ERA GLONASS Terminal
Communications Protocol
Service Support Protocol

Version: 1.6

Code: ERA GLONASS

Contents

1. INTRODUCTION

This document describes the Service Support Protocol for the ERA GLONASS system. The document gives all necessary data on the format and rules of message transfer of the Service Support Protocol and is to be used together with the document "ERA GLONASS Terminal, Communications Protocol, Transport Level" for the development of data transfer subsystems on the side of the ERA GLONASS terminals and the System Operator.

The document is intended for use by:

- terminal equipment manufacturers

- car manufacturers

- service developers and providers

- System Operator.

The OSI network model defines the following levels: physical, channel, network, transport, session, data presentation and applications. The correlation of the OSI network model, TCP/IP protocol stack and data transfer protocols of the ERA GLONASS system is presented in Table 1.

Table 1: Conformity of the levels of the OSI model, TCP/IP protocol stack and ERA GLONASS protocols

<table>
<tr><th colspan="2">OSI Model</th><th colspan="2">TCP/IP protocol stack</th><th rowspan="2">TCP/IP protocols</th><th rowspan="2">ERA GLONASS protocols</th></tr>
<tr><th>Level No.</th><th>Level description</th><th>Level No.</th><th>Level description</th></tr>
<tr><td>7</td><td>Applications</td><td rowspan="3">4</td><td rowspan="3">Applications</td><td rowspan="3">FTP, HTTP, POP3, IMAP, telnet, SMTP, DNS, TFTP</td><td rowspan="2">Service Support Protocol</td></tr>
<tr><td>6</td><td>Data presentation</td></tr>
<tr><td>5</td><td>Session</td><td>Transport Level</td></tr>
<tr><td>4</td><td>Transport</td><td>3</td><td>Transport</td><td>TCP, UDP</td><td>TCP</td></tr>
<tr><td>3</td><td>Network</td><td>2</td><td>Internetwork</td><td>IP</td><td>IP</td></tr>
<tr><td>2</td><td>Channel</td><td rowspan="2">1</td><td rowspan="2">Network access</td><td rowspan="2"></td><td></td></tr>
<tr><td>1</td><td>Physical</td><td></td></tr>
</table>

2. ABBREVIATIONS

Abbreviation	Description
ST	Subscriber Terminal
GLONASS	GLObal NAvigation Satellite System
UIM	User Interface Module
RA	Road Accident
TUI	Terminal User Interface
MSD	Minimum Set of Data
NIS	Navigation Information Systems
RAM	Random Access Memory
SW	Software
TP	Telematic Platform
V	Vehicle
ERA	Emergency Response to Accidents
EGTS	Era Glonass Telematics Standard

3. REFERENCES

[1] ERA GLONASS Terminal, Communications Protocol, Transport Level

[2] ERA GLONASS Terminal, Communications Protocol, Service Support Protocol, Fleet management service

[3] ERA GLONASS Terminal, Communications Protocol, Service Support Protocol, Emergency Response to Accidents

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

[5] ISO 639-2

[6] ITU-T E.164

4. PURPOSE OF THE SERVICE SUPPORT PROTOCOL

The Service support protocol is intended for data exchange between the system elements to ensure the functioning of information Services. Each Service is assigned to a separate Service which is the key element within the system built on the Protocol.

The Service support protocol has the following basic functions:

- Exchanging information messages containing the data to be processed by various Services, and also requests for the delivery of information by the Services
- Notifying on the results of Service Support data delivery and processing
- Matching data to a certain type of Service
- Identifying data attributes (quantity, type, structure, size, coding, etc.)

4.1 EXCHANGING INFORMATION MESSAGES

The basic structure of the Service support protocol comprising all data necessary for the processing of information or request for a service is the Record. Each record can have several subrecords containing the necessary data and defining the actions to be performed by the Service processing the subrecord.

4.2 NOTIFYING ON THE RESULT OF DATA DELIVERY AND PROCESSING AT THE SERVICE SUPPORT LEVEL

At the Service Support Level the notice of the dispatching party on the result of the data delivery and processing is provided by the mechanism of information record confirmation by means of special subrecords containing the identifier of the record received/processed.

4.3 MATCHING DATA TO A CERTAIN TYPE OF SERVICE

To match any record to a particular Service, the Service type identifier shall be used which defines the functional features and attributes of the data processed. The Service type is its identifier at intraplatform routing and is Protocol-unique.

4.4 IDENTIFYING DATA ATTRIBUTES

The data in the Service support protocol are recorded as a subrecord having a Service-unique identifier, and also a strictly defined data structure depending on the subrecord. The use of such data structure in the Protocol ensures unique identification of the data type, their physical meaning, size and packing.

5. DATA STRUCTURE DESCRIPTION

5.1 GENERAL STRUCTURE

The general structure of the Service support protocol forming a part of the Transport Level Protocol package can contain one or several sequential Records with different data structures intended for various Services. The general structure is shown in Fig. 1.

Service Support Level Data			
RID= 1 Record	**RID=2 Record**	**. . .**	**RID=N Record**

Fig. 1: General Service Support Data Structure

5.2 INDIVIDUAL RECORD STRUCTURE

5.2.1 RECORD COMPOSITION

An individual record of the Service support protocol has a Record Header and Record Data. The structure of an individual record is shown in Fig. 2.

Record Header	Record Data		
	Subrecord 1	. . .	Subrecord N

Fig. 2: Structure of an individual record of the Service Support Level

The Record Header contains parameters defining the types of Services of the addressee and the dispatcher, the record identifier, the object identifier (for example, the Terminal), the length of the data transferred, and also various flags defining the presence of optional parameters and the processing method.

The Record can contain one or several Subrecords defining the types and containing the transferred data.

5.2.2 RECORD STRUCTURE

The structure of an individual record of the Service Support Level is defined in Table 2.

Table 2: Format of an individual record of the Service support protocol

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Type	Data type	Byte
RL (Record Length)								M	USHORT	2

RN (Record Number)							M	USHORT	2
RFL (Record Flags)							M	BYTE	1
SSOD	RSOD	GRP	RPP		TMFE	EVFE	OBFE		
OID (Object Identifier)							O	UINT	4
EVID (Event Identifier)							O	UINT	4
TM (Time)							O	UINT	4
SST (Source Service Type)							M	BYTE	1
RST (Recipient Service Type)							M	BYTE	1
RD (Record Data)							M	BINARY	3...65498

- RL – parameter defining the data size from the RD field
- RN – record number. The values in the field vary by the cyclic counter rules in the range of 0 to 65535, i.e. at reaching 65535 the next value should be 0. The value from this field is used to confirm the entry
- RFL – contains bit flags defining the presence in the package of OID, EVID and TM fields characterizing the data contained in the entry
- SSOD – (Source Service On Device), bit flag defining the location of the source Service
  1 = Source Service is located on the ST side
  0 = Source Service is located on the TP side
- RSOD – (Recipient Service On Device), bit flag defining the location of the recipient service
  1 = Recipient service is located on the ST side
  0 = Recipient service is located on the TP side
- GRP – (Group), bit flag defining the appurtenance of data transferred to a specific group, its identifier stated in the OID field
  1 = data intended for a group
  0 = no group attribution
- RPP – (Record Processing Priority), bit field defining the record processing priority by the Service
  00 – highest
  01 – high
  10 – medium
  11 – low
- TMFE – (Time Field Exists), bit field defining the presence of a TM field in the package
  1 = TM field is present
  0 = no TM field
- EVFE – (Event ID Field Exists), bit field defining the presence of an EVID field in the package
  1 = EVID field is present
  0 = no EVID field
- OBFE – (Object ID Field Exists), bit field defining the presence of an OID field in the package
  1 = OID field is present

0 = no OID field

- OID – identifier of the object which generated the record or for which the record is intended (Unique ST identifier), or a group identifier (at GRP=1). If several records are sent sequentially from ST in one Transport Level Package intended for various services, but from the same object, the OID field may be present only in the first entry, and in the subsequent ones it can be omitted.
- EVID – unique event identifier. The EVID field sets a certain global identifier of the event and is used when it is necessary to establish logical relation of a set of several information entities with the unique event, provided that the entities can be distributed over different information packages and time. Provided that applied SW is able to combine all these entities together at the moment of giving the information on the event to the user. For example, if pressing an alarm button is connected to a series of photos, the EVID field should be specified in each service record connected with the event throughout the transmission of all entities connected with the given event, no matter how long it takes to transmit the whole pool of information.
- TM – time of generating the record by the Sender (seconds beginning with 0:00:00 AM, 1/1/2010 UTC). If there are several records related to one object and moment of time transmitted within one Transport Level Package, the TM time field can be transmitted only as part of the first record.
- SST – Type identifier of the Source Service having generated the record. For example, the Service processing navigation data at ST, Service of commands at TP etc.
- RST – Type identifier of the Service- recipient of the given record. For example, the Service processing the navigation data at TP, Service processing commands at ST etc.
- RD – field containing the information inherent in certain types of Service (one or several subrecords of the Service of the type specified in SST or RST fields, depending on the kind of the information transmitted).

