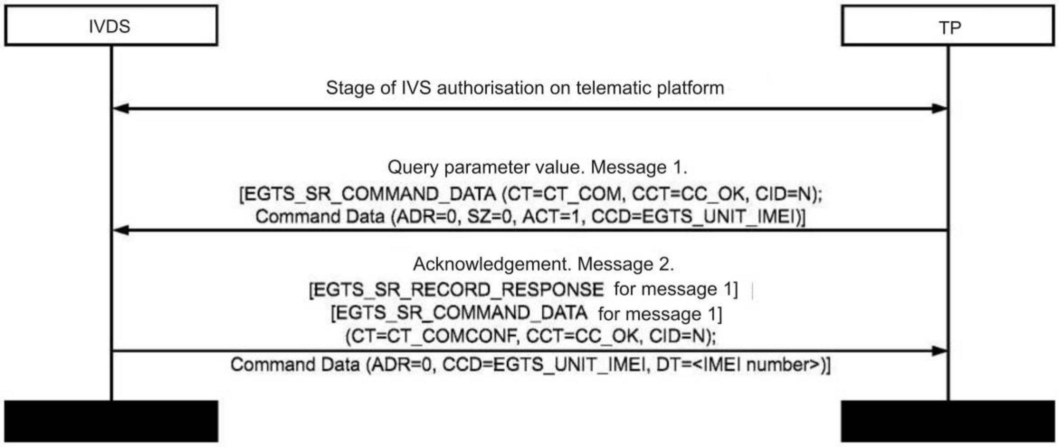

IVDS
TP
Stage of IVS authorisation on telematic platform
Query parameter value. Message 1.
[EGTS_SR_COMMAND_DATA (CT=CT_COM, CCT=CC_OK, CID=N);
Command Data (ADR=0, SZ=0, ACT=1, CCD=EGTS_UNIT_IMEI)]
Acknowledgement. Message 2.
[EGTS_SR_RECORD_RESPONSE for message 1]
[EGTS_SR_COMMAND_DATA for message 1]
(CT=CT_COMCONF, CCT=CC_OK, CID=N);
Command Data (ADR=0, CCD=EGTS_UNIT_IMEI, DT=<IMEI number>)]

Fig. 12 — Query parameter value

6.7.4 EGTS_FIRMWARE_SERVICE

This service is intended for downloading data to the IVDS, including configuration data and firmware updates for modules and units of the IVDS itself as well as of its connected peripheral equipment.

As regards this service, several sub-records are used for interaction; the relevant descriptions and codes of such sub-records are presented in Table 35.

Table 35 — List of sub-records attributed to EGTS_FIRMWARE_SERVICE

Code	Name	Description
0	EGTS_SR_RECORD_ RESPONSE	Used to acknowledge Service Support Layer Protocol records included in packets of the type EGTS_PT_APPDATA
33	EGTS_SR_SERVICE_ PART_DATA	Intended for sending data split into parts and transmitted to the IVDS one by one. This sub-record is used to transfer large objects that can not be sent to the IVDS in a single packet due to their size.
34	EGTS_SR_SERVICE_ FULL_DATA	Used to transfer data in a single packet rather than by parts to the IVDS.

6.7.4.1 EGTS_SR_SERVICE_PART_DATA sub-record

The EGTS_SR_SERVICE_PART_DATA sub-record may be used by a service to send individual objects to the IVDS. The sub-record format is specified in Table 36.

Table 36 — Format of EGTS_SR_SERVICE_PART_DATA sub-record used for EGTS_FIRMWARE_SERVICE

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Type	Data type	Size in bytes
ID (Identity)								M	USHORT	2
PN (Part Number)								M	USHORT	2
EPQ (Expected Parts Quantity)								M	USHORT	2
ODH (Object Data Header)								O	BINARY	0...71
OD (Object Data)								M	BINARY	1...65400

Notes

1 ID — unique identifier of the object being transmitted. It is incremented by 1 each time the transmission of a new object is started. This parameter provides for unique identification of the object which a given part belongs to.

2 PN — sequential number of the object part currently transmitted.

3 EPQ — expected number of object parts to be transmitted.

4 ODH — header with parameters that describe the object being transmitted. This header is sent for the first object part only, but not for any parts that follow it. The ODH header format is shown in Table 37.

5 OD — object data themselves.

The EPQ parameter identifies the number of parts to be transmitted, and the PN parameter identifies the part being transmitted. The field ID uniquely identifies the object this part belongs to. The EPQ and PN values may be in the range from 1 to 65535, and the latter value may not exceed the former one; the sub-record data are ignored otherwise.

The object ID, the PN and EPQ fields and the record source identifier OID from the service routing header may be used to identify which parts of which objects are received for processing. If the channel throughput is high enough, this allows transmitting objects simultaneously when the software of various IVDS units or peripheral equipment needs be updated.

The object header format of this sub-record is shown in Table 37.

Table 37 — Object header format in EGTS_SR_SERVICE_PART_DATA sub-records for EGTS_FIRMWARE_SERVICE

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Type	Data type	Size in bytes
OA (Object Attribute)								M	BYTE	1
—				OT (Object Type)		MT (Module Type)				
CMI (Component or Module Identifier)								M	BYTE	1
VER (Version)								M	USHORT	2
WOS (Whole Object Signature)								M	USHORT	2
FN (File Name)								O	STRING	0...64
D (Delimiter)								M	BYTE	1

The parameters (fields) listed in Table 37 have the following meaning:

-OA — attribute identifying where the object being transmitted belongs;

- OT — object type defining its contents. The following values are defined for this field:

a) 00 — firmware data;

b) 01 — configuration parameter block;

- MT — type of module this object is intended for. The following values are defined for this field:

a) 00 — peripheral equipment;

b) 01 — IVDS;

- CMI — component number if the object belongs to the IVDS itself, or identifier of the peripheral module/port connected to the IVDS, depending on the MT value;

- VER— object version (its high byte defines the major version specified before a decimal point, and low byte defines the minor version specified after it, e.g., 2.34 is represented by 0x0222);

- WOS — signature (checksum) of the whole object. The CRC16-CCITT algorithm is used to calculate it;

- FN — object file name (this field is optional, and may have zero length);

- D — delimiter of string parameters (always 0).

6.7.4.2 EGTS_SR_SERVICE_FULL_DATA sub-record

The sub-record format is specified in Table 38.

Table 38 — Format of EGTS_SR_SERVICE_FULL_DATA sub-record used in EGTS_FIRMWARE_SERVICE

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Type	Data type	Size in bytes
ODH (Object Data Header)								M	BINARY	7...71
OD (Object Data)								M	BINARY	1...65400

The parameters (fields) listed in Table 38 have the following meaning:

- ODH — header with parameters that describe the object being transmitted. This parameter is mandatory for EGTS_SR_SERVICE_FULL_DATA, and must be present in each such sub-record;
- OD — object data themselves.

6.7.4.3 EGTS_SR_RECORD_RESPONSE sub-record

This sub-record has the same structure as the one detailed in 6.7.2.1, and is used to acknowledge that the EGTS_SR_SERVICE_PART_DATA and EGTS_SR_SERVICE_FULL_DATA records have been received or processed. All EGTS_SR_SERVICE_PART_DATA sub-records except for the last one shall be responded with the EGTS_PC_IN_PROGRESS result code in the EGTS_SR_RECORD_RESPONSE sub-record if their processing was successful. As to the last EGTS_SR_SERVICE_PART_DATA sub-record and each EGTS_SR_SERVICE_FULL_DATA one, the returned EGTS_SR_RECORD_RESPONSE sub-record shall contain the EGTS_PC_OK code in case of successful reception and processing, to be interpreted by the service as a successful attempt to deliver the whole object.

6.8 Timing and numerical parameters of Service Support Layer Protocol in case of packet data transmission

The timing and numerical parameters used in the Service Support Layer Protocol are described in Table 39.

Table 39 — Timing and numerical parameters of Service Support Layer Protocol

Name	Data type	Range of values	Default value	Description
EGTS_SL_NOT_ AUTH_TO	BYTE	0...255	6	Time to wait for an IVDS message containing the data required for IVDS authorisation on the telematic platform side after the IVDS establishes a new TCP/IP connection, in seconds. If no message arrives during this time, the platform shall break the TCP/IP connection established with the IVDS.

7 Accident emergency response service of Service Support Layer Protocol

7.1 Purpose of accident emergency response service

The accident emergency response service is intended for implementation of the Base Service provided by the System. In the framework of the Service Support Layer Protocol, this service is called EGTS_ECALL_SERVICE, and its code is 10.

7.2 Minimum set of IVDS functions required for use of EGTS_ECALL_SERVICE

The following set of functions shall be implemented in the IVDS in order to use the EGTS_ECALL_SERVICE:

- support of command processing service EGTS_COMMANDS_SERVICE specified in 6.7.3;
- support of EGTS_ECALL_REQ and EGTS_ECALL_MSD_REQ commands sent by System Operator via SMS, and transmission of replies and acknowledgements for them;

- GPRS transfer of acceleration profile data (EGTS_SR_ACCEL_DATA sub-record);
- GPRS transfer of vehicle movement path data (EGTS_SR_TRACK_DATA sub-record);
- processing IVDS parameter setting commands sent by System Operator via GPRS and SMS, and transmission of replies and acknowledgements for them.

7.3 Structure and description of EGTS_ECALL_SERVICE sub-records

Several sub-records are used for interaction within the framework of EGTS_ECALL_SERVICE; their descriptions and codes are detailed in Table 40.

Table 40 — List of EGTS_ECALL_SERVICE sub-records

Code	Name	Description
0	EGTS_SR_RECORD_ RESPONSE	Used to acknowledge Service Support Layer Protocol records contained in packets of the EGTS_PT_APPDATA type.
20	EGTS_SR_ACCEL_DATA	Used by the IVDS to transfer acceleration profiles to the telematic platform.
40	EGTS_SR_RAW_MSD_DATA	Used by the IVDS to transfer raw MSD data to the telematic platform.
62	EGTS_SR_TRACK_DATA	Used to transfer vehicle movement path data to the telematic platform in case of road accidents.

