

## Протокол передачи навигационных данных NDTP (Navigation Data Transfer Protocol)

### Структура стека протоколов.

Участниками обмена данными являются:

- мобильный блок (МБ);
- сервер сбора данных (ССД);
- автоматизированное рабочее место диспетчера (АРМ ).

Стек протоколов обмена данными между участниками системы, согласно модели OSI, является семиуровневым и представлен ниже. Все уровни реализованы стандартными средствами, для МБ – встроенным стеком GPRS модема, для ССД и АРМ – средствами ОС.

Протокол передачи данных NDTP (Navigation Data Transfer Protocol), состоящий из двух уровней:

- NPL - Navigation data transfer Protocol (Low level) - протокол нижнего уровня (сеансовый).
- NPH - Navigation data transfer Protocol (High level) - протокол верхнего уровня (представления).

В общем случае сетевой, канальный и физический уровень определяются операционной системой и оборудованием.

Все данные в пакетах NPL и NPH передаются в little-endian\* формате, если не оговорено обратное. В описаниях структуры пакетов длина полей указывается в байтах, либо var - для полей с переменной длиной.

Сеансовый уровень (протокол NPL).

На сеансовом уровне осуществляется шифрование и маршрутизация пакетов.

Формат пакета NPL, протокола нижнего уровня (NPL) имеет следующий формат:

	поле	длина	тип	описание
заголовок пакета NPL	<signature>	2	int16	Содержит сведения о сигнатуре пакета NPL, всегда принимает значение равное 0x7E7E
	<data_size>	2	unsigned int16	Определяет размер данных находящихся в поле <data>. Для незашифрованных и зашифрованных пакетов <data_size> всегда равно размеру NPH данных. Если данные передаются в зашифрованном виде, размер поля <data> равен длине данных выровненной по границе 8 байт (требования алгоритма Blowfish). При этом дополнительные байты в расшифрованном пакете не используются. К примеру, если длина NPH пакета равна 18, то <data_size> = 18, а длина поля <data> равна 24.

\* little-endian формат - формат хранения и передачи двоичных данных, при котором сначала передается младший (наименее значимый) бит (байт).

<flags>	2	int16	<p>Определяет опции пакета:</p> <p><b>0 бит - NPL_FLAG_ENCRYPTION</b> – определяет шифрование поля &lt;data&gt;, принимает значение: 0 – нет, 1 – да. Поле &lt;data&gt; шифруется по алгоритму Blowfish**.</p> <p><b>1 бит - NPL_FLAG_CRC</b> - определяет расчет контрольной суммы пакета (CRC). Принимает значения: 0 – нет, 1- да. Если отправитель пакета рассчитывает CRC он должен выставить данный флаг (1), в таком случае получатель имеет возможность проверить валидность пакета.</p> <p><b>2 бит - NPL_FLAG_DELAY</b> – определяет необходимо ли сохранить команду на сервере, если устройство или АРМ недоступны сейчас.</p> <p>0 - пакет не надо сохранять на сервере в очереди отложенных команд ни при каких обстоятельствах,</p> <p>1 — пакет необходимо сохранить в очереди отложенных команд только тогда, когда адресат сейчас недоступен.</p> <p><b>3 бит - NPL_FLAG_TYPE_ENCRYPTION</b> – определяет тип используемого алгоритма шифрования. Данный бит необходимо смотреть, если установлен бит <b>NPL_FLAG_ENCRYPTION</b>, в противном случае пакет не закодирован!</p> <p>0 - алгоритм blowfish</p> <p>1 - алгоритм шифрования ГОСТ 28147_89</p> <p>Адресатом пакета может быть только конкретное устройство или АРМ.</p>
<crc>	2	unsigned int16	Содержит контрольную сумму поля <data>. CRC всегда рассчитывается и проверяется по незашифрованному пакету, что позволяет дополнительно проверять правильность шифрования пакета. То есть при формировании зашифрованного пакета сначала рассчитывается CRC, затем шифруются данные. При разборе пакета сначала расшифровываются данные, затем проверяется CRC.
<type>	1	byte	Указывает, тип передаваемых данных. Существует три типа данных: NPL_TYPE_ERROR - ошибка протокола NPL NPL_TYPE_NPH - пакет данных NPH NPL_TYPE_DEBUG - отладочная информация
<peer_address>	4	unsigned int32	Определяет адрес участника соединения: NPL_ADDRESS_SERVER - сервер. NPL_ADDRESS_BROAD_DISPATHERS - все диспетчеры в группе NPL_ADDRESS_BROAD_DEVICES - все мобильные устройства в группе