5.3 GENERAL STRUCTURE OF SUBRECORDS

The format of a separate subrecord of the Service support protocol is defined in Table 3.

Table 3: Format of a separate subrecord of the Service support protocol

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

- SRT – subrecord type (subtype of the data transmitted within the limits of the general set of types of one Service). Type 0 – special, reserved for the subrecord confirming the data for each service. Specific values of the type numbers of the subrecords are defined by the

Service logic. The Protocol only specifies that this number should be present, and that the zero identifier is reserved.

- SRL – subrecord data length in bytes in the SRD field
- SRD – subrecord data. The contents of this field is specific for each combination of the Service identifier and subrecord type

There shall be a confirmation sent to each information record of the Service Support Level containing a subrecord with the information on the identifier of the record confirmed and the result of its processing. The description and format of confirmation are presented in p. 6.2.1 of this document. The algorithm of work of the mechanism of conformation of the Service Support Level is shown in Fig. 3.

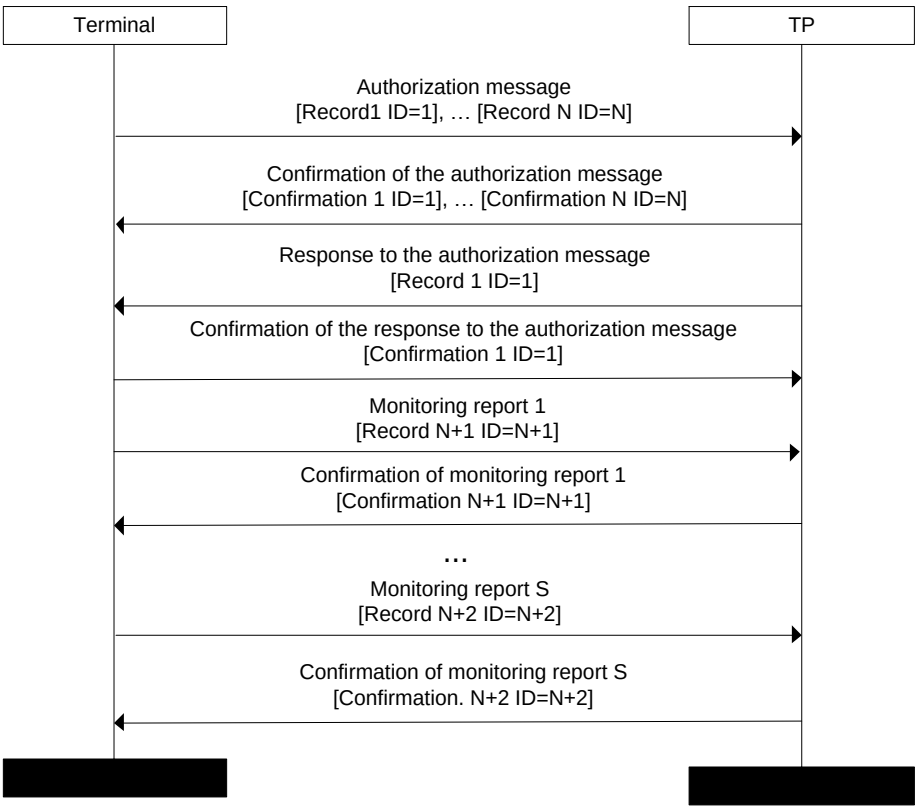

Fig. 3: Message exchange diagram

Each message of the Protocol contains a header and a check sum of the Transport Level and one or more records of the Service Support Level. Provided that one message can contain both information records and confirmations of records received earlier.

6. DESCRIPTION OF SERVICES

6.1 LIST OF SERVICES

A Service herein means a TP infrastructure element ensuring functional execution of the algorithm of a particular Service with the use of the Protocol described. The list of the Services supported, their decimal identifiers and their description are shown in Table 4.

Table 4: List of Services supported by the Protocol

Code	Name	Description	OE*	RS**	RS-OS***
1	EGTS_AUTH_SERVICE	This service is intended for ST authentication at TP. When using the TCP/IP protocol ST should pass this procedure, and further interaction is possible only if the procedure is completed successfully	+		+
2	EGTS_TELEDATA_SERVICE	This service is intended for processing of telematic information (coordinate data, sensor actuation, etc.) supplied by ST. The service is described in a separate document [2]	+		+
4	EGTS_COMMANDS_SERVICE	This service is intended for processing of control and configuration commands, information messages and statuses transmitted between ST, TP and operators	+	+	+
9	EGTS_FIRMWARE_SERVICE	This service is intended for transfer to ST of the configuration and software (SW) for the hardware of ST, and also various peripherals connected to ST and supporting the SW remote update option	+	+	+
10	EGTS_ECALL_SERVICE	This service is intended for the ERA functional performance. The service is described in a separate document [3]	+	+	+

* - optional equipment

** - standard system, ERA only

*** - standard system with optional services

6.2 EGTS_AUTH_SERVICE

This type of Service is employed for the ST authentication procedure at TP, and also receipt of ST account data and information on the infrastructure at ST (SW structure and versions of modules, blocks, peripherals, vehicle information). The service should be used by ST only in case of work under the TCP/IP protocol after each new connection with TP is created.

This section applies only to aftermarket/retrofit equipment and standard automobile systems supporting the optional services developed to NIS requirements.

The section does not apply to standard systems which support only the basic accident response service.

6.2.1 DESCRIPTION OF SUBRECORDS

The list of subrecords used by EGTS_AUTH_SERVICE is presented in Table 5.

Table 5: List of subrecords used by EGTS_AUTH_SERVICE

Code	Name	Description
0	EGTS_SR_RECORD_RESPONSE	The subrecord is intended to confirm the record processing of the Service support protocol. This subrecord is supported by all Services
1	EGTS_SR_TERM_IDENTITY	The subrecord is used by ST when requesting authorization at TP and contains the ST account data
2	EGTS_SR_MODULE_DATA	The subrecord is intended for transmission to TP of information on the infrastructure at ST, structure, state and parameters of ST units and modules. This subrecord is optional and the ST developer independently decides whether the fields should be filled in and the subrecord should be sent. One subrecord describes one module. One record can be used to sequentially transmit several such subrecords, which allows to transmit data on separate components of all ST hardware and peripherals
3	EGTS_SR_VEHICLE_DATA	The subrecord is intended for ST to transmit to TP information on the relevant vehicle
6	EGTS_SR_AUTH_PARAMS	The subrecord is intended for TP supply ST with data on the method and parameters of coding required for further interaction
7	EGTS_SR_AUTH_INFO	The subrecord is intended to transmit to TP authenticated

		data of ST using the parameters which have been earlier transmitted by TP for data coding
8	**EGTS_SR_SERVICE_INFO**	**The subrecord is intended to inform the receiving party, ST or TP, depending on the transmission direction, on the Services supported, and also to request a certain set of the Services necessary (from ST to TP)**
9	**EGTS_SR_RESULT_CODE**	**The subrecord is used by TP to inform ST on the results of the ST authentication procedure.**

6.2.1.1 EGTS_SR_RECORD_RESPONSE SUBRECORD

The subrecord structure is shown in Table 6.

Table 6: EGTS_SR_RECORD_RESPONSE subrecord format

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

EGTS_SR_RECORD_RESPONSE subrecord fields:

- CRN – confirmation record number (RN field from the record processed);
- RST – record processing status.