7.3.1 EGTS_SR_RECORD_RESPONSE sub-record

This sub-record has the same format as the one specified in 6.7.2.1.

7.3.2 EGTS_SR_ACCEL_DATA sub-record

The sub-record format is specified in Table 41.

Table 41 — Format of EGTS_SR_ACCEL_DATA sub-record for EGTS_ECALL_SERVICE

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Type	Data type	Size in bytes
SA (Structures Amount)								M	BYTE	1
ATM (Absolute Time)								M	UINT	4
ADS1 (Accelerometer Data Structure 1)								M	BINARY	8
ADS2 (Accelerometer Data Structure 2)								0	BINARY	8
…								…	…	…
ADS255 (Accelerometer Data Structure 255)								0	BINARY	8

The parameters (fields) listed in Table 41 have the following meaning:

- SA — number of transmitted data structures with accelerometer readings;
- ATM — measurement time of the first accelerometer data structure transmitted (number of seconds since 00:00:00 01.01.2010 UTC);

Note — Several EGTS_SR_ACCEL_DATA sub-records each containing its own start time (in the "ATM" field) may be specified in sequence in order to transmit the required number of accelerometer samples.

- ADS1...ADS255 — accelerometer data structures. The structure format is described in Table 42.

At least one ADS structure shall be transmitted in the EGTS_SR_ACCELDATA sub-record.

Table 42 — Format of accelerometer data structures EGTS_SR_ACCEL_DATA used in EGTS_ECALL_SERVICE

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Type	Data type	Size in bytes
RTM (RelativeTime)								M	USHORT	2
XAAV (X Axis Acceleration Value)								M	SHORT	2
YAAV (Y Axis Acceleration Value)								M	SHORT	2
ZAAV (Z Axis Acceleration Value)								M	SHORT	2

The parameters (fields) listed in Table 42 have the following meaning:

- RTM — measurement time increment with respect to the previous record (or with respect to the ATM field value, if the first record is transmitted), in milliseconds;
- XAAV — linear acceleration along *X* axis, 0.1 m/s^2;
- YAAV — linear acceleration along *Y* axis, 0.1 m/s^2;
- ZAAV — linear acceleration along *Z* axis, 0.1 m/s^2.

The resolution of acceleration field values shall be no worse than 0.01G.

7.3.3 EGTS_SR_RAW_MSD_DATA sub-record

The EGTS_SR_RAW_MSD_DATA sub-record format is described in Table 43.

Table 43 —Format of EGTS_SR_RAW_MSD_DATA sub-record used in EGTS_ECALL_SERVICE

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Type	Data type	Size in bytes
FM (Format)								M	BYTE	1
MSD (Minimum Set of Data)								M	BINARY	0...116

The parameters (fields) listed in Table 43 have the following meaning:

- FM — format of data included in the MSD field of this sub-record. The following values are defined for this field in this version of the document:

a) 0 — format unknown;

b) 1 — packet encoding as per GOST 33464.

Any FM field values not specified in this Standard are subject to additional agreement between the IVDS manufacturer and System Operator;

- MSD — minimum set of data.

7.3.4 EGTS_SR_TRACK_DATA sub-record

The EGTS_SR_TRACK_DATA sub-record format is described in Table 44.

Table 44 — Format of EGTS_SR_ TRACK_DATA sub-record used in EGTS_ECALL_SERVICE

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Type	Data type	Size in bytes
SA (Structures Amount)								M	BYTE	1
ATM (Absolute Time)								M	UINT	4
TDS1 (Track Data Structure 1)								M	BINARY	1…12
TDS2 (Track Data Structure 2)								O	BINARY	1…12
…								…	…	…
TDS 255 (Track Data Structure 255)								O	BINARY	1…12

The parameters (fields) listed in Table 44 have the following meaning:

- SA — number of points in the vehicle movement path being sent;
- ATM — reference time of measurements (number of seconds since 00:00:00 01.01.2010 UTC). This value used as an initial time for the first transmitted structure, with 1 second precision.

A more accurate measurement time is determined taking into account the RTM field containing the data structure for an individual point of the movement path;

- TDS1…TDS255 — data structures containing the parameters for a single point of the vehicle movement path. The structure format is specified in Table 45.

At least one TDS structure shall be transmitted in the EGTS_SR_TRACK_DATA sub-record.

Table 45 — Format of data structure for a single point of vehicle movement path, as defined for EGTS_SR_TRACK_DATA sub-record used in EGTS_ECALL_SERVICE

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Type	Data type	Size in bytes
TNDE	LOHS	LAHS	RTM (Relative Time)					M	BYTE	1
LAT (Latitude)								O	UINT	4
LONG (Longitude)								O	UINT	4
SPDL (Speed Low Bits)								O	USHORT	2
DIRH	SPDH (Speed Hi Bits)									
DIR (Direction)								O	BYTE	1

The parameters (fields) listed in Table 45 have the following meaning:

- TNDE — Track Node Data Exist; bit flag indicating whether the component data regarding the point of the movement path are present in this TDS structure (LAT, LONG, SPDL, DIRH, SPDH and DIR fields):

a) 1 — data present;

b) 0 — data not present (no reliable coordinate and speed data of the required accuracy could be obtained for the specified time moment). The LAT, LONG, SPDL, DIRH, SPDH and DIR fields are not sent in this structure, and its size equals to 1 byte;

- LOHS — bit flag identifying the longitude hemisphere:

a) 0 — east longitude;

b) 1 — west longitude;

- LAHS — bit flag identifying the latitude hemisphere:

a) 0 — north latitude;

b) 1 — south latitude;

- RTM — time increment relative to the previous record (or to the ATM field value if this record is the first one) in units of 0.1 s. This value defines the measurement time for a given path point. The maximum permitted increment is 3.2 s;

- LAT — absolute latitude, degrees (WGS 84)/90 · 0xFFFFFFFF, rounded to an integer;

- LONG — absolute longitude, degrees (WGS 84)/180 · 0xFFFFFFFF, rounded to an integer;

- SPDL, SPDH — low (SPDL) and high (SPDH) bits of the speed parameter (15 bits used), in units of 0.01 km/h. The maximum speed value transferred in this field is 327.67 km/h;

- DIRH — most significant bit (bit 8) of the DIR parameter;

- DIR — vehicle movement direction expressed in degrees counting clockwise with respect to North (its highest bit is in the DIRH field). This value shall be within the range from 0 to 359.

7.4 Use of EGTS_COMMANDS_SERVICE

The description, structure and format of EGTS_COMMANDS_SERVICE sub-records used for the provision of the Base Service are detailed in 6.7.3.

7.5 List and description of commands, parameters and acknowledgements used in EGTS_ECALL_SERVICE

7.5.1 The list and description of IVDS commands and acknowledgements required for implementation of the Base Service as well as the list of IVDS parameters are given in Tables 46 and 47.

7.5.2 The IVDS parameters listed in subsections "IVDS acceleration profile recording in case of road accidents" and "Movement path recording in case of road accidents" (Table 47) are not mandatory if such functions are not implemented in the IVDS.

7.5.3 The following parameters in addition to the ones specified in 6.7.3.2 shall be supported by any IVDS installed in standard equipment configuration:

- EGTS_ECALL_TEST_NUMBER;
- EGTS_ECALL_SIGNAL_INTERNAL;
- EGTS_ECALL_SIGNAL_EXTERNAL;
- EGTS_ECALL_SOS_BUTTON_TIME;
- EGTS_ECALL_CCFT;
- EGTS_ECALL_INVITATION_SIGNAL_DURATION;
- EGTS_ECALL_SEND_MSG_PERIOD;
- EGTS_ECALL_AL_ACK_PERIOD;
- EGTS_ECALL_MSD_MAX_TRANSMISSION_TIME;
- EGTS_ECALL_NAD_DEREGISTRATION_TIMER;
- EGTS_ECALL_DIAL_DURATION;
- EGTS_ECALL_AUTO_DIAL_ATTEMPTS;
- EGTS_ECALL_MANUAL_DIAL_ATTEMPTS;
- EGTS_ECALL_MANUAL_CAN_CANCEL;
- EGTS_ECALL_SMS_FALLBACK_NUMBER;
- EGTS_CRASH_RECORD_TIME;
- EGTS_CRASH_RECORD_RESOLUTION;
- EGTS_CRASH_PRE_RECORD_TIME;
- EGTS_CRASH_PRE_RECORD_RESOLUTION;
- EGTS_TRACK_RECORD_TIME;
- EGTS_TRACK_RECORD_RESOLUTION;
- EGTS_TRACK_PRE_RECORD_TIME;
- EGTS_VEHICLE_VIN;
- EGTS_VEHICLE_TYPE;
- EGTS_VEHICLE_PROPULSION_STORAGE_TYPE.