\* <http://www.schneier.com/blowfish.html> - алгоритм Blowfish

				NPL_ADDRESS_BROAD_ALL - все участники клиенты в группе NPL_ADDRESS_RESERVED_MIN... NPL_ADDRESS_RESERVED_MAX - зарезервированные адреса. NPL_ADDRESS_DEVICE_MIN... NPL_ADDRESS_DEVICE_MAX - адреса мобильных устройств. NPL_ADDRESS_DISPATCHER_MIN... NPL_ADDRESS_DISPATCHER_MAX - адреса диспетчеров. NPL_ADDRESS_INVALID - не используется.
	<request_id> >	2	unsigned int16	Идентификатор пакета (ID) рекомендуется делать уникальным хотя бы в рамках одной сессии передачи данных. Например, выбрать некоторое значение ID при установке соединения и для каждого последующего пакета увеличивать его ID на единицу. При достижении 0xFFFFFFFF следующее значение ID будет равно 0x00000000 и т.д.
данные пакета NPL	<data>	var	var	данные

Пакеты протокола NPL однонаправленные, подтверждения не требуют.

#### *Маршрутизация пакетов протокола NPL*

Сервер выполняет роль почтальона, при получении пакета, сервер определяет, кому предназначен пакет, согласно параметру, указанному в поле <peer\_address> пакета NPL. Сервер ретранслирует пакет указанному клиенту, указав в поле <peer\_address> адрес клиента, от которого данный пакет получен. В случае ошибки, к примеру, целевой клиент не доступен или занят, сервер уведомляет клиента, отправившего пакет.

При значении поля <peer\_address> равном нулю (0) (адрес сервера) сервер принимает пакет и сохраняет значения в своей памяти (не реализовано) обработку. Если в поле <peer\_address> принятого пакета указан адрес клиента, которому адресован пакет, сервер осуществляет прием и ретрансляцию указанному клиенту (МБ, АРМ).

Если поле <peer\_address> равно NPL\_ADDRESS\_BROAD\_DISPATCHERS (все диспетчера в группе) – рассылка осуществляется всем АРМам, при <peer\_address> равном NPL\_ADDRESS\_BROAD\_DEVICES (все мобильные устройства в группе) – рассылка всем мобильным блокам.

Для клиента поле <peer\_address> входящего пакета указывает адрес, от кого пришел данный пакет. В исходящем пакете указывается адрес, кому предназначен данный пакет.

участник соединения	направление пакета	Интерпретация поля <peer_address>
Сервер	входящий	адрес получателя
	исходящий	адрес отправителя
Клиент	входящий	адрес отправителя

	исходящий	адрес получателя
--	-----------	------------------

Ниже представлены примеры передачи пакетов. В круглых скобках указан адрес участника соединения, в квадратных - пакеты, > и < - направление передачи.

Передача КЛИЕНТ - СЕРВЕР:

Запрос

CLIENT(100) > [peer\_address = 0 [data]] > SERVER(0)

Ответ

CLIENT(100) < [peer\_address = 0 [data]] < SERVER(0)

Передача КЛИЕНТ - КЛИЕНТ (сервер ретранслирует пакеты заменяя в них поле <peer\_address>):

Запрос

CLIENT(100) > [peer\_address = 200 [data]] > SERVER(0) > [peer\_address = 100 [data]] > CLIENT(200)

Ответ

CLIENT(100) < [peer\_address = 200 [data]] < SERVER(0) < [peer\_address = 100 [data]] < CLIENT(200)

Ошибка передачи КЛИЕНТ - КЛИЕНТ:

Запрос

CLIENT(100) > [peer\_address = 200 [data]] > SERVER(0) ---//no connecton/---  
CLIENT(200)

Ответ

CLIENT(100) < [peer\_address = 0 [error]] < SERVER(0)

Для защищенного соединения сервер выполняет перешифровку пакетов в передаче КЛИЕНТ-КЛИЕНТ.