  When confirmation is received by the Dispatcher it shall analyze the RST field of the SR_ RECORD_RESPONSE subrecord and, if a successful processing status is received, deletes the record from the databank, otherwise respective actions will be taken in case of error and depending upon the reason thereof.

6.2.1.2 EGTS_SR_TERM_IDENTITY SUBRECORD

The subrecord structure is shown in Table 7.

Table 7: EGTS_SR_TERM_IDENTITY Service EGTS_AUTH_SERVICE subrecord format

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

EGTS_SR_TERM_IDENTITY subrecord fields:

- TID – unique identifier assigned at programming ST. The value 0 in this field means that ST has not been configured or has not been completely configured. The given identifier is assigned by the operator and provides unique description for the ST account data set. TID is assigned at ST installation as optional equipment and transmission of the ST account data to the operator (IMSI, IMEI, serial_id). In case ST is used as a standard device, TID is made known to the operator by the vehicle manufacturer together with the vehicle registration data (VIN, IMSI, IMEI).
- HDIDE – bit flag defining the presence of the HDID field in the subrecord (if the bit=1, the field is transmitted, if the bit =0, the field is not transmitted)
- IMEIE – bit flag which defines the presence of the IMEI field in the subrecord (if the bit=1, the field is transmitted, if the bit =0, the field is not transmitted)
- IMSIE – bit flag defining the presence of the IMSI field in the subrecord (if the bit=1, the field is transmitted, if the bit =0, the field is not transmitted)
- LNGCE – bit flag defining the presence of the LNGC field in the subrecord (if the bit=1, the field is transmitted, if the bit =0, the field is not transmitted)
- SSRA – bit flag intended to define the Service use algorithm (if the bit equals 1, a "simple" algorithm is used, if it equals 0, the algorithm of service use “requests” is used)
- NIDE – bit flag defining the presence of the NID field in the subrecord (if the bit=1, the field is transmitted, if the bit =0, the field is not transmitted)
- BSE – bit flag defining the presence of the BS field in the subrecord (if the bit=1, the field is transmitted, if the bit =0, the field is not transmitted)
- MNE - bit flag defining the presence of the MSISDN field in the subrecord (if the bit=1, the field is transmitted, if the bit =0, the field is not transmitted)
- HDID –"home" TP identifier (detailed account information on the terminal is stored on the given TP)
- IMEI – mobile device (modem) identifier. When it is impossible to define this parameter, ST should fill in this field with the 0 value for all 15 symbols
- IMSI – mobile subscriber identifier. When it is impossible to define this parameter, the device should fill in this field with the 0 value for all 16 symbols
- LNGC – code of the language to be preferred when using at ST according to ISO 639-2, for

example, “rus” – Russian

- NID – identifier of the operator network in which ST is registered. 20 low bits are used. It consists of a pair of MCC-MNC codes (on the basis of ITU-T E.212 recommendations). The structure of the NID field is shown in Table 8
- BS – maximum ST receipt buffer size in bytes. The size of each information package transmitted to ST should not exceed the given value. The value of the BS field can vary (1024, 2048, 4096), and depends on the hardware and software versions of specific ST
- MSISDN – mobile subscriber’s telephone number. When it is impossible to define this parameter, the device should fill in this field with the 0 value for all 15 symbols (the format is described in [6]).

The transmission of the HDID field is defined by ST settings and is recommended where it is possible to connect ST to TP other than the "home" one, for example, at the use of a territorially distributed TP network. If only one "home" TP is used, no HDID transmission is required.

The “simple” Service use algorithm means that there all services are accessible for ST, and in this mode ST is allowed to send the data for the service required at once. Depending on the authorization in use on TP for the given ST, in reply to the service data package a confirmation record can be returned with the corresponding error flag. In systems with simple allocation of rights to the use of Services it is recommended to employ the "Simple" algorithm. It reduces the volume of the traffic and the time of ST authorization.

The algorithm of "requests" for the use of services means that, before using a Service (sending data), ST should receive from TP information about the Services available for use. The request for the use of services can be sent both at the stage of authorization and after it. At the authorization stage, the request for the use of a particular service is made by adding subrecords of the type of SR_SERVICE_INFO and setting bit 7 of the SRVP field to 1. After the authorization procedure the service use request can also be made with the help of SR_ SERVICE_INFO subrecords.

Table 8: NID field format of the EGTS_SR_TERM_IDENTITY subrecord, EGTS_AUTH_SERVICE

Bits 20…23	Bits 10…19	Bits 0…9	Type	Data type	Bytes
-	MCC (Mobile Country Code)	MNC (Mobile Network Code)	M	BINARY	3

The set of MCC and MNC specifies the unique identifier of the GSM, CDMA, TETRA, UMTS network operator, as well as some satellite network operators.

NID field parameters of the EGTS_SR_TERM_IDENTITY subrecord:

- MCC – country code
- MNC – home mobile network code

6.2.1.3 EGTS_SR_MODULE_DATA SUBRECORD

The subrecord structure is shown in Table 9.

Table 9: EGTS_SR_MODULE_DATA subrecord format for EGTS_AUTH_SERVICE

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Type	Data type	Bytes
MT (Module Type)								M	BYTE	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)								O	STRING	0 … 32
D (Delimiter)								M	BYTE	1
DSCR (Description)								O	STRING	0 … 32
D (Delimiter)								M	BYTE	1

SR_MODULE_DATA subrecord fields:

- MT – module type, defines the functional appurtenance of the module (1 – basic module; 2 – input-output module; 3 – navigation receiver module; 4 – wireless communication module). The numbering of the module types given here is the one recommended. At actual implementation of the authorization Service other numbering of the types, including all external peripheral controllers, can be used
- VID – manufacturer's code
- FWV – module hardware version (upper byte – number to the left of the separator– major version, lower byte – number to the right of the separator – minor version; for example, version 2.34 will be presented as 0x0222)
- SWV – module software version (upper byte – number to the left of the separator, lower byte – number to the right of the separator)
- MD – module software update code
- ST – status (1 - on, 0 - off,> 127 – malfunction - see Processing result codes)
- SRN – module series number
- D – string parameter divider (always equals 0);
- DSCR – brief description of the module.

6.2.1.4 EGTS_SR_VEHICLE_DATA SUBRECORD

The subrecord structure is shown in Table 10. If the standard system with optional services is used, this subrecord have to be transmitted together with EGTS_SR_TERM_IDENTITY. In this case the VIN field data is used for ST identification.

Table 10: EGTS_SR_VEHICLE_DATA subrecord format of EGTS_AUTH_SERVICE

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

EGTS_SR_VEHICLE_DATA subrecord fields:

- VIN – Vehicle Identification Number (the structure is described in ISO 3779)
- VHT – vehicle type

  Bit 31 - 4: not used
  Bit 3-0:
  0001 – passenger car (Class M1)
  0010 = bus (Class M2)
  0011 = bus (Class M3)
  0100 = light-duty truck (Class N1)
  0101 = heavy-duty truck(Class N2)
  0110 = heavy-duty truck(Class N3)
  0111 = motorcycle (Class L1e)
  1000 = motorcycle (Class L2e)
  1001 = motorcycle (Class L3e)
  1010 = motorcycle (Class L4e)
  1011 = motorcycle (Class L5e)
  1100 = motorcycle (Class L6e)
  1101 = motorcycle (Class L7e)
- VPST – Vehicle Propulsion Storage Type

  If all bits are zero, the type is not set
  Bit 31 - 6: not used
  Bit 5: 1 = hydrogen
  Bit 4: 1 = electric power (above 42 v and 100 Ah)
  Bit 3: 1 = liquid propane (LPG)
  Bit 2: 1 = condensed natural gas (CNG)
  Bit 1: 1 = diesel fuel
  Bit 0: 1 = gasoline

6.2.1.5 EGTS_SR_AUTH_PARAMS SUBRECORD

The subrecord structure is shown in Table 11.