Table 46 — List of commands sent to IVDS

Command name	Code	Type, number and limit value of parameters	Description
EGTS_ECALL_REQ	0x0112	BYTE/0,1	Command to initiate the emergency call with the IVDS. Sent using SMS only. Contains a single parameter that determines the event type: 0 — manual call; 1 — automatic call
EGTS ECALL MSD REQ	0x0113	BINARY (MID_INT, TRANSPORT_BYTE)	Command to repeat MSD transmission. Sent using SMS only. Contains two parameters: MID — message identifier of the requested MSD. If MID = 0, a new message is sent; TRANSPORT — channel type used by the IVDS to send the MSD: 0 — any, to be selected by the IVDS; 2 — SMS Note — When this command is received with TRANSPORT=2, sending the EGTS_SR_COMMAND_DATA sub-record with the CC_OK acknowledgement code in the CCT (Command Confirmation Type) field is not mandatory.
EGTS_ACCEL_DATA	0x0114	—	Command to send acceleration profile data. Sent using SMS only.
EGTS_TRACK _DATA	0x0115	—	Command to send movement path data. Sent using SMS only.
EGTS_ECALL_ DEREGISTRATION	0x0116	—	Command to de-register the IVDS in the wireless mobile communication network

Table 47 — List of IVDS parameters

Parameter name	Code	Parameter type	Default value	Description	Applicability[1)]	Possible change[2}]
Genera-purpose settings						
EGTS_ECALL_TEST_NUMBER	0x020D	STRING	""	Telephone number for test calls in the Road Accident Emergency Response System	AUX, STD, STD+	Yes
Service configuration and configuration data. Base Service of Road Accident Emergency Response System						
EGTS_ECALL_ON	0x0210	BOOLEAN	TRUE	True if emergency are calls possible	AUX, STD, STD+	Yes
EGTS_ECALL_CRASH_ SIGNAL_INTERNAL	0x0211	BOOLEAN	TRUE	True if a built-in IVDS acceleration sensor is used to detect RTA events	AUX	Yes

Table 47 (continued)

Parameter name	Code	Parameter type	Default value	Description	Applicability[1)]	Possible change[2}]
EGTS_ECALL_CRASH_ SIGNAL_EXTERNAL	0x0212	BOOLEAN	TRUE	True if a vehicle sensor external to the IVDS is used, e.g., a sensor initiated by operation of airbag(s) or other passive safety systems	AUX	Yes
EGTS_ECALL_SOS_BUTTON_ TIME	0x0213	INT	200	Time duration the "Emergency Call" button must be pressed to initiate the call, regardless of the ignition line state, ms	AUX	Yes
EGTS_ECALL_NO_ AUTOMATIC_TRIGGERING	0x0214	BOOLEAN	FALSE	Emergency Call mode initiation in automatic mode is OFF.	AUX, STD, STD+	Yes
EGTS_ASI15_TRESHOLD	0x0215	FLOAT	1.8	Operating threshold of the automatic detector of RTA events in terms of ASI5 index values for assessment of possible damage	AUX	Yes
EGTS_ECALL_MODE_PIN	0x0216	INT/0. ..8	0	Line indicating that the system is in ERA mode; NONE mode is not active; X — PINX line is active when the system is in this mode	AUX	Yes
EGTS_ECALL_CCFT	0x0217	INT	60	Counter setting for automatic call termination, min.	AUX, STD, STD+	Yes
EGTS_ECALL_INVITATION_ SIGNAL_DURATION	0x0218	INT	2000	Duration of INVITATION signal, ms	AUX, STD, STD+	Yes
EGTS_ECALL_SEND_MSG_ PERIOD	0x0219	INT	5000	SEND MSG message period, ms	AUX, STD, STD+	Yes
EGTS_ECALL_AL_ACK_ PERIOD	0x021A	INT	5000	AL-ACK period, ms	AUX, STD, STD+	Yes
EGTS_ECALL_MSD_MAX_ TRANSMISSION_TIME	0x021B	INT	20	Maximum MSD transmission time, s	AUX, STD, STD+	Yes
EGTS_ECALL_NAD_ DEREGISTRATION_TIMER	0x021D	INT	8	Time to expire before the GSM or UMTS module performs deregistration in the network, h	AUX, STD, STD+	Yes

Table 47 (continued)

Parameter name	Code	Parameter type	Default value	Description	Applicability[1)]	Possible change[2}]
EGTS_ECALL_DIAL_DURATION	0x021E	INT	5	Total dialling duration during initiation of the emergency call, min.	AUX, STD, STD+	Yes
EGTS_ECALL_AUTO_DIAL_ ATTEMPTS	0x021F	INT	10	Number of dialling attempts in case of automatically initiated calls. This value may not be set to 0.	AUX, STD, STD+	Yes
EGTS_ECALL_MANUAL_DIAL_ ATTEMPTS	0x0220	INT	10	Number of dialling attempts in case of manual emergency call initiation. This value may not be set to 0.	AUX, STD, STD+	Yes
EGTS_ECALL_MANUAL_CAN_ CANCEL	0x0222	BOOLEAN	TRUE	True if manually initiated emergency call may be cancelled by the user	AUX, STD, STD+	Yes
EGTS_ECALL_SMS_ FALLBACK_NUMBER	0x0223	STRING	"112"	Number used by the IVDS to send SMS with the minimum set of data at the request of System Operator.	AUX, STD, STD+	Yes
Acceleration profile recording in case of road accidents						
IGNITION_OFF_FOLLOW_UP_ TIME1	0x0224	INT	120	Time duration of acceleration profile recording with the ignition turned off in case of an accident, min	AUX	Yes
IGNITION_OFF_FOLLOW_UP_ TIME2	0x0225	INT	240	Time duration of accident event identification with the ignition turned off, min	AUX	Yes
EGTS_CRASH_RECORD_TIME	0x251	I NT/0...250	250	Record length of acceleration profile data during the accident, ms	AUX	Yes
EGTS_CRASH_RECORD_ RESOLUTION	0x0252	INT/1...5	1	Sample length in acceleration profile records during the accident, ms	AUX	Yes
EGTS_CRASH_PRE_RECORD_ TIME	0x0253	I NT/0...20000	20000	Record length of acceleration profile data before the accident, ms	AUX	Yes

Table 47 (continued)

Parameter name	Code	Parameter type	Default value	Description	Applicability[1)]	Possible change[2}]
EGTS_CRASH_PRE_RECORD_ RESOLUTION	0x0254	INT/5... 100	5	Sample length in acceleration profile records before the accident, ms	AUX	Yes
Movement path recording in case of road accidents						
EGTS_TRACK_RECORD_TIME	0x025A	INT/0... 180	10	Record length of vehicle path data during the accident, s. Setting this value to zero means that the vehicle movement path is not recorded during the accident.	AUX	Yes
EGTS_TRACK_PRE_ RECORD_TIME	0x025B	INT/0...600	20	Record length of vehicle path data before the accident, s. Setting this value to zero means that the vehicle movement path is not recorded before the accident.	AUX	Yes
EGTS_TRACK_RECORD_ RESOLUTION	0x025C	INT/1... 30	10	Sample length in vehicle path records, 100 ms	AUX	Yes
Vehicle parameters						
EGTS_VEHICLE_VIN	0x0311	STRING	""	VIN in accordance with [1] (Appendix 7)	AUX, STD, STD+	Yes

EGTS_VEHICLE_PROPULSION_ STORAGE_TYPE	0x0313	INT	0	Vehicle propulsion storage type. Multiple bits may be set if several energy sources are installed. If all bits are zero, the type is unspecified.
				a) bits 31—6: reserved;
				b) bit 5: hydrogen;
				c) bit 4: electricity (above 42 V and 100 A/h);
				d) bit 3: liquid propane (LPG);
				e) bit 2: compressed natural gas (CNG);
				f) bit 1: diesel;
				g) bit 0: gasoline
					AUX, STD, STD+	Yes

Table 47 (continued

Parameter name	Code	Parameter type	Default value	Description	Applicability[1)]	Possible change[2}]
EGTS_VEHICLE_TYPE	0x0312	INT	0	Vehicle type: 1 — passenger vehicle (Class M1) 2 — bus (Class M2) 3 — bus (Class M3) 4 — light cargo vehicle (Class N1) 5 — heavy cargo vehicle (Class N2) 6 — heavy cargo vehicle (Class N3) 7 — motorcycle (Class L1e) 8 — motorcycle (Class L2e) 9 — motorcycle (Class L3e) 10 — motorcycle (Class L4e) 11 — motorcycle (Class L5e) 12 — motorcycle (Class L6e) 13 — motorcycle (Class L7e)	AUX, STD, STD+	Yes

[1)] "AUX" — for IVDS in auxiliary equipment configuration; "STD" — for IVDS in standard equipment configuration that is only intended for the Base Service of the System; "STD+" for IVDS in standard equipment configuration that is also intended for other System services in addition to the Base Service.

[2)] "Yes" — means that the specified initial value of the IVDS parameter may change after the initial IVDS setup, "No" that the initial settings are not subject to changes while the IVDS is used.

8 AL-ACK message format

8.1 The AL-ACK issued by the Road Accident Emergency Response System to the IVDS side for confirmation that the minimum set of data received using an in-band modem is valid shall also be sent using an in-band modem.

8.2 Each AL-ACK message shall conform to the format described in Table 48.

Table 48 — Format of AL_ACK messages

AL-ACK data field	Bit position in data field	Meaning
Reserved field No. 1	4	Unused
Reserved field No. 2	3	Unused
Data validity indicator	2	0 — received data are valid (Positive ACK); 1 — call termination (Cleardown)
Data format version	1	0 — current format; 1 — reserved for future use

Appendix A
(reference)

Description of design principle of Navigation Information System based on Transport Layer Protocol

The minimum required and sufficient element of a system using the Transport Layer Protocol is a telematic platform. In order to describe its key component responsible for management of intra-platform interaction and routing, the concept of "dispatcher" is introduced.

The Protocol uses different approaches to inter-platform routing where the data (information packets) are transmitted between separate telematic platforms, and to intra-platform routing where the data are transmitted between individual services of a single platform. The term "service" means a dedicated telematic platform component that provides for execution of a particular servicing algorithm using the Transport Layer Protocol described herein. For both routing types mentioned above, interaction is performed through the dispatcher.

Data producers and data consumers in a system based on the Transport Layer Protocol are the services generating packets on the originator side and those processing the packets received from other services on the recipient side. Each service implements individual business logics as appropriate for its specific servicing processes. The service type is its key functional specification used by the dispatcher for intra-platform routing. A complementary pair of services is normally involved in interaction where the first service is running on the subscriber terminal (an IVDS in the context of this Standard), e.g., generating packets which include coordinate data and sensor readings, while the second service is running on the telematic platform and is processing such data.