*Проблемы:*

*"Черные ящики" работают, через одну клиентскую программу, по существующей схеме потребуется дополнительный роутинг пакетов протокола нижнего уровня.*

*Типы пакетов NPL*

Пакеты NPL имеют следующие типы:

NPL\_TYPE\_ERROR - ошибка протокола NPL

NPL\_TYPE\_NPH - пакет данных NPH

Тип пакета: NPL\_TYPE\_ERROR

Коды о ошибке протокола NPL, передаются пакетами NPL\_TYPE\_ERROR, которые при передаче не шифруются. Поле передачи данных содержит код ошибки и имеет следующий формат:

поле	длина	тип	Описание
<error_code>	4	unsigned int32	Содержит коды ошибки: NPL_ERR_OK NPL_ERR_UNDEFINED NPL_ERR_INVALID_PEER_ADDRESS NPL_ERR_PEER_NOT_AVAILABLE NPL_ERR_PEER_PERM_DENIED

Существуют следующие ошибки протокола NPL:

Общие ошибки:

NPL\_ERR\_OK - запрос выполнен успешно

NPL\_ERR\_UNDEFINED - код для ошибок не имеющих описания

Ошибки маршрутизации пакетов:

NPL\_ERR\_INVALID\_PEER\_ADDRESS - недопустимый адрес участника соединения

NPL\_ERR\_PEER\_NOT\_AVAILABLE - участника соединения недоступен

NPL\_ERR\_PEER\_PERM\_DENIED - доступ запрещен

Все типы ошибок перечислены в файле ndtp.h.

Тип пакета: NPL\_TYPE\_NPH

Тип пакета NPL\_TYPE\_NPH - пакет NPH, передается на уровне представления (протокол NPH).

Уровень представления (протокол NPH).

Каждый участник соединения(МБ, АРМ) обладает набором функций (услуг) которые он может предоставить другим участникам соединения, все функции логически разделены на группы - **типы обслуживания**. Таким образом, набор **типов обслуживания**, которые поддерживает определенный участник соединения, определяет интерфейс его взаимодействия с другими участниками соединения.

Для каждого типа обслуживания определены свои типы пакетов и логика работы.

Некоторые типы пакетов могут использоваться в нескольких типах обслуживания (например: пакет NPH\_RESULT – пакет подтверждения, отсылается на запрос, не требующий получения данных). Участник соединения может не поддерживать некоторые пакеты в определенном типе обслуживания.

Обмен данными на уровне представления ведется с помощью пакетов NPH.

Пакет NPH имеет следующий формат:

	поле	длина	тип	описание
заголовок пакета NPH	<service_id>	2	unsigned int16	тип обслуживания
	<type>	2	unsigned int16	тип пакета
	<flags>	2	unsigned int16	Флаги пакета (определяет необходимость подтверждения). Bit 0 NPH_FLAG_REQUEST - определяет, что данный пакет требует подтверждения, принимает значения: 0 – пакет не требует подтверждения; 1 – пакет требует подтверждения. Бит может быть установлен, а подтверждение не будет высылаться так как в первую очередь подтверждение регламентируется протоколом обмена. То есть на запрос протоколом регламентируется ответ и когда передатчик посылает данные, то приемник ждет подтверждения. Вот именно в тех случаях

				когда должно посылаться подтверждение и следует воспринимать данный бит.
	<request_id>	4	unsigned int32	Уникальный идентификатор пакета, используется для подтверждения запроса. Для типа обслуживания NPH_SRV_NAVDATA, является идентификатором сквозной нумерации навигационных данных, инкрементируемым при формировании отметки.*
данные пакета NPH	<data>	var	var	Поле содержит данные, является необязательным. Наличие и структура поля <data> должны однозначно определяться типом обслуживания (<service_id>) и типом пакета (<type>).