Table 11: EGTS_SR_AUTH_PARAMS subrecord format of EGTS_AUTH_SERVICE

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Type	Data type	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

EGTS_SR_AUTH_PARAMS subrecord fields:

- EXE – bit flag defining the presence of the EXP field followed by delimiter D (if 1, the fields are present);
- SSE – bit flag defining the presence of the SS field followed by delimiter D (if 1, the fields are present);
- MSE – bit flag defining the presence of the MSZ field (if 1, the field is present);
- ISLE – bit flag defining the presence of the ISL field (if 1, the field is present);
- PKE – bit flag defining the presence of the PKL and PBK fields (if 1, the fields are present);
- ENA –bit field defining the required package encryption algorithm. If this field contains the 0 0 value, no encryption is employed, and the EGTS_SR_AUTH_PARAMS subrecord contains only one byte, otherwise, depending on the algorithm type, the presence of additional parameters is defined by other bits of the FLG field
- PKL – public key length, bytes
- PBK – public key data
- ISL – resulting length of the identification data
- MSZ – parameter applied during encryption
- SS – special server byte sequence used during encryption
- D – string parameters delimiter (always equals 0)
- EXP – special sequence used during encryption.

If the requested encryption algorithm (where required) is supported, the authorizing party generates and sends an EGTS_SR_AUTH_INFO record coded using the said algorithm. Provided that bits 11 and 12 in the KEYS field of the Transport Level header are set at the corresponding values, and all subsequent data exchange is performed using the coding.

If the requested encryption algorithm is not supported, the initiating party sends an EGTS_SR_ RECORD_RESPONSE subrecord with the corresponding error flag.

6.2.1.6 EGTS_SR_AUTH_INFO SUBRECORD

The subrecord structure is shown in Table 12.

Table 12: EGTS_SR_AUTH_INFO subrecord format for EGTS_AUTH_SERVICE

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Type	Data type	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

EGTS_SR_AUTH_INFO subrecord fields:

- UNM – user name
- D – string parameters delimiter (always equals 0)
- UPSW – user password
- SS – special service sequence of bytes transmitted in the EGTS_SR_AUTH_PARAMS subrecord (optional field, its presence depending on the encryption algorithm employed).

6.2.1.7 EGTS_SR_SERVICE_INFO SUBRECORD

The subrecord structure is shown in Table 13.

Table 13: EGTS_SR_SERVICE_INFO subrecord format for EGTS_AUTH_SERVICE

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

EGTS_SR_SERVICE_INFO subrecord fields:

- ST – server type defining the functional appurtenance (for instance, EGTS_TELEDATA_SERVICE, EGTS_ECALL_SERVICE etc.)
- SST – defines the current state of the service (Table14)
- SRVP – defines the service parameters
- SRVA – (Service Attribute) bit flag, service attribute

  0 = supported service
  1 = requested service
- SRVRP – (Service Routing Priority) bit field, the priority in translation of data thereon (in case of system scaling and several copies of the Applications for a single service type) defined by bits 0 and 1

  00 =highest;
  01 = high;
  10 = medium;
  11 = low;

Table 14: List of possible Service states

Code	Name	Description
0	EGTS_SST_IN_SERVICE	In service (access allowed)
128	EGTS_SST_OUT_OF_SERVICE	Out of service (disconnected)
129	EGTS_SST_DENIED	Service denied
130	EGTS_SST_NO_CONF	Service not configured
131	EGTS_SST_TEMP_UNAVAIL	Service temporary unavailable

6.2.1.8 EGTS_SR_RESULT_CODE subrecord

The subrecord structure is shown in Table 15.

Table 15: EGTS_SR_ RESULT_CODE subrecord for EGTS_AUTH_SERVICE

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

EGTS_SR_SERVICE_INFO subrecord fields:

- RCD – code defining the result of authorization. See Appendix 1 [1]

6.2.2 AUTHORIZATION PROCEDURE

To ensure ST operation in the operator infrastructure it needs a unique identifier - UNIT_ID, which should match certain values of IMEI, IMSI and some other ST account data necessary for interaction in the operator system.

ST can be configured in any of the following ways:

- In the passive mode after the «Additional Functions» button is pressed and ST is registered in the GSM or UMTS network, the cellular operator infrastructure traces the occurrence of a new device and initiates dispatch of a coded SMS with the account data to it. Encryption is made using a key and an algorithm known to the ST and kept by the time of configuring in the databank of the operator. To define encryption keys and algorithms at ST the corresponding fields from the header of the Transport Level protocol are used, as well as the data on the keys embedded in the ST memory. The account data are transmitted in the form of a configuration file using the EGTS_SR_SERVICE_FULL_DATA subrecord or EGTS_SR_SERVICE_PART_DATA subrecord of EGTS_FIRMWARE_SERVICE. The configuration file should contain: EGTS_GPRS_APN parameter (parameters of the access point to establish a GPRS session), EGTS_SERVER_ADDRESS parameter defining the address and port of the server with which a TCP/IP connection is to be established, ST UNIT_ID unique identifier. The configuration file can also contain other parameters necessary for ST operation. ST further decodes the SMS received, checks the correctness of the data structures, calculates and compares the check sums with the values received in the message. If the decryption and checking are successful, ST establishes a GPRS session and connects to the specified server through TCP/IP. After authentication is complete successful configuration is confirmed in the form of a EGTS_SR_RECORD_RESPONSE subrecord with the EGTS_PC_OK code in reply to the EGTS_SR_SERVICE_FULL_DATA or EGTS_SR_SERVICE_PART_DATA record of EGTS_FIRMWARE_SERVICE. An algorithm of such ST configuring is presented in fig. 4.
- After ST is registered in the GSM or UMTS network a GPRS session and TCP/IP connection with the server is initiated using the address stored in the ST memory. During the authentication procedure the operator infrastructure analyzes the TID parameter from the EGTS_SR_TERM_IDENTITY subrecord (Table 7). If TID value equals 0, the configuring procedure is started through EGTS_FIRMWARE_SERVICE as described in the preceding scenario, by sending the configuration file using the EGTS_SR_SERVICE_FULL_DATA or EGTS_SR_SERVICE_PART_DATA subrecord. Further after the configuration file confirmation is received from ST, the authorization result with the EGTS_PC_ID_NFOUND code is returned, specifying that no TID=0 has been found in the system. After that the server, without breaking off the connection with ST, expects repeated ST authorization, but already with the correct TID parameter. An algorithm of such ST configuring is shown in drawing 5.