All services within a single telematic platform are connected to the dispatcher, and are not directly linked with each other.

A telematic platform may communicate with other platforms and perform data exchange based on the routing data. For the purpose of routing, the dispatcher connects to a local storage containing the data on any adjacent telematic platforms and the services available on them, as well as the information on the services operating within his own platform. When the dispatchers of different telematic platforms communicate with each other, they exchange information on the services available on each side as well as on the status of such services. Finding a route is reduced to a search of direction (connection) matching the type of the requested service. If the requested service is on the dispatcher's platform itself, intra-platform routing is only used for interaction. It means that when the appropriate permissions are granted, the search of services proceeds using the routing data on adjacent telematic platforms, and if an appropriate route is found and permitted for use, the request is relayed to that platform, and the dispatcher ID of the remote platform is used as a destination address.

Analogously, an IVDS also interacts with telematic platform services through the dispatcher. In this case, the IVDS is identified by special packets that contain a unique IVDS number assigned to the IVDS during its registration in the System as well as other accounting data and information pertaining to the IVDS internal structure and to the state of IVDS modules and units.

The structural diagram illustrating how the elements of the system based on the described Transport Layer Protocol interact is shown in Figure A.1. Each service has a certain type which is indicated by the SID parameter on this Figure.

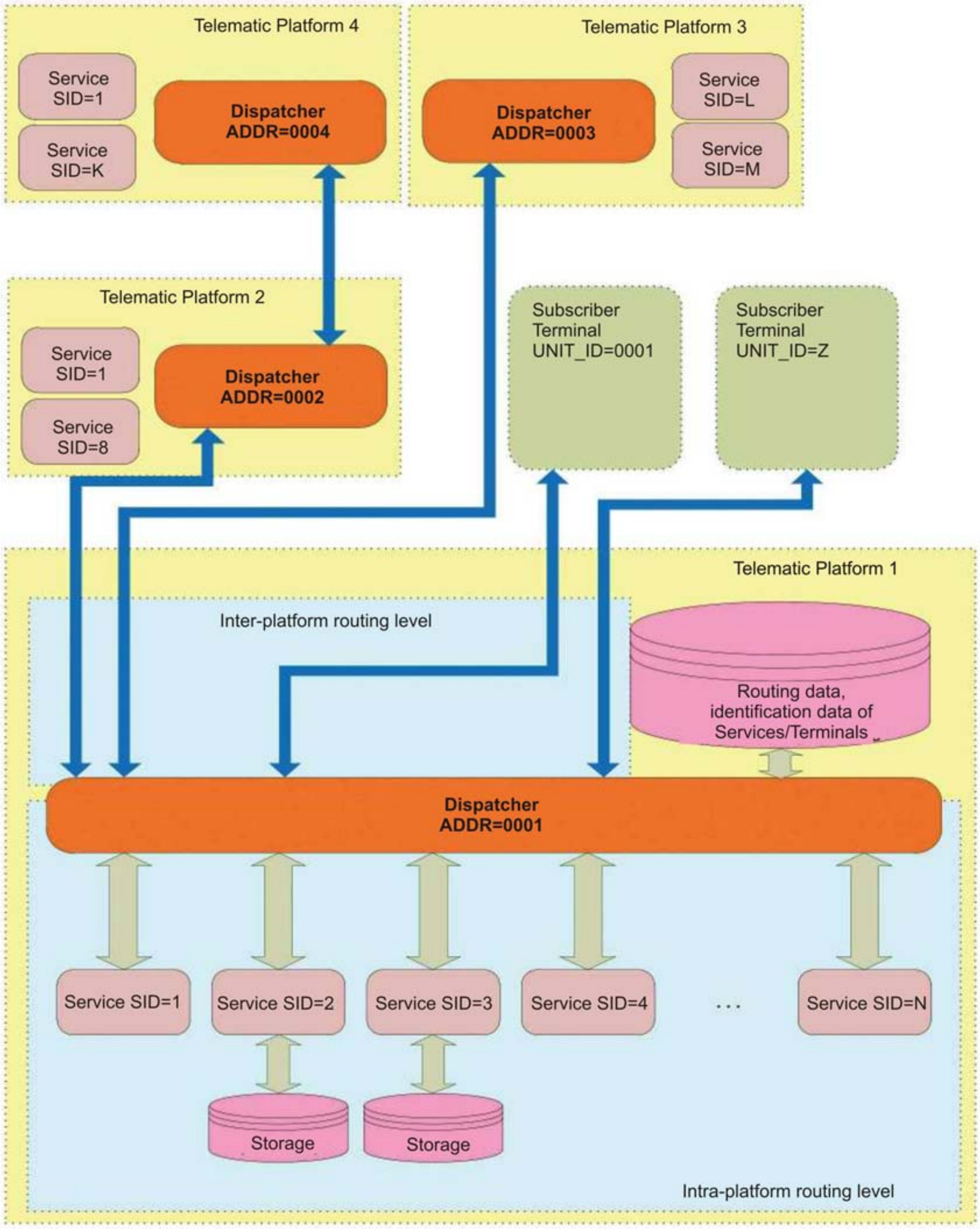


Figure A.1 — Block diagram of interaction between elements of System based on Transport Layer Protocol

Appendix B
(reference)

Consideration of Transport Layer Protocol from viewpoint of NGTP concept

Three interaction entities play a major role in the NGTP-based design concept of telematic systems: telematic device, provider of telematic services, and dispatcher. Interaction takes place using standardised interfaces and covers protocol elements other than the provider of telematic services which is integrated with the dispatcher in the protocol.

A telematic device (an IVDS as regards this Standard) is normally integrated into the vehicle, but may also be a personal navigation device or a mobile telephone.

A provider of telematic services (PTS) is an entity intended for data exchange between the services and telematic devices (TD).

According to the NGPT, a dispatcher is an intermediate party between the PTS and PU that provides a standard interface for communication of the TD with other system components which are necessary for operation of services. The dispatcher operates with the data that belong to its own layer, and does not parse any data of the service layer.

The NGTP header is identical to the leading header bytes of the Transport Layer Protocol: Protocol Version (1 byte), Security Context (2 bytes), NGTP Header Length (1 byte), and NGTP Header Encoding (1 byte).

In the NGTP, an IVDS is identified by VIN/DrivelD, while in this protocol, by UNIT_ID.

The VIN field is used to identify IVDS manufactured in standard equipment configuration.

Similar to the NGTP, this protocol is aimed at flexible routing of services between the IVDS and the telematic platform. With such routing, implementation of a new service needs no adaptation of the protocol as the latter is responsible for the data routing only, whereas the processing is carried out in the service itself. It suffices to setup proper dispatcher's routing for a new service type using administration utilities as far as the system is designed for the Transport Layer Protocol.

Another concept the NGTP operates with is the "event" that defines a certain generic data pattern used for integration of diversified information structures into a generalised data array. Each event identifier is also associated with an attribute that identifies the event generation time. The use of such generalisation mechanism is an intrinsic feature of the Transport Layer Protocol where each Service Support Layer Protocol record may contain an event ID generated by the source of such records for a definite time slice, e.g., when a road accident takes place.

Unlike the NGTP which uses different interfaces between the TD and dispatcher, between the dispatcher and PTS, and between the PTS and services, the Transport Layer Protocol for IVDS uses a common interface for communication of components.

The NGTP makes use of the term "trigger" which implies some kind of notification sent to system components in order to report that certain data have been received for them. When such "trigger" is received, the recipient must request and process those data. The Transport Layer Protocol does not make use of the "triggers," and the information is immediately sent to the destination instead.

Appendix C
(mandatory)

Processing result codes

The processing result codes are listed in Table C.1.