\* Для типа обслуживания NPH\_SRV\_NAVDATA навигатор в качестве request\_id использует счетчик сформированных пакетов. Этот счетчик имеет тип unsigned int32 и циклично инкрементируется при формировании навигационной отметки на протяжении всего срока работы.

Ниже описаны основные типы обслуживания протокола NPH, и принадлежность типов пакетов:

Тип обслуживания	Тип пакета
<service_id>	<type>
Основные команды. Должны поддерживаться всеми участниками соединения	
NPH_SRV_GENERIC_CONTROLS	NPH_SGC_CONN_REQUEST <i>Запрос установки соединения</i>
	NPH_SGC_CONN_AUTH_STRING <i>Ответ на установку защищенного соединения (пакет с массивом данных для аутентификации клиента)</i>
	NPH_SGC_SERVICE_REQUEST <i>Запрос типов обслуживания</i>
	NPH_SGC_SERVICES <i>Ответ на запрос типа обслуживания (массив поддерживаемых типов обслуживания)</i>
	NPH_SGC_PEER_DESC_REQUEST <i>Запрос строки описания участника соединения</i>
	NPH_SGC_PEER_DESC <i>Ответ на запрос описания участника соединения</i>
	NPH_GET_PAR_PHOTO <i>Запроса параметров записи фотоснимков</i>
	NPH_SET_PAR_PHOTO <i>Установка параметров записи фотоснимков</i>
	NPH_PAR_PHOTO <i>Параметры записи фотоснимков в навигаторе</i>
	NPH_GET_PAR_PHOTO <i>Запроса фотоснимков из памяти прибора.</i>

Продолжение

Тип обслуживания	Тип пакета
<service_id>	<type>
	NPH_SET_PRDO <i>Программирование дискретных выходов</i>
	NPH_GET_PRDO <i>Запрос дискретных выходов</i>
	NPH_PRDO <i>Ответ</i>
	NPH_SET_MODALARM <i>Программирование режима охраны навигатора</i>
	NPH_GET_MODALARM <i>Запрос режима охраны навигатора</i>
	NPH_MODALARM <i>Данные режима охраны навигатора</i>
	NPH_SET_PRIА <i>Программирование параметров GPRS</i>
	NPH_GET_PRIА <i>Запрос параметров GPRS</i>
	NPH_PRIА <i>Ответ</i>
	NPH_SET_PRNAV <i>Программирование параметров навигации</i>
	NPH_GET_PRNAV <i>Запрос параметров навигации</i>
	NPH_PRNAV <i>Ответ</i>
	NPH_SET_LOADFIRM <i>Команда для считывания прошивки с FTP</i>
	NPH_GET_INFO <i>Запрос информации о навигаторе</i>
	NPH_INFO <i>Ответ</i>
	NPH_GET_BALANCE <i>Запрос баланса на SIM карте</i>
	NPH_BALANCE <i>Ответ баланса на SIM карте</i>
	NPH_SET_CURTIME <i>Установка текущего времени</i>
	NPH_SET_ROUTE_AUTOINFORMER <i>Установка маршрута и рейса для автоинформатора</i>
	NPH_GET_ROUTE_AUTOINFORMER <i>Запрос маршрута и рейса по которому включен автоинформатор</i>
	NPH_ROUTE_AUTOINFORMER <i>Ответ на запрос маршрута и рейса автоинформатора</i>
	NPH_RESET_INT_STATE <i>Команда для сброса внутренних состояний навигатора</i>
	NPH_GET_SIM_IMSI <i>Команда для запроса IMSI кода у SIM карты</i>
	NPH_SIM_IMSI

	<i>IMSI код вставленной SIM карты</i>
	<b>NPH_SET_SEC_SERVER</b>
	<i>Изменение настройки использования дополнительного сервера</i>