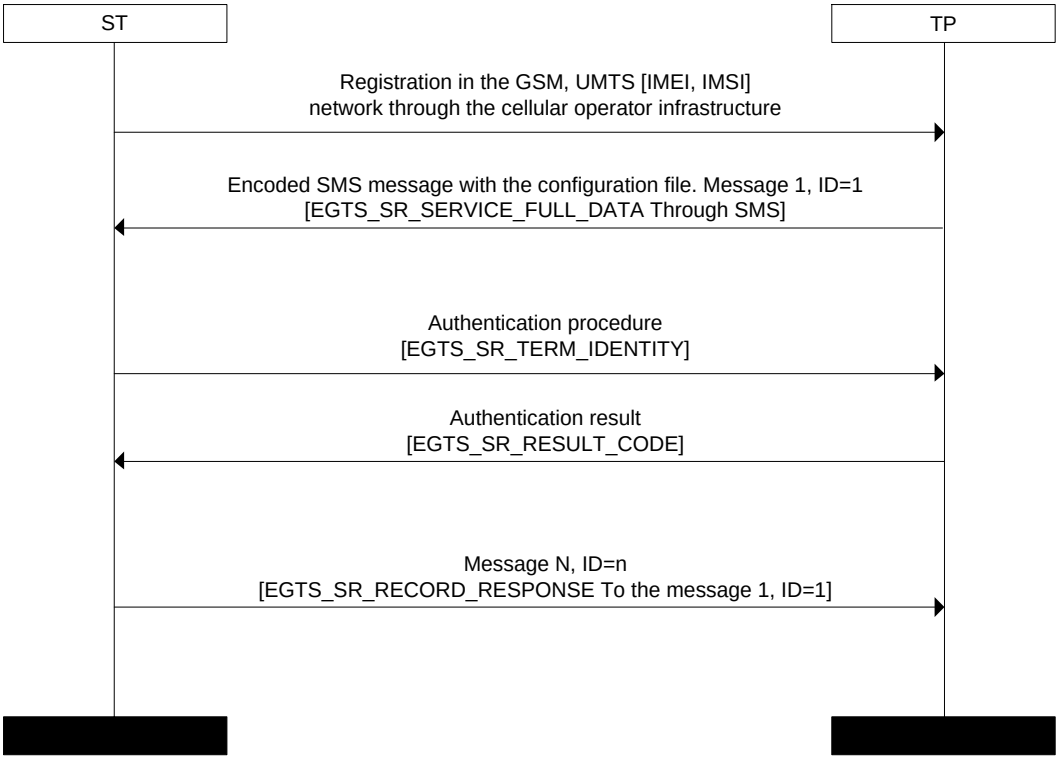

Fig. 4: Algorithm of ST configuration through SMS

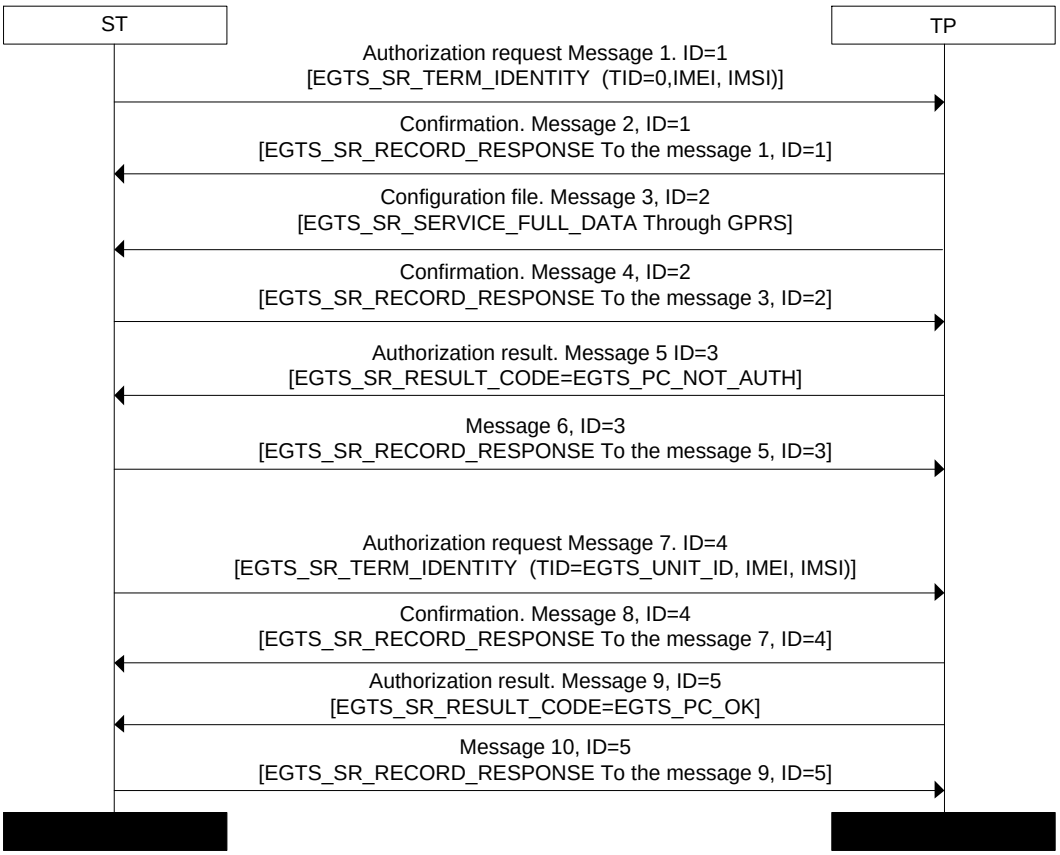

Fig. 5: Algorithm for ST configuration through GPRS

If authorization is successful, TP, depending on the service use request algorithm, adds before the EGTS_SR_RESULT_CODE subrecord other subrecords of the EGTS_SR_SERVICE_INFO type, defining the composition of the services allowed for ST and supported by TP. This means that promptly after authorization ST is able to use only the listed Services, even if a “simple” algorithm for Service use support is meant.

If the Service usage "request" algorithm is employed, ST cannot use the Services without the relevant permission received from TP. Provided that the permission for some requested services can arrive later. For example, when the services are on remote TPs, responses to the requests can arrive from these TPs asynchronously. In such case TP, using available routing data, sends an asynchronous request for the use of the remote TP Services if the HDID identifier is specified in the EGTS_SR_TERM_IDENTITY subrecord at ST authorization.

The message exchange algorithm at the stage of ST authorization at TP is presented in the diagram given in Fig. 6.

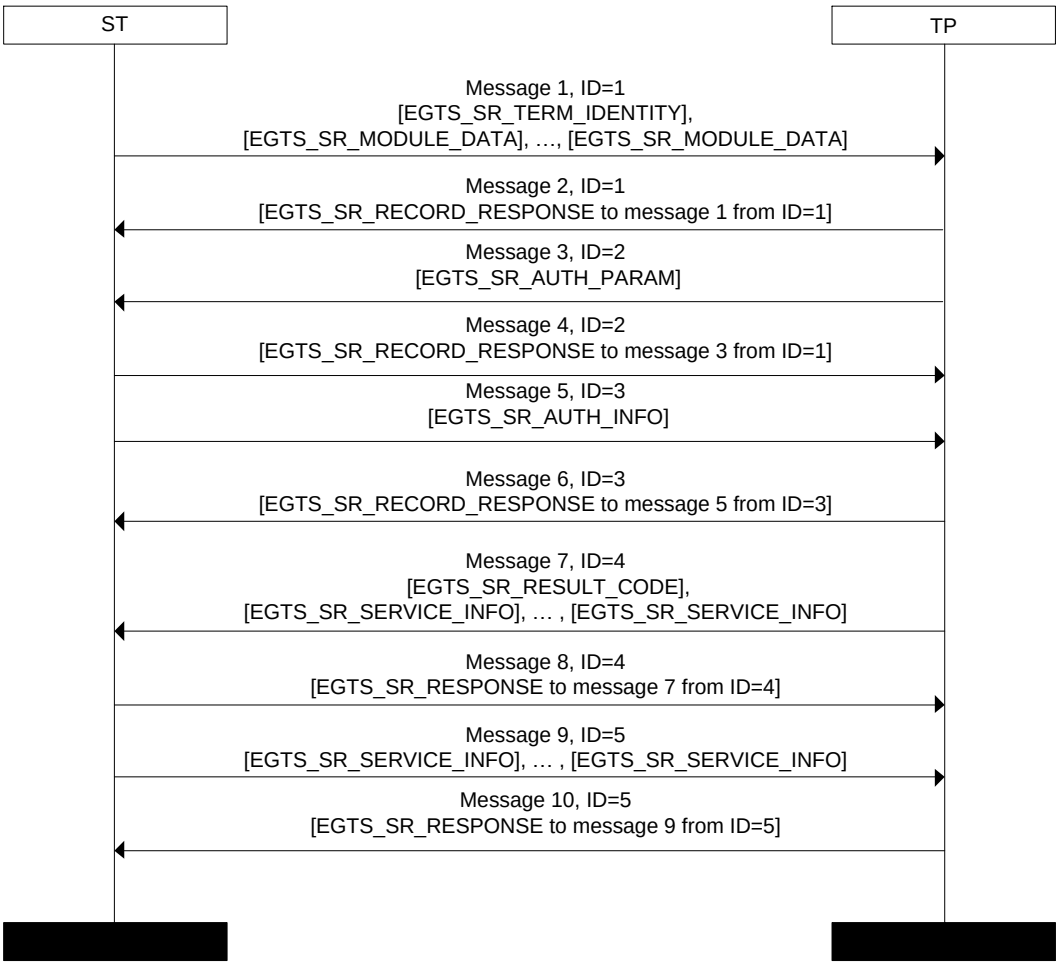

Fig. 6: Message exchange at the stage of ST authorization at TP

After ST successfully connects to TP using the TCP/IP protocol, ST should be authorized. To transfer initial authentication data, ST is to send a Message containing the EGTS_SR_TERM_IDENTITY subrecord (Message 1) within the time EGTS_SL_NOT_AUTH_TO (see Table 38).