Table C.1 — Processing result codes

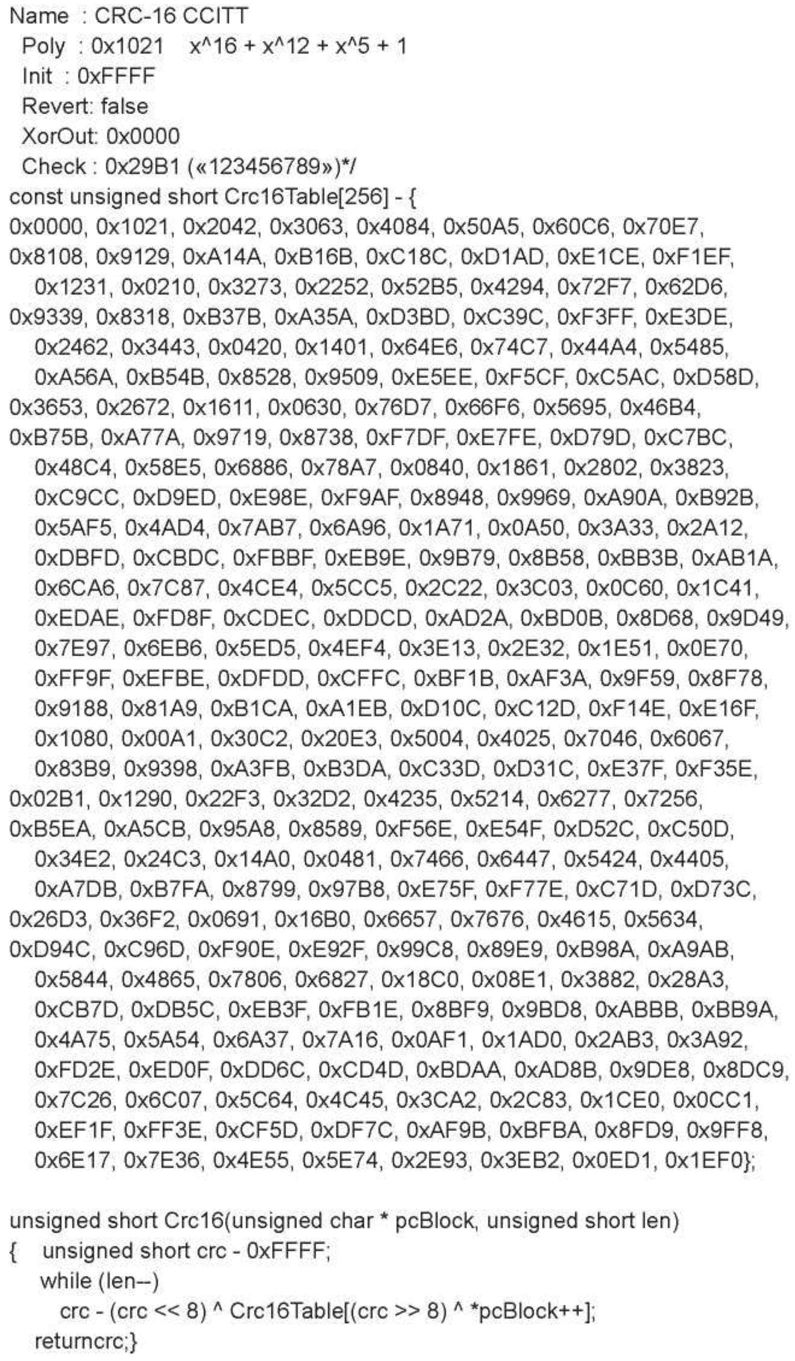
Value	Name	Description
0	EGTS_PC_OK	Processing succeeded
1	EGTS_PC_IN_PROGRESS	Processing in progress (processing result is not known yet)
128	EGTS_PC_UNS_PROTOCOL	Protocol not supported
129	EGTS_PC_DECRYPT_ERROR	Decryption error
130	EGTS_PC_PROC_DENIED	Processing not permitted
131	EGTS_PC_INC_HEADERFORM	Incorrect header format
132	EGTS_PC_INC_DATAFORM	Incorrect data format
133	EGTS_PC_UNS_TYPE	Type not supported
134	EGTS_PC_NOTEN_PARAMS	Incorrect number of parameters
135	EGTS_PC_DBL_PROC	Processing retry attempt
136	EGTS_PC_PROC_SRC_DENIED	Data processing not permitted for this source
137	EGTS_PC_HEADERCRC_ERROR	Header checksum error
138	EGTS_PC_DATACRC_ERROR	Data checksum error
139	EGTS_PC_INVDATALEN	Invalid data length
140	EGTS_PC_ROUTE_NFOUND	Route not found
141	EGTS_PC_ROUTE_CLOSED	Route closed
142	EGTS_PC_ROUTE_DENIED	Route not permitted
143	EGTS_PC_INVADDR	Invalid address
144	EGTS_PC_TTLEXPIRED	Time to live expired
145	EGTS_PC_NO_ACK	No acknowledgement received
146	EGTS_PC_OBJ_N FOUND	Object not found
147	EGTS_PC_EVNT_N FOUND	Event not found
148	EGTS_PC_SRVC_NFOUND	Service not found
149	EGTS_PC_SRVC_DENIED	Service denied
150	EGTS_PC_SRVC_U N KN	Service type unknown
151	EGTS_PC_AUTH_DENIED	Authorisation denied
152	EGTS_PC_ALREADY_EXISTS	Object already present
153	EGTS_PC_ID_N FOUND	Identifier not found
154	EGTS_PC_INC_DATETIME	Incorrect date and time

Table C.1 (continued)

Value	Name	Description
155	EGTS_PC_IO_ERROR	Input/output error
156	EGTS_PC_NO_RES_AVAIL	Insufficient resources
157	EGTS_PC_MODULE_FAULT	Internal module fault
158	EGTS_PC_MODULE_PWR_FLT	Power circuit fault
159	EGTS_PC_MODULE_PROC_FLT	Module processor fault
160	EGTS_PC_MODULE_SW_FLT	Module software fault
161	EGTS_PC_MODU LE_FW_FLT	Module firmware fault
162	EGTS_PC_MODULE_IO_FLT	Module I/O fault
163	EGTS_PC_MODULE_MEM_FLT	Internal module memory fault
164	EGTS_PC_TEST_FAILED	Test failed
Note — Error message packets (EGTS_PC_DECRYPT_ERROR, EGTS_PC_UNS_PROTOCOL, EGTS_PC_INC_DATAFORM, EGTS_PC_DATACRC_ERROR, EGTS_PC_INC_HEADERFORM, EGTS_PC_HEADERCRC_ERROR) are intended for equipment tests, and may be excluded in final software/IVDS releases.		

Appendix D
(reference)

Example implementation of CRC16 checksum calculation in C language



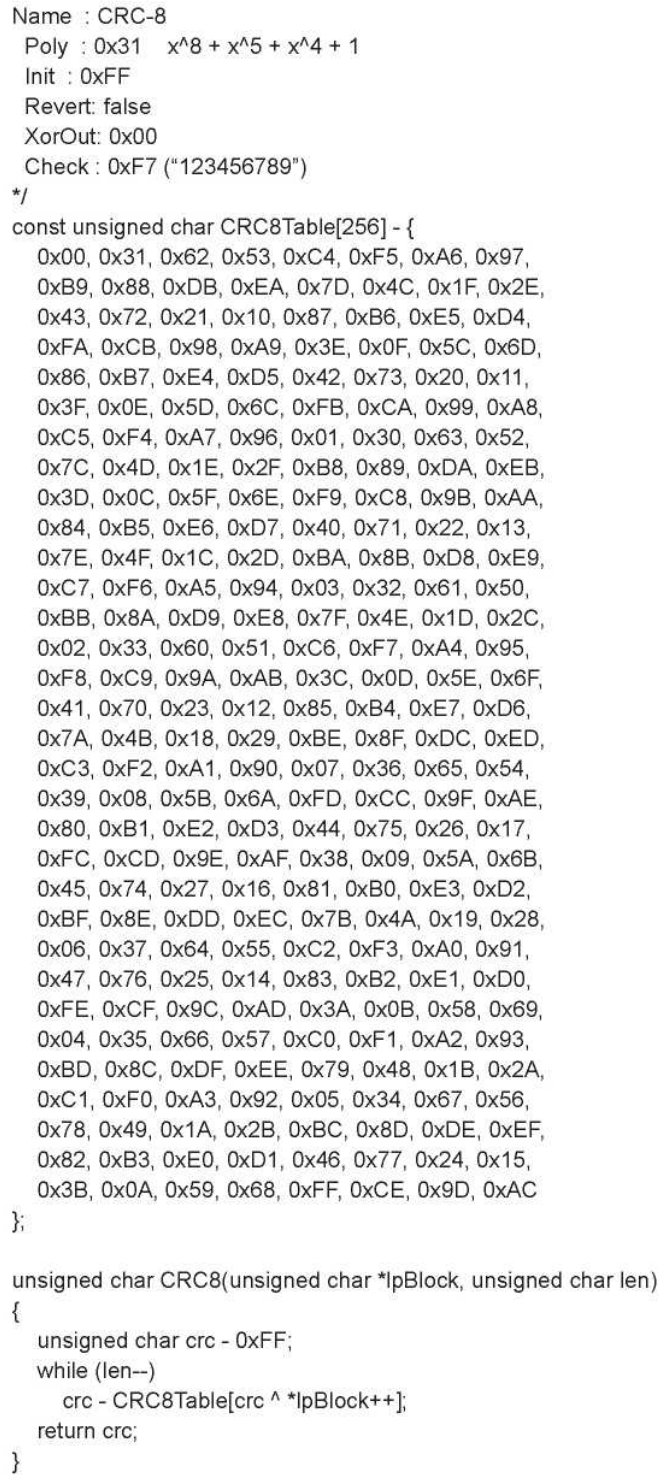
```
/*
 Name  : CRC-16 CCITT
   Poly  : 0x1021    x^16 + x^12 + x^5 + 1
   Init  : 0xFFFF
   Revert: false
   XorOut: 0x0000
   Check : 0x29B1 («123456789»)*/
 const unsigned short Crc16Table[256] - {
 0x0000, 0x1021, 0x2042, 0x3063, 0x4084, 0x50A5, 0x60C6, 0x70E7,
 0x8108, 0x9129, 0xA14A, 0xB16B, 0xC18C, 0xD1AD, 0xE1CE, 0xF1EF,
    0x1231, 0x0210, 0x3273, 0x2252, 0x52B5, 0x4294, 0x72F7, 0x62D6,
 0x9339, 0x8318, 0xB37B, 0xA35A, 0xD3BD, 0xC39C, 0xF3FF, 0xE3DE,
    0x2462, 0x3443, 0x0420, 0x1401, 0x64E6, 0x74C7, 0x44A4, 0x5485,
    0xA56A, 0xB54B, 0x8528, 0x9509, 0xE5EE, 0xF5CF, 0xC5AC, 0xD58D,
 0x3653, 0x2672, 0x1611, 0x0630, 0x76D7, 0x66F6, 0x5695, 0x46B4,
 0xB75B, 0xA77A, 0x9719, 0x8738, 0xF7DF, 0xE7FE, 0xD79D, 0xC7BC,
    0x48C4, 0x58E5, 0x6886, 0x78A7, 0x0840, 0x1861, 0x2802, 0x3823,
    0xC9CC, 0xD9ED, 0xE98E, 0xF9AF, 0x8948, 0x9969, 0xA90A, 0xB92B,
    0x5AF5, 0x4AD4, 0x7AB7, 0x6A96, 0x1A71, 0x0A50, 0x3A33, 0x2A12,
    0xDBFD, 0xCBDC, 0xFBBF, 0xEB9E, 0x9B79, 0x8B58, 0xBB3B, 0xAB1A,
    0x6CA6, 0x7C87, 0x4CE4, 0x5CC5, 0x2C22, 0x3C03, 0x0C60, 0x1C41,
    0xEDAE, 0xFD8F, 0xCDEC, 0xDDCD, 0xAD2A, 0xBD0B, 0x8D68, 0x9D49,
    0x7E97, 0x6EB6, 0x5ED5, 0x4EF4, 0x3E13, 0x2E32, 0x1E51, 0x0E70,
    0xFF9F, 0xEFBE, 0xDFDD, 0xCFFC, 0xBF1B, 0xAF3A, 0x9F59, 0x8F78,
    0x9188, 0x81A9, 0xB1CA, 0xA1EB, 0xD10C, 0xC12D, 0xF14E, 0xE16F,
    0x1080, 0x00A1, 0x30C2, 0x20E3, 0x5004, 0x4025, 0x7046, 0x6067,
    0x83B9, 0x9398, 0xA3FB, 0xB3DA, 0xC33D, 0xD31C, 0xE37F, 0xF35E,
 0x02B1, 0x1290, 0x22F3, 0x32D2, 0x4235, 0x5214, 0x6277, 0x7256,
 0xB5EA, 0xA5CB, 0x95A8, 0x8589, 0xF56E, 0xE54F, 0xD52C, 0xC50D,
    0x34E2, 0x24C3, 0x14A0, 0x0481, 0x7466, 0x6447, 0x5424, 0x4405,
    0xA7DB, 0xB7FA, 0x8799, 0x97B8, 0xE75F, 0xF77E, 0xC71D, 0xD73C,
 0x26D3, 0x36F2, 0x0691, 0x16B0, 0x6657, 0x7676, 0x4615, 0x5634,
 0xD94C, 0xC96D, 0xF90E, 0xE92F, 0x99C8, 0x89E9, 0xB98A, 0xA9AB,
    0x5844, 0x4865, 0x7806, 0x6827, 0x18C0, 0x08E1, 0x3882, 0x28A3,
    0xCB7D, 0xDB5C, 0xEB3F, 0xFB1E, 0x8BF9, 0x9BD8, 0xABBB, 0xBB9A,
    0x4A75, 0x5A54, 0x6A37, 0x7A16, 0x0AF1, 0x1AD0, 0x2AB3, 0x3A92,
    0xFD2E, 0xED0F, 0xDD6C, 0xCD4D, 0xBDAA, 0xAD8B, 0x9DE8, 0x8DC9,
    0x7C26, 0x6C07, 0x5C64, 0x4C45, 0x3CA2, 0x2C83, 0x1CE0, 0x0CC1,
    0xEF1F, 0xFF3E, 0xCF5D, 0xDF7C, 0xAF9B, 0xBFBA, 0x8FD9, 0x9FF8,
    0x6E17, 0x7E36, 0x4E55, 0x5E74, 0x2E93, 0x3EB2, 0x0ED1, 0x1EF0};

 unsigned short Crc16(unsigned char * pcBlock, unsigned short len)
 {    unsigned short crc - 0xFFFF;
      while (len--)
         crc - (crc << 8) ^ Crc16Table[(crc >> 8) ^ *pcBlock++];
    returncrc;}
```