Тип обслуживания	Тип пакета
<service_id>	<type>
Передача навигационных данных клиент-сервер	
<b>NPH_SRV_NAVDATA</b>	<b>NPH_SND_HISTORY</b>
	<i>Ретроспективные данные</i>
	<b>NPH_SND_REALTIME</b>
	<i>Данные, поступающие в реальном времени</i>
Передача данных от дополнительных устройств (сообщений)	
<b>NPH_SRV_EXTERNAL_DEVICE</b>	<b>NPH_SED_DEVICE_TITLE_DATA</b>
	<i>Первый пакет сообщения (заголовок)</i>
	<b>NPH_SED_DEVICE_DATA</b>
	<i>Второй и последующие пакеты сообщения</i>
	<b>NPH_SED_DEVICE_RESULT</b>
	<i>Ответа на принятое сообщение</i>
Команды, предназначенные для обработки сервером сбора данных	
<b>NPH_SRV_SERVER_GENERIC</b>	
Запрос списка и состояния клиентов	
<b>NPH_SRV_CLIENT_LIST</b>	<b>NPH_SCL_CLIENT_LIST_REQUEST</b>
	<i>Запрос списка активных МБ</i>
	<b>NPH_SCL_CLIENT_LIST</b>
	<i>Пакет ответа, содержащим список номеров активных ТС</i>
	<b>NPH_SCL_CLIENT_STATUS_REQUEST</b>
	<i>Запрос статуса клиента</i>
Пакеты относящиеся ко всем типам обслуживания	
	<b>NPH_RESULT</b>
	<i>Пакет ответа на запрос относится ко всем типам обслуживания</i>



Тип пакета **NPH\_RESULT** относится ко всем типам обслуживания.  
*Общий пакет подтверждения: NPH\_RESULT*

Пакет NPH\_RESULT является общим для всех типов обслуживания, если запрос не предусматривает получения каких либо данных, в ответ на пакет запроса посылается пакет подтверждения NPH\_RESULT.

Пакет подтверждения NPH\_RESULT имеет следующий формат поля данных:

поле	длина	тип	описание
<error>	4	unsigned int32	0 в случае успешного выполнения запроса или код ошибки

Поле **<error>** пакета NPH\_RESULT может принимать следующие значения:

0 - успешное выполнение запроса;

Общие ошибки:

NPH\_RESULT\_OK - запрос выполнен успешно;

NPH\_RESULT\_UNDEFINED - код для ошибок не имеющих описания;

NPH\_RESULT\_BUSY - участник соединения не может обработать пакет в данный момент;

NPH\_RESULT\_SERVICE\_NOT\_SUPPORTED - тип обслуживания не поддерживается;

NPH\_RESULT\_SERVICE\_NOT\_ALLOWED - тип обслуживания запрещен для данного участника соединения;

NPH\_RESULT\_SERVICE\_NOT\_AVAILABLE - тип обслуживания не доступен в данный момент;

NPH\_RESULT\_PACKET\_NOT\_SUPPORTED - неизвестный тип пакета, либо тип пакет не поддерживается;

NPH\_RESULT\_PACKET\_INVALID\_FORMAT - неверный формат пакета;

NPH\_RESULT\_PACKET\_INVALID\_PARAMETER - неверный параметр пакета;

Ошибки установки соединения:

NPH\_RESULT\_PROTO\_VER\_NOT\_SUPPORTED - версия протокола не поддерживается;

NPH\_RESULT\_CLIENT\_NOT\_REGISTERED - клиент не зарегистрирован на сервере (в БД);

NPH\_RESULT\_CLIENT\_TYPE\_NOT\_SUPPORTED - тип клиента не поддерживается;

NPH\_RESULT\_CLIENT\_AUTH\_FAILED - ошибка аутентификации клиента.

Все типы ошибок перечислены в файле ndtp.h.