On receiving a message with a EGTS_SR_TERM_IDENTITY subrecord, TP responds with Message 2 confirming receipt of EGTS_SR_RECORD_RESPONSE in reply to the record with the ID=1 identifier. Further, depending on the settings (whether encryption, additional authorization algorithm is used), TP sends a package (Message 3) with a EGTS_SR_AUTH_PARAM subrecord containing the parameters necessary for the encryption and/or extended authorization algorithm. If no encryption or extended authorization algorithm is used, TP can send an EGTS_SR_RESULT_CODE subrecord with the result of ST authorization instead of the EGTS_SR_AUTH_PARAM subrecord.

ST further sends Message 4 with ID=2 and EGTS_SR_RECORD_RESPONSE to Message 3 with ID=2. When using an expanded authorization algorithm and/or encryption, ST transmits Message 5 employing the rules of encryption specified in message 3 from TP and containing an EGTS_SR_AUTH_INFO subrecord with the data for the extended authorization.

After EGTS_SR_AUTH_INFO is received, TP sends Message 6 with confirmation in reply to message 5 with ID=3 and performs the authorization procedure. TP generates Message 7 with the authorization result as an EGTS_SR_RESULT_CODE subrecord, and also in case of successful authorization can add information on the services allowed for the given ST in the form of EGTS_SR_SERVICE_INFO subrecords.

Then, ST generates Message 8 with confirmation in reply to Message 7 with ID=4. ST can generate Message 9 and add EGTS_SR_SERVICE_INFO subrecords containing the information on the requested Services (if the “on-request” Service use procedure is employed) and/or Services supported at ST.

TP further generates Message 10 with confirmation in reply to Message 9 with ID=5.

At this point the authorization stage is completed and ST passes on to the stage of exchange of information messages with TP according to the operating mode established at ST.

In case the authorization procedure is unsuccessful (incorrect ST authentication data, denial of ST access to TP, etc.), after the message containing a EGTS_SR_RESULT_CODE subrecord with the corresponding code is dispatched, TP shall break off the TCP/IP connection established by the terminal.

6.3 EGTS_COMMANDS_SERVICE

This service is intended for processing of commands, messages and confirmations exchanged between ST, TP and client applications.

6.3.1 SUBRECORD DESCRIPTION

For interaction within the framework of the given Service, one subrecord is used - EGTS_SR_COMMAND_DATA, which description and code are shown in Table 16.

Table 16: List of subrecords for EGTS_COMMAND_SERVICE

Code	Name	Description
0	**EGTS_SR_RECORD_RESPONSE**	**Subrecord used to confirm the Service support protocol record processing. This subrecord is to be supported by all Services**
51	**EGTS_SR_COMMAND_DATA**	**Subrecord used by ST and TP to transmit commands, information messages, delivery confirmations, command execution confirmations, message reading confirmations.**

6.3.1.1 EGTS_SR_COMMAND_DATA subrecord

The subrecord structure is shown in Table 17.

Table 17: EGTS_SR_COMMAND_DATA subrecord format for EGTS_COMMANDS_SERVICE

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Type	Data type	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 (Authorization Code Length)								O	BYTE	1
AC (Authorization Code)								O	BINARY	0 … 255
CD (Command Data)								O	BINARY	0 … 65205

- CT – Command type
  0001 = CT_COMCONF - confirmation of receipt, processing or execution of a command

0010 = CT_MSGCONF - confirmation of receipt, display and/or processing of an information message
0011 = CT_MSGFROM - information message from ST
0100 = CT_MSGTO - information message to be displayed by ST monitor
0101 = CT_COM - command to be executed on ST
0110 = CT_DELCOM - deletion of an earlier transmitted command queuing for execution
0111 = CT_SUBREQ - additional subrequest for execution (to the command transmitted earlier)
1000 = CT_DELIV - confirmation of delivery of a command or information message

- CCT – confirmation type (for CT_COMCONF, CT_MSGCONF, CT_DELIV command types)

  0000 = CC_OK - successful execution, affirmative response;
  0001 = CC_ERROR - processing error
  0010 = CC_ILL - command cannot be executed as it is not included in the list of commands allowed (defined by the protocol) or the execution of the command is not authorized
  0011 = CC_DEL - command deleted successfully
  0100 = CC_NFOUND - command to be deleted is not found
  0101 = CC_NCONF - successful execution, negative response
  0110 = CC_INPROG - command processing in progress, additional time is necessary (execution result is not yet known)
- CID – command or message identifier. The value from this field is to be used by the party processing /performing the command or message to generate confirmation. The confirmation is to contain the same value in the CID field as that in the command or message at the time of dispatch.
- SID – command or confirmation originator identifier (internal user identifier of Fleet Management System for example)
- ACFE – (Authorization Code Field Exists) bit flag defining the presence of the ACL and AC fields in the subrecord

  1 = ACL and AC fields present in the subrecord
  0 = no ACL or AC fields present in the subrecord
- CHSFE – (Charset Field Exists) bit flag defining the presence of the CHS field in the subrecord

  1 = CHS field present in the subrecord
  0 = no CHS field present in the subrecord
- CHS – character encoding used in the CD field which contains the command core. When this field is not used, the default character set shall be CP-1251. The following values of the CHS field (decimal view) are defined:

  0 = CP-1251
  1 = IA5 (CCITT T.50)/ASCII (ANSI X3.4)
  2 = binary data
  3 = Latin 1 (ISO-8859-1)
  4 = binary data
  5 = JIS (X 0208-1990)
  6 = Cyrllic (ISO-8859-5)
  7 = Latin/Hebrew (ISO-8859-8)
  8 = UCS2 (ISO/IEC-10646)
- ACL – length in bytes of the AC field containing the authorization code at the side of the recipient.
- AC – authorization code used at the side of the recipient (ST), ensuring restricted access to execution of certain commands. If the code specified in the field does not match the expected value, ST should respond to such command or message by a conformation with the CC_ILL type.
- CD – command core, parameters, data returned to the request command using the character set

from the CHS field or the default value. The field size is defined on the basis of the overall length of the Service support protocol record and the length of the preceding fields in the subrecord. The command format is described in Table 18. A list of commands, their format and description is given in p.6.4.2. This field can have a zero length (be absent) if no data are transmitted in response to a command or message for ST.

Table 18: Terminal command format

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Type	Data type	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

- ADR – Address of the module for which this command is intended. The address is defined based on the initial configuration of ST or from the list of modules which can be received, when the terminal is registered through EGTS_AUTH_SERVICE and transmission of EGTS_SR_MODULE_DATA subrecords.
- SZ – Memory size for the relevant parameter (to be used together with ACT=2). When adding a new parameter in ST, this field indicates that for the new parameter 2^{SZ} byte memory in ST is required.
- ACT – action description, used for command type, field CT=CT_COM of the EGTS_SR_COMMAND _DATA subrecord. The field can have the following values:

0 = command parameters. Used to transmit parameters for a command defined by the code from the CCD field

1 = value request. Used to request information stored on ST. The requested parameter is defined by the code from the CCD field

2 = value setting. Used to set a new value for a specific parameter in ST. The parameter set is defined by the code from the CCD field, and its value is defined from the DT field

3 = addition of a new parameter in ST. The code of the new parameter is indicated in the CCD field, its type - in the SZ field, and value - in the DT field

4 = deletion of an existing parameter from ST. The code of the deleted parameter is indicated in the CCD field

- CCD – command code from Table 35 at ACT=0 or parameter code from Table 35 at ACT=1. 4
- DT – requested data or parameters necessary for the command execution. The data are entered in this field in a format depending on the type of the command specified in p. 5 [1].

Confirmation of a command transmitted earlier at CT=CT_COMCONF, in case the accompanying information is transmitted from ST, has the format described in Table 19. The structure described is contained in the CD field (Table 17).

Table 19: Terminal command confirmation format

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

- ADR – address of the module from which confirmation is transmitted. The address is defined based on the initial ST configuration or on the list of modules which can be obtained when the terminal is registered through EGTS_AUTH_SERVICE and transmitting EGTS_SR_MODULE_DATA subrecords.
- CCD – code of a command, message from Table 20 or parameter from Table 22, according to which the accompanying information is transmitted in the DT field.
- DT – accompanying data, their type and composition to be defined by the CCD field value. A list and composition of the accompanying data transmitted in the confirmation to certain commands is given in Table 21.