Appendix E
(reference)

Example implementation of CRC8 checksum calculation in C language

/*



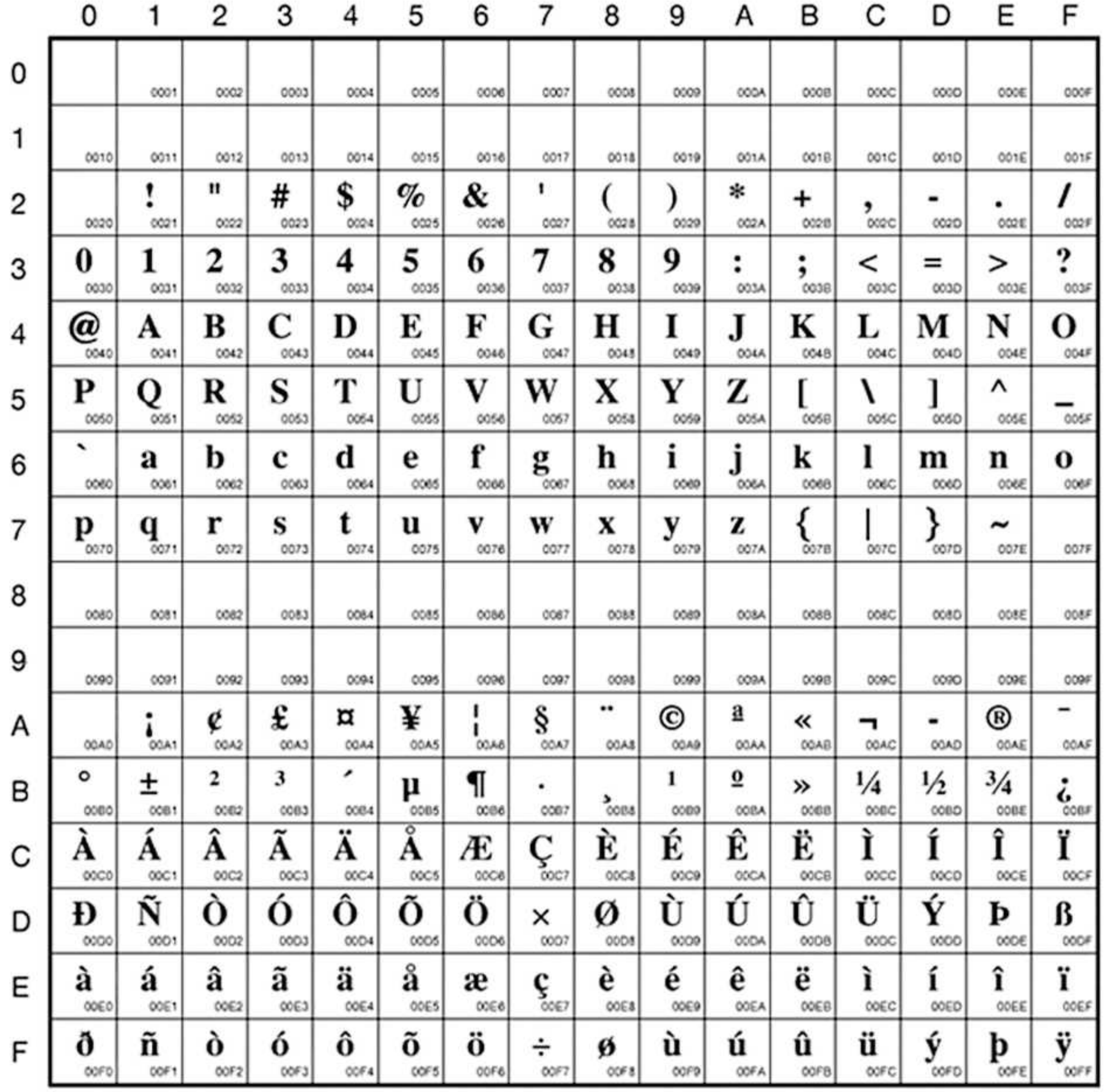
```
 Name  : CRC-8
   Poly  : 0x31    x^8 + x^5 + x^4 + 1
   Init  : 0xFF
   Revert: false
   XorOut: 0x00
   Check : 0xF7 (“123456789”)
 */
 const unsigned char CRC8Table[256] - {
     0x00, 0x31, 0x62, 0x53, 0xC4, 0xF5, 0xA6, 0x97,
     0xB9, 0x88, 0xDB, 0xEA, 0x7D, 0x4C, 0x1F, 0x2E,
     0x43, 0x72, 0x21, 0x10, 0x87, 0xB6, 0xE5, 0xD4,
     0xFA, 0xCB, 0x98, 0xA9, 0x3E, 0x0F, 0x5C, 0x6D,
     0x86, 0xB7, 0xE4, 0xD5, 0x42, 0x73, 0x20, 0x11,
     0x3F, 0x0E, 0x5D, 0x6C, 0xFB, 0xCA, 0x99, 0xA8,
     0xC5, 0xF4, 0xA7, 0x96, 0x01, 0x30, 0x63, 0x52,
     0x7C, 0x4D, 0x1E, 0x2F, 0xB8, 0x89, 0xDA, 0xEB,
     0x3D, 0x0C, 0x5F, 0x6E, 0xF9, 0xC8, 0x9B, 0xAA,
     0x84, 0xB5, 0xE6, 0xD7, 0x40, 0x71, 0x22, 0x13,
     0x7E, 0x4F, 0x1C, 0x2D, 0xBA, 0x8B, 0xD8, 0xE9,
     0xC7, 0xF6, 0xA5, 0x94, 0x03, 0x32, 0x61, 0x50,
     0xBB, 0x8A, 0xD9, 0xE8, 0x7F, 0x4E, 0x1D, 0x2C,
     0x02, 0x33, 0x60, 0x51, 0xC6, 0xF7, 0xA4, 0x95,
     0xF8, 0xC9, 0x9A, 0xAB, 0x3C, 0x0D, 0x5E, 0x6F,
     0x41, 0x70, 0x23, 0x12, 0x85, 0xB4, 0xE7, 0xD6,
     0x7A, 0x4B, 0x18, 0x29, 0xBE, 0x8F, 0xDC, 0xED,
     0xC3, 0xF2, 0xA1, 0x90, 0x07, 0x36, 0x65, 0x54,
     0x39, 0x08, 0x5B, 0x6A, 0xFD, 0xCC, 0x9F, 0xAE,
     0x80, 0xB1, 0xE2, 0xD3, 0x44, 0x75, 0x26, 0x17,
     0xFC, 0xCD, 0x9E, 0xAF, 0x38, 0x09, 0x5A, 0x6B,
     0x45, 0x74, 0x27, 0x16, 0x81, 0xB0, 0xE3, 0xD2,
     0xBF, 0x8E, 0xDD, 0xEC, 0x7B, 0x4A, 0x19, 0x28,
     0x06, 0x37, 0x64, 0x55, 0xC2, 0xF3, 0xA0, 0x91,
     0x47, 0x76, 0x25, 0x14, 0x83, 0xB2, 0xE1, 0xD0,
     0xFE, 0xCF, 0x9C, 0xAD, 0x3A, 0x0B, 0x58, 0x69,
     0x04, 0x35, 0x66, 0x57, 0xC0, 0xF1, 0xA2, 0x93,
     0xBD, 0x8C, 0xDF, 0xEE, 0x79, 0x48, 0x1B, 0x2A,
     0xC1, 0xF0, 0xA3, 0x92, 0x05, 0x34, 0x67, 0x56,
     0x78, 0x49, 0x1A, 0x2B, 0xBC, 0x8D, 0xDE, 0xEF,
     0x82, 0xB3, 0xE0, 0xD1, 0x46, 0x77, 0x24, 0x15,
     0x3B, 0x0A, 0x59, 0x68, 0xFF, 0xCE, 0x9D, 0xAC
 };

 unsigned char CRC8(unsigned char *lpBlock, unsigned char len)
 {
     unsigned char crc - 0xFF;
     while (len--)
         crc - CRC8Table[crc ^ *lpBlock++];
     return crc;
 }
```

Appendix F
(reference)

Character encoding tables

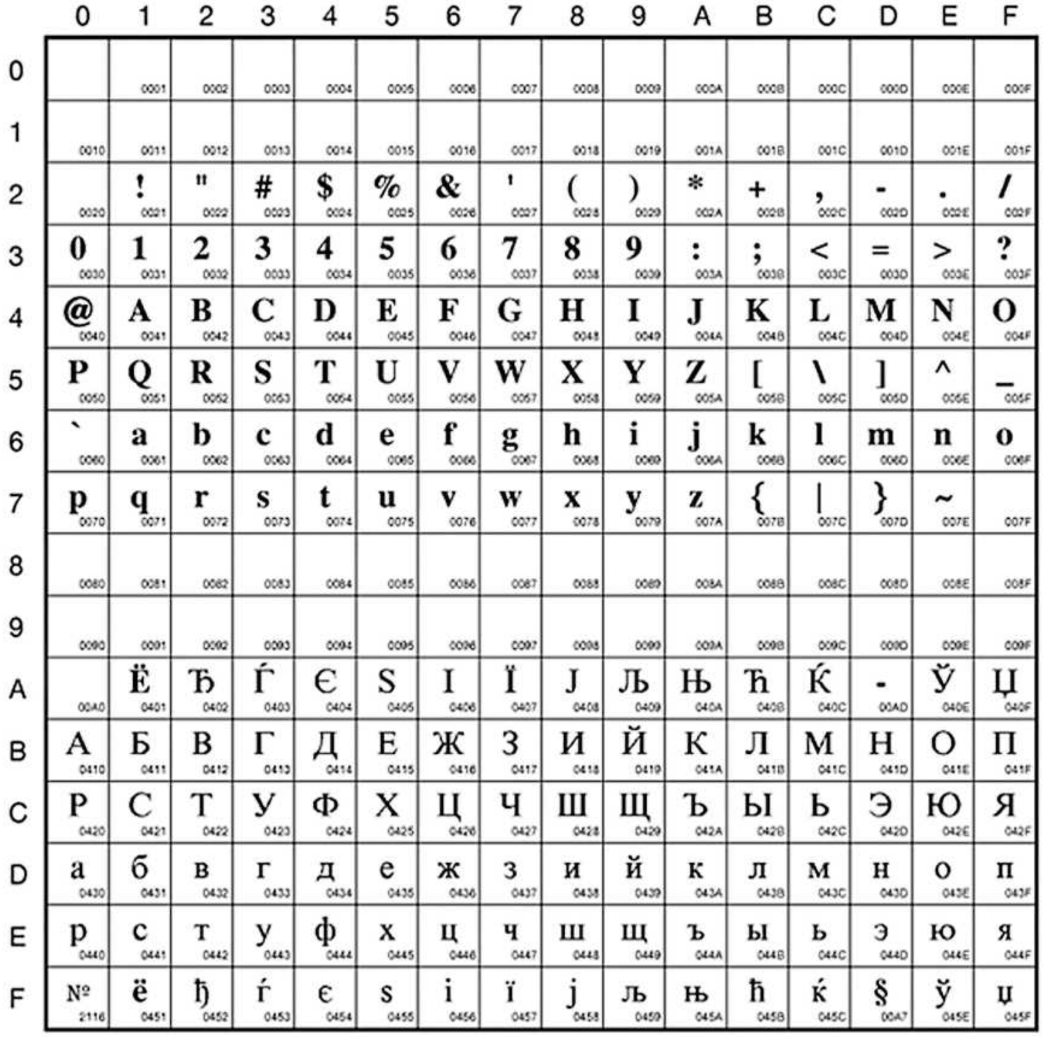
F.1 Character encoding of Latin alphabet is shown in Figure F.1.



	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0		0001	0002	0003	0004	0005	0006	0007	0008	0009	000A	000B	000C	000D	000E	000F
1	0010	0011	0012	0013	0014	0015	0016	0017	0018	0019	001A	001B	001C	001D	001E	001F
2	0020	! 0021	" 0022	# 0023	$ 0024	% 0025	& 0026	' 0027	(0028	) 0029	* 002A	+ 002B	, 002C	- 002D	. 002E	/ 002F
3	0 0030	1 0031	2 0032	3 0033	4 0034	5 0035	6 0036	7 0037	8 0038	9 0039	: 003A	; 003B	< 003C	= 003D	> 003E	? 003F
4	@ 0040	A 0041	B 0042	C 0043	D 0044	E 0045	F 0046	G 0047	H 0048	I 0049	J 004A	K 004B	L 004C	M 004D	N 004E	O 004F
5	P 0050	Q 0051	R 0052	S 0053	T 0054	U 0055	V 0056	W 0057	X 0058	Y 0059	Z 005A	[005B	\ 005C	] 005D	^ 005E	_ 005F
6	` 0060	a 0061	b 0062	c 0063	d 0064	e 0065	f 0066	g 0067	h 0068	i 0069	j 006A	k 006B	l 006C	m 006D	n 006E	o 006F
7	p 0070	q 0071	r 0072	s 0073	t 0074	u 0075	v 0076	w 0077	x 0078	y 0079	z 007A	{ 007B	| 007C	} 007D	~ 007E	007F
8	0080	0081	0082	0083	0084	0085	0086	0087	0088	0089	008A	008B	008C	008D	008E	008F
9	0090	0091	0092	0093	0094	0095	0096	0097	0098	0099	009A	009B	009C	009D	009E	009F
A	00A0	¡ 00A1	¢ 00A2	£ 00A3	¤ 00A4	¥ 00A5	¦ 00A6	§ 00A7	¨ 00A8	© 00A9	ª 00AA	« 00AB	¬ 00AC	- 00AD	® 00AE	¯ 00AF
B	° 00B0	± 00B1	² 00B2	³ 00B3	´ 00B4	µ 00B5	¶ 00B6	· 00B7	¸ 00B8	¹ 00B9	º 00BA	» 00BB	¼ 00BC	½ 00BD	¾ 00BE	¿ 00BF
C	À 00C0	Á 00C1	Â 00C2	Ã 00C3	Ä 00C4	Å 00C5	Æ 00C6	Ç 00C7	È 00C8	É 00C9	Ê 00CA	Ë 00CB	Ì 00CC	Í 00CD	Î 00CE	Ï 00CF
D	Ð 00D0	Ñ 00D1	Ò 00D2	Ó 00D3	Ô 00D4	Õ 00D5	Ö 00D6	× 00D7	Ø 00D8	Ù 00D9	Ú 00DA	Û 00DB	Ü 00DC	Ý 00DD	Þ 00DE	ß 00DF
E	à 00E0	á 00E1	â 00E2	ã 00E3	ä 00E4	å 00E5	æ 00E6	ç 00E7	è 00E8	é 00E9	ê 00EA	ë 00EB	ì 00EC	í 00ED	î 00EE	ï 00EF
F	ð 00F0	ñ 00F1	ò 00F2	ó 00F3	ô 00F4	õ 00F5	ö 00F6	÷ 00F7	ø 00F8	ù 00F9	ú 00FA	û 00FB	ü 00FC	ý 00FD	þ 00FE	ÿ 00FF