6.3.2 DESCRIPTION OF COMMANDS, PARAMETERS AND CONFIRMATIONS

Table 20 contains a list and description of ST commands.

Table 20: List of commands for ST

Command name	Code	Parameter type, number and limits	Description
EGTS_RAW_DATA	0x0000	BINARY (up to 65200 bytes)	Command for transmission of arbitrary data. It is used, for example, to transmit commands, messages and data to remote terminal units, modules connected to the Terminal mainframe, in a format defined by the relevant module. Provided that the terminal need not analyze the data from the DT field (Table 33) but is to transmit them unchanged to the address defined by the ADR field (Table 33)

EGTS_TEST_MODE	0x0001	BYTE	Terminal test start/stop command. 1 – start of test, 0 – end of test.
EGTS_TEST_GET_ERRORS	0x0004	-	Error code request
EGTS_TEST_CLEAR_ERRORS	0x0005	-	Error code clearing. To process this command the operator shall set correct values of the ACL and AC fields from Table 32.
EGTS_CONFIG_RESET	0x0006	-	Return to manufacturer's factory settings. All user-defined parameters are deleted, the system returns to factory settings. To process this command the operator shall set correct values of the ACL and AC fields from Table 32.
EGTS_SET_AUTH_CODE	0x0007	BINARY	Installation of the authorization code at ST. To process this command the operator shall set correct values of the ACL and AC fields from Table 32. After the command is acknowledged, ST will use the new data for comparison with the value from the AC field in some commands sent to ST.
EGTS_RESTART	0x0008	-	Command to restart the basic SW of ST. To process this command the operator shall set correct values of the ACL and AC fields from Table 32

Table 21: List of confirmations in response to ST commands and messages

Command name	Code	Parameter type and number	Description
EGTS_RAW_DATA	0x0000	BINARY (up to 65200 bytes)	Data supplied by peripherals, modules connected to the Terminal mainframe, in the module-defined format
EGTS_SELF_TEST_ RESULT	0x0002	STRING	Self-test result message. Generated by ST automatically without the operator's request
EGTS_TEST_GET_ERRORS	0x0004	BINARY (16 byte)	List of error codes of the Terminal units, modules and subsystem statuses

Table 22: ST parameter list

Parameter name	Code	Parameter type	Default value	Description
Radio mute (only for optional equipment configuration)				
EGTS_RADIO_MUTE_DELAY	0x0201	INT	500	Delay between the radio mute signal setting and the sound replay commencement, msec
EGTS_RADIO_UNMUTE_DELAY	0x0202	INT	500	Delay between the radio mute signal removal and the sound replay end, msec
General settings				
EGTS_GPRS_APN	0x0203	STRING	""	Parameter defining the GPRS access point
EGTS_SERVER_ADDRESS	0x0204	STRING	""	Server address and port for TCP/IP communication
EGTS_SIM_PIN	0x0205	INT	0	SIM card PIN code
EGTS_AUTOMATIC_REGISTRATION	0x0207	BOOLEAN	1	Flag allowing SIM automatic network registration after power-up

EGTS_SELFTEST_INTERVAL	0x0208	INT	0	Self test interval. If the value is set at 0, no self-test is performed.
EGTS_POST_TEST_REGISTRATION_TIME	0x0209	INT	0	Time interval during which the terminal remains registered in the network after self-test results are sent to the system operator, sec
EGTS_TEST_MODE_END_DISTANCE	0x020A	INT	300	Distance at which the test mode is automatically turned off, meters
EGTS_GARAGE_MODE_END_DISTANCE	0x020B	INT	300	Distance at which the “garage” mode is automatically turned off, meters
EGTS_GARAGE_MODE_PIN	0x020C	ENUM {NONE=0, PIN_1 =1, .. PIN_8=8}	0	Line signaling that the system is in the “in garage” mode. NONE – no mode signaling PIN_X – PIN_X line activated when the system is in the given mode
EGTS_TEST_MODE_WATCHDOG	0x020E	INT	10	Watchdog interval in the test mode, min
Configuration and service configuration data				
Data package transmission				
EGTS_USE_GPRS_WHITE_LIST	0x0230	BOOLEAN	1	Parameter indicating the need to use GPRS_WHITE_LIST when transmitting data in a package
EGTS_GPRS_WHITE_LIST	0x0231	ARRAY OF STRING [20]	“”, “”, ””, ””, “”, “”, “”, ””, ””, “”, “”, “”, ””, ””, “”, “”, “”, ””, ””, “”	List of networks in which data package transmission is allowed. If GPRS_WHITWE_LIST is empty, data package transmission is prohibited, MCC (Mobile Country Code) 3 characters +MNC(Mobile Network Code) 3 characters
Test mode				
EGTS_TEST_REGISTRATION_TIMEOUT	0x0241	INT	5	If the terminal was registered in the network by pressing the optional services button, and the test session startup command was not received from the system operator within the given interval, the terminal shall

				terminate network registration, min
EGTS_TEST_REGISTRATION_PERIOD	0x0242	INT	0	If the terminal was registered in the network by pressing the optional services button, subsequent registration of the terminal in the network by pressing the optional services button is possible only after the given interval. If the value is set to 0, no restrictions on subsequent registration of the Terminal in the network are imposed, min
Other parameters				
EGTS_GNSS_POWER_OFF_TIME	0x0301	INT	0	Time interval after which the GNSS receiver power is cut off after the ignition turn-off, msec
EGTS_GNSS_DATA_RATE	0x0302	INT/ 1, 2,5,10	1	GNSS receiver data output rate, Hz
EGTS_GNSS_MIN_ELEVATION	0x0303	INT/ 5…15	5	Minimum elevation (cutoff) angle of navigation spacecraft, deg
Unit parameters				
EGTS_UNIT_SERIAL_NUMBER	0x0400	STRING	“”	Serial number of the unit
EGTS_UNIT_HW_VERSION	0x0401	STRING	“”	Hardware platform version
EGTS_UNIT_SW_VERSION	0x0402	STRING	“”	Software version
EGTS_UNIT_VENDOR_ID	0x0403	INT	0	Unit vendor identifier
EGTS_UNIT_ID	0x0404	INT	0	Unique unit identifier assigned by the system operator at first activation of the unit
EGTS_UNIT_IMEI	0x0405	STRING	“”	IMEI Number
EGTS_UNIT_RS485_BAUD_RATE	0x0406	INT	19200	RS485 port bit rate
EGTS_UNIT_RS485_STOP_BITS	0x0407	INT	1	Number of stop bits in case of RS485 data transmission
EGTS_UNIT_RS485_PARITY	0x0408	INT/0,1,2	0	Parity test in case of RS485 data transmission 0 – no test is performed 1 – ODD type test

				2 – EVEN type test
EGTS_UNIT_LANGUAGE_ID	0x0410	INT	0	Preferred language for voice communication to ISO 639 0x5F – Russian
EGTS_UNIT_HOME_DISPATCHER_ID	0x0411	INT	0	Identifier of the telematic platform storing information on the unit account data, list of services offered and their statuses.
EGTS_SERVICE_AUTH_METHOD	0x0412	INT	1	Service use method. 1- simple method (implies that all services are accessible for the terminal by default), 0- with confirmation (only permitted services may be used as advised by the telematic platform)
EGTS_SERVER_CHECK_IN_PERIOD	0x0413	INT	30	Time between attempts to establish TCP/IP connection with the server, sec
EGTS_SERVER_CHECK_IN_ATTEMPTS	0x0414	INT	5	Number of attempts to establish TCP/IP connection with the server, after which repeated attempt to establish a top level session (GPRS) will be made
EGTS_SERVER_PACKET_TOUT	0x0415	INT	5	Time during which the terminal awaits confirmation for the dispatched package from the server, sec
EGTS_SERVER_PACKET_RETRANSMIT_ATTEMPTS	0x0416	INT	3	Number of attempts to retransmit a non-confirmed package, after which the terminal performs repeated session initiation at the TCP/IP level.
EGTS_UNIT_MIC_LEVEL	0x0417	INT/0…10	8	Microphone level
EGTS_UNIT_SPK_LEVEL	0x0418	INT/0…10	6	Speaker volume level

The values of the following parameters can be requested, but cannot be modified or deleted using the command Service: EGTS_UNIT_SERIAL_NUMBER, EGTS_UNIT_HW_VERSION, EGTS_UNIT_SW_VERSION, EGTS_UNIT_VENDOR_ID, EGTS_UNIT_IMEI. Values of these parameters are preset by manufacturers of the relevant Terminal modules and units, as well as by developers of SW for the same.