Figure F.1 — Character encoding of Latin alphabet

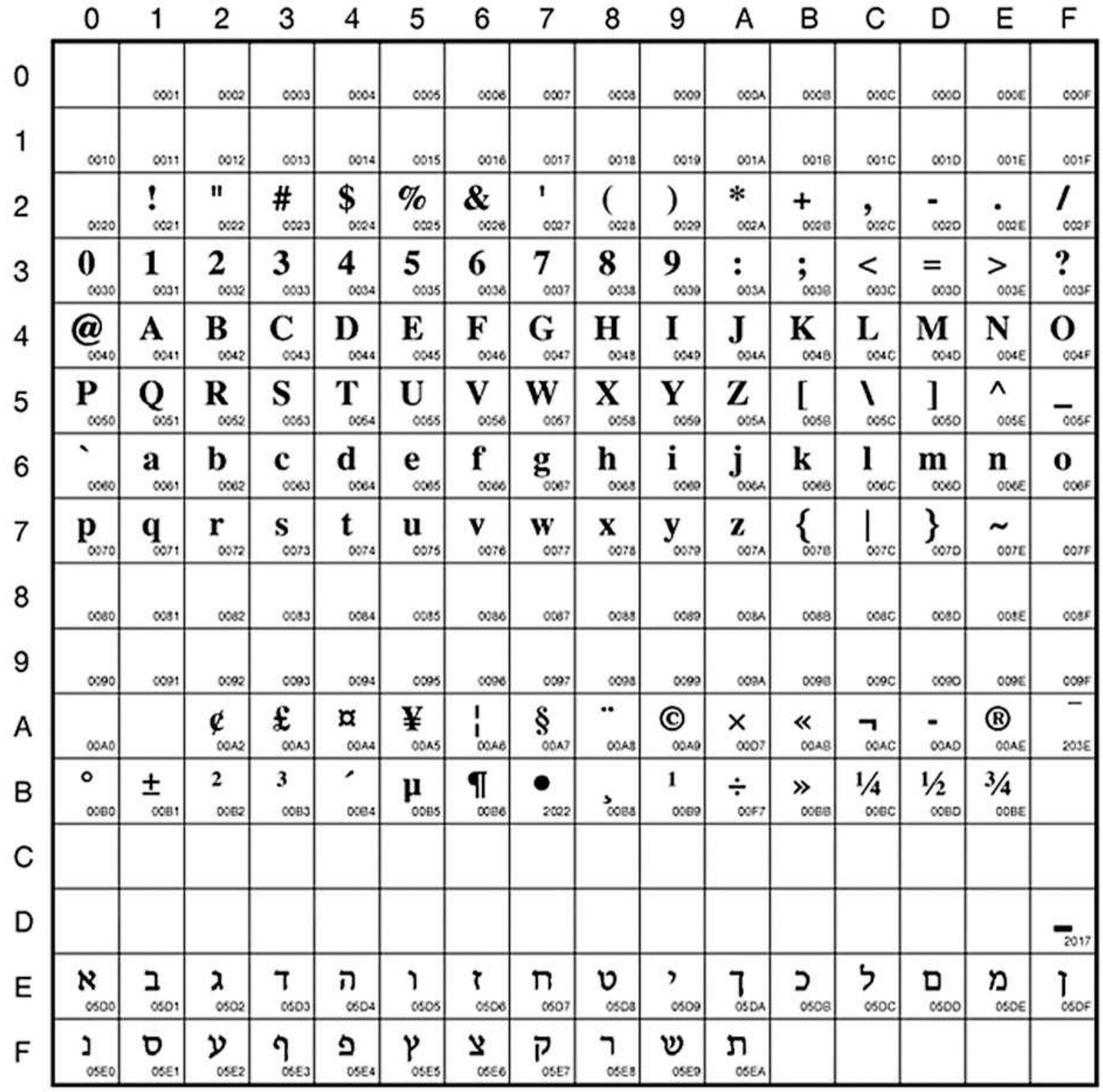
F.2 Character encoding of Latin and Cyrillic alphabets is shown in Figure F.2.



	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0		0001	0002	0003	0004	0005	0006	0007	0008	0009	000A	000B	000C	000D	000E	000F
1	0010	0011	0012	0013	0014	0015	0016	0017	0018	0019	001A	001B	001C	001D	001E	001F
2	0020	! 0021	" 0022	# 0023	$ 0024	% 0025	& 0026	' 0027	(0028	) 0029	* 002A	+ 002B	, 002C	- 002D	. 002E	/ 002F
3	0 0030	1 0031	2 0032	3 0033	4 0034	5 0035	6 0036	7 0037	8 0038	9 0039	: 003A	; 003B	< 003C	= 003D	> 003E	? 003F
4	@ 0040	A 0041	B 0042	C 0043	D 0044	E 0045	F 0046	G 0047	H 0048	I 0049	J 004A	K 004B	L 004C	M 004D	N 004E	O 004F
5	P 0050	Q 0051	R 0052	S 0053	T 0054	U 0055	V 0056	W 0057	X 0058	Y 0059	Z 005A	[005B	\ 005C	] 005D	^ 005E	_ 005F
6	` 0060	a 0061	b 0062	c 0063	d 0064	e 0065	f 0066	g 0067	h 0068	i 0069	j 006A	k 006B	l 006C	m 006D	n 006E	o 006F
7	p 0070	q 0071	r 0072	s 0073	t 0074	u 0075	v 0076	w 0077	x 0078	y 0079	z 007A	{ 007B	| 007C	} 007D	~ 007E	007F
8	0080	0081	0082	0083	0084	0085	0086	0087	0088	0089	008A	008B	008C	008D	008E	008F
9	0090	0091	0092	0093	0094	0095	0096	0097	0098	0099	009A	009B	009C	009D	009E	009F
A	00A0	Ё 0401	Ђ 0402	Ѓ 0403	Є 0404	Ѕ 0405	І 0406	Ї 0407	Ј 0408	Љ 0409	Њ 040A	Ћ 040B	Ќ 040C	- 00AD	Ў 040E	Џ 040F
B	А 0410	Б 0411	В 0412	Г 0413	Д 0414	Е 0415	Ж 0416	З 0417	И 0418	Й 0419	К 041A	Л 041B	М 041C	Н 041D	О 041E	П 041F
C	Р 0420	С 0421	Т 0422	У 0423	Ф 0424	Х 0425	Ц 0426	Ч 0427	Ш 0428	Щ 0429	Ъ 042A	Ы 042B	Ь 042C	Э 042D	Ю 042E	Я 042F
D	а 0430	б 0431	в 0432	г 0433	д 0434	е 0435	ж 0436	з 0437	и 0438	й 0439	к 043A	л 043B	м 043C	н 043D	о 043E	п 043F
E	р 0440	с 0441	т 0442	у 0443	ф 0444	х 0445	ц 0446	ч 0447	ш 0448	щ 0449	ъ 044A	ы 044B	ь 044C	э 044D	ю 044E	я 044F
F	№ 2116	ё 0451	ђ 0452	ѓ 0453	є 0454	ѕ 0455	і 0456	ї 0457	ј 0458	љ 0459	њ 045A	ћ 045B	ќ 045C	§ 00A7	ў 045E	џ 045F

Figure F.2 — Character encoding of Latin and Cyrillic alphabets

F.3 Character encoding of Latin and Hebrew alphabets is shown in Figure F.3.



	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0		0001	0002	0003	0004	0005	0006	0007	0008	0009	000A	000B	000C	000D	000E	000F
1	0010	0011	0012	0013	0014	0015	0016	0017	0018	0019	001A	001B	001C	001D	001E	001F
2	0020	! 0021	" 0022	# 0023	$ 0024	% 0025	& 0026	' 0027	(0028	) 0029	* 002A	+ 002B	, 002C	- 002D	. 002E	/ 002F
3	0 0030	1 0031	2 0032	3 0033	4 0034	5 0035	6 0036	7 0037	8 0038	9 0039	: 003A	; 003B	< 003C	= 003D	> 003E	? 003F
4	@ 0040	A 0041	B 0042	C 0043	D 0044	E 0045	F 0046	G 0047	H 0048	I 0049	J 004A	K 004B	L 004C	M 004D	N 004E	O 004F
5	P 0050	Q 0051	R 0052	S 0053	T 0054	U 0055	V 0056	W 0057	X 0058	Y 0059	Z 005A	[005B	\ 005C	] 005D	^ 005E	_ 005F
6	` 0060	a 0061	b 0062	c 0063	d 0064	e 0065	f 0066	g 0067	h 0068	i 0069	j 006A	k 006B	l 006C	m 006D	n 006E	o 006F
7	p 0070	q 0071	r 0072	s 0073	t 0074	u 0075	v 0076	w 0077	x 0078	y 0079	z 007A	{ 007B	| 007C	} 007D	~ 007E	007F
8	0080	0081	0082	0083	0084	0085	0086	0087	0088	0089	008A	008B	008C	008D	008E	008F
9	0090	0091	0092	0093	0094	0095	0096	0097	0098	0099	009A	009B	009C	009D	009E	009F
A	00A0		¢ 00A2	£ 00A3	¤ 00A4	¥ 00A5	¦ 00A6	§ 00A7	¨ 00A8	© 00A9	× 00D7	« 00AB	¬ 00AC	- 00AD	® 00AE	‾ 203E
B	° 00B0	± 00B1	² 00B2	³ 00B3	´ 00B4	µ 00B5	¶ 00B6	● 2022	¸ 00B8	¹ 00B9	÷ 00F7	» 00BB	¼ 00BC	½ 00BD	¾ 00BE	
C																
D																‗ 2017
E	א 05D0	ב 05D1	ג 05D2	ד 05D3	ה 05D4	ו 05D5	ז 05D6	ח 05D7	ט 05D8	י 05D9	ך 05DA	כ 05DB	ל 05DC	ם 05DD	מ 05DE	ן 05DF
F	נ 05E0	ס 05E1	ע 05E2	ף 05E3	פ 05E4	ץ 05E5	צ 05E6	ק 05E7	ר 05E8	ש 05E9	ת 05EA					

Figure F.3 — Character encoding of Latin and Hebrew alphabets

Bibliography

[1] Technical Regulation TR CU 018/2011 of the Customs Union "On Safety of Wheeled Vehicles", approved by Order No. 877 dated December 9, 2011, of the Customs Union Commission (in edition of the Eurasian Economic Commission No. 6 dated 30.01.201)

[2] ISO/IEC 7498-1-94 Information technology — Open Systems Interconnection — Basic reference model. Part 1: The basic model

[3] ETSI TS 126 267 (3GPPTS 26.267) Technical Specification Group Services and System Aspects; eCall Data Transfer; In-band modem solution; General description, Release 8

[4] ISO/IEC 10967-1:2012 Information technology — Language independent arithmetic — Part 1: Integer and floating point arithmetic

[5] GSM 03.38 (ETS 300 628) Digital cellular telecommunication system (Phase 2); Alphabets and language-specific information

[6] GSM 03.40 (ETS 300 536) Digital cellular telecommunication system (Phase 2)

UDC 656.13:004:006.354 ICS 33.020

Keywords: in-vehicle emergency call device/system, road traffic accident, routing, minimum set of data, service support layer protocol, transport layer protocol, road accident emergency response system, emergency call, emergency service, emergency message

Editor *A.K. Bazdov*
Technical Editor *V.J. Fotieva*
Proof-reader M.S. *Bychnay*
Computer Imposition *E.E. Krugova*