Units installed in a standard system configuration should support the following parameters:

- EGTS_GPRS_APN

- EGTS_SERVER_ADDRESS
- EGTS_SIM_PIN
- EGTS_AUTOMATIC_REGISTRATION
- EGTS_SELFTEST_INTERVAL
- EGTS_POST_TEST_REGISTRATION_TIME
- EGTS_TEST_MODE_END_DISTANCE
- EGTS_GARAGE_MODE_END_DISTANCE
- EGTS_TEST_MODE_WATCHDOG
- EGTS_USE_GPRS_WHITE_LIST
- EGTS_GPRS_WHITE_LIST
- EGTS_TEST_REGISTRATION_TIMEOUT
- EGTS_TEST_REGISTRATION_PERIOD
- EGTS_GNSS_POWER_OFF_TIME
- EGTS_GNSS_DATA_RATE
- EGTS_GNSS_MIN_ELEVATION
- EGTS_UNIT_SERIAL_NUMBER
- EGTS_UNIT_HW_VERSION
- EGTS_UNIT_SW_VERSION
- EGTS_UNIT_VENDOR_ID
- EGTS_UNIT_ID
- EGTS_UNIT_LANGUAGE_ID
- EGTS_UNIT_IMEI
- EGTS_UNIT_HOME_DISPATCHER_ID

6.4 EGTS_FIRMWARE_SERVICE

This Service is intended to transfer to ST the SW configuration and updates for the HW of the modules and units of both ST and peripherals connected to ST.

6.4.1 DESCRIPTION OF SUBRECORDS

To ensure interaction within the framework of the Service, several subrecords are used, their description and code are given in Table 23.

Table 23: List of subrecords for EGTS_FIRMWARE_SERVICE

Code	Name	Description
0	**EGTS_SR_RECORD_RESPONSE**	**Subrecord used to confirm the Service support protocol record from a EGTS_PT_APPDATA type package.**
33	**EGTS_SR_SERVICE_PART_DATA**	**Subrecord intended to transmit to ST data divided into parts and transmitted sequentially. The subrecord is used to transmit large objects which, due to their length, cannot be transmitted to ST in a single package.**
34	**EGTS_SR_SERVICE_FULL_DATA**	**Subrecord intended to transmit to ST data not divided into parts, but transmitted in a single package**

6.4.1.1 EGTS_SR_SERVICE_PART_DATA subrecord

This subrecord can be use by the Service to transmit entities to ST.

The subrecord structure is shown in Table 24.

The EPQ parameter shows the number of parts to be transmitted, the PN parameter indicates the current part number. The ID field gives a unique definition of the entity to which the part transmitted belongs. The EPQ and PN values for the subrecord are to be in the range from 1 to 65535, provided that the PN value is not to exceed the EPQ value. Otherwise, the data from the subrecord will be ignored.

The object identifier ID, fields PN and EPQ, as well as the record source identifier OID from the service routing level header provide a unique description of the part received for processing and its appurtenance to a specific object. This enables, provided the channel capacity is sufficient, simultaneous transmission of entities to update the SW of various HW parts of ST and peripherals.

Table 24: EGTS_SR_SERVICE_PART_DATA subrecord format for EGTS_FIRMWARE_SERVICE

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Type	Data type	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

- ID – unique identifier of the entity transmitted. Incremented when transmission of a new entity is commenced. This parameter gives a unique identification of the part appurtenance to a specific entity.
- PN – consecutive number of the current part of the transmitted entity
- EPQ - expected number of the parts of the transmitted entity
- ODH - header containing parameters characterizing the transmitted entity. This header is transmitted only for the first part of the entity. When transmitting the second and subsequent parts, this field is not transmitted. The OGH header structure is shown in Table 25.
- OD - data of the transmitted entity

Table 25: Format of the transmitted entity header of the EGTS_SR_SERVICE_PART_DATA subrecord for EGTS_FIRMWARE_SERVICE

<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>Bytes</th></tr>
<tr><td colspan="8">OA (Object Attribute)</td><td rowspan="2">M</td><td rowspan="2">BYTE</td><td rowspan="2">1</td></tr>
<tr><td colspan="3">-</td><td colspan="2">OT (Object Type)</td><td colspan="2">MT (Module Type)</td></tr>
<tr><td colspan="8">CMI (Component or Module Identifier)</td><td>M</td><td>BYTE</td><td>1</td></tr>
<tr><td colspan="8">VER (Version)</td><td>M</td><td>USHORT</td><td>2</td></tr>
<tr><td colspan="8">WOS (Whole Object Signature)</td><td>M</td><td>USHORT</td><td>2</td></tr>
<tr><td colspan="8">FN (File Name)</td><td>O</td><td>STRING</td><td>0…64</td></tr>
<tr><td colspan="8">D (Delimiter)</td><td>M</td><td>BYTE</td><td>1</td></tr>
</table>

- OA – transmitted entity appurtenance attribute
- OT – entity type by content. The field can have the following values:

  00 = internal SW data (firmware)
  01 = configuration parameters unit
- MT – type of the module for which the transmitted entity is intended. The field can have the following values:

  00 = peripheral equipment
  01 = ST
- CMI – component number in case of the entity appurtenance directly to ST or identifier of a peripheral module/port connected to ST, depending on the MT parameter value
- VER – transmitted entity version (upper byte - number to the left of the separator – indicates the major version, lower byte – number to the right of the separator – indicates the minor version; for example,

version 2.34 will be presented as 0x0222))

- WOS – entire entity signature (checksum). CRC16-CCITT algorithm is used
- FN – transmitted entity file name (the field is optional and can have a zero length)
- D – delimiter of string parameters (always equals 0)

6.4.1.2 EGTS_SR_SERVICE_FULL_DATA subrecord

The subrecord structure is shown in Table 26.

Table 26: EGTS_SR_SERVICE_FULL_DATA subrecord format for EGTS_FIRMWARE_SERVICE

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Type	Data type	Bytes
ODH (Object Data Header)								M	BINARY	7…71
OD (Object Data)								M	BINARY	1…65400

- ODH – header which contains parameters characterizing the entity transmitted. The structure of the parameter fully corresponds to the structure described in Table 25. For a EGTS_SR_SERVICE_FULL_DATA subrecord, the ODH parameter is obligatory and is present in each such subrecord.
- OD - data directly embedded in the transmitted entity

6.4.1.3 EGTS_SR_RECORD_RESPONSE subrecord

The subrecord structure is the same as described in p.6.2.1.1 and is used to confirm the receipt and processing of the EGTS_SR_SERVICE_PART_DATA and EGTS_SR_SERVICE_FULL_DATA subrecords. Furthermore, for all EGTS_SR_SERVICE_PART_DATA subrecords, except the last one, in case of successful processing, the EGTS_SR_RECORD_RESPONSE is to incorporate the result code which is equal to the EGTS_PC_IN_PROGRESS. For the final EGTS_SR_SERVICE_PART_DATA subrecord and each EGTS_SR_SERVICE_FULL_DATA subrecord, provided that the receipt and processing at ST are successful, the EGTS_SR_RECORD_RESPONSE subrecord is to be transmitted and contain the EGTS_PC_OK code which the Service shall recognize as a successful attempt to dispatch the entire entity.

7. TEMPORAL AND QUANTITATIVE PARAMETERS OF THE SERVICE SUPPORT PROTOCOL WHEN USING PACKAGE DATA TRANSMISSION

The temporal and quantitative parameters of the Service support protocol are shown in Table 27.

Table 27: Temporal and quantitative parameters of the Service support protocol

Name	Data type	Range of values	Default value	Description
EGTS_SL_NOT_AUTH_TO	BYTE	0 … 255	6	Waiting time for the ST message containing the data for the authorization procedure at TP after the terminal establishes a new TCP/IP connection, sec. If no message is received within this time, TP shall break the TCP/IP connection established with ST.