*Тип обслуживания NPH\_SRV\_GENERIC\_CONTROLS*

Тип обслуживания **NPH\_SRV\_GENERIC\_CONTROLS** имеет следующие типы пакетов:

**NPH\_SGC\_CONN\_REQUEST** - Запрос установки соединения;

**NPH\_SGC\_CONN\_AUTH\_STRING** - Ответ на установку защищенного соединения (пакет с массивом данных для аутентификации клиента);

**NPH\_SGC\_SERVICE\_REQUEST** - Запрос типов обслуживания;

**NPH\_SGC\_SERVICES** - Ответ на запрос типа обслуживания (массив поддерживаемых типов обслуживания);

**NPH\_SGC\_PEER\_DESC\_REQUEST** - Запрос строки описания участника соединения;

## NPH\_SGC\_PEER\_DESC - Ответ на запрос описания участника соединения.

### Установка соединения с сервером.

Соединение с сервером может быть защищенным или незащищенным. Параметры соединения задаются инициатором соединения в поле <connection\_flags> пакета NPH\_SGC\_CONN\_REQUEST. В первом случае все пакеты передаются в зашифрованном виде за исключением пакетов установки соединения:

NPH\_SGC\_CONN\_REQUEST,  
NPH\_SGC\_CONN\_AUTH\_STRING.

В случае отказа в установке соединения (на любом этапе) сервер посылает клиенту незашифрованный пакет NPH\_RESULT с кодом ошибки.

#### Проблемы:

в такой схеме отсутствует аутентификация сервера (можно реализовать в обратном порядке)

Пакет запроса установки соединения NPH\_SGC\_CONN\_REQUEST имеет следующий формат поля <data>:

поле	длина	тип	описание
<proto_version_high>	2	unsigned int16	версия протокола NDTP (старший номер)
<proto_version_low>	2	unsigned int16	версия протокола NDTP (младший номер)
<connection_flags>	2	unsigned int16	опции соединения, определяет настройки соединения, которые будут использоваться после установки соединения. На данный момент их две: бит0: шифровать пакеты (0 - нет, 1 — да) бит1: рассчитывать CRC пакетов (0 - нет, 1 — да) бит2: подключение симулятора (0 — подключается обычный клиент, 1 подключается симулятор) бит3: тип алгоритма шифрования. 0 - blowfish 1 – ГОСТ бит8: наличие поля IMEI (0 - нет, 1 — да) бит9: наличие поля IMSI (0 - нет, 1 — да) - остальные биты не используются.
<peer_address>	4	unsigned int32	адрес участника соединения, пославшего данный пакет
<max_packet_size>	4	unsigned int32	максимальный размер пакета, который сможет обработать этот участник соединения
<peer_info>	struct	var	необязательное поле, информация об участнике соединения, наличие и структура зависят от типа участника соединения <peer_type>
	15	Unsigned char[15]	<b>IMEI.</b> Идентификатор мобильного устройства. При невозможности определения данного

			параметра, данное поле должно заполняться 0 во всех 15 символах
	16	Unsigned char[16]	<b>IMSI.</b> Идентификатор мобильного абонента. При невозможности определения данного параметра, данное поле должно заполняться 0 во всех 16 символах

Так как сервер не устанавливает соединения, то пакет запроса соединения посылают только клиенты(МБ, АРМ).

Поля **<proto\_version\_high>** и **<proto\_version\_low>** определяют версию протокола, по которой собирается работать клиент. Например, для версии 6.02 (она же 6.2):

**<proto\_version\_high>** = 6 и **<proto\_version\_low>** = 2, а для версии 6.20 **<proto\_version\_high>** = 6 и **<proto\_version\_low>** = 20.

Пакет с массивом данных для аутентификации клиента

**NPH\_SGC\_CONN\_AUTH\_STRING**, имеет следующий формат поля **<data>**:

поле	длина	тип	Описание
<b>&lt;data&gt;</b>	var	char[]	массив данных. Длина массива определяется по полю <b>&lt;data_size&gt;</b> пакета <b>NPL</b>

После установки соединения оба участника должны придерживаться настроек соединения и могут игнорировать пакеты им не соответствующие. К примеру, если установлено защищенное соединение, то любой из участников вправе игнорировать незашифрованные пакеты.

Инициатором соединения является клиент, схема установки соединения описана ниже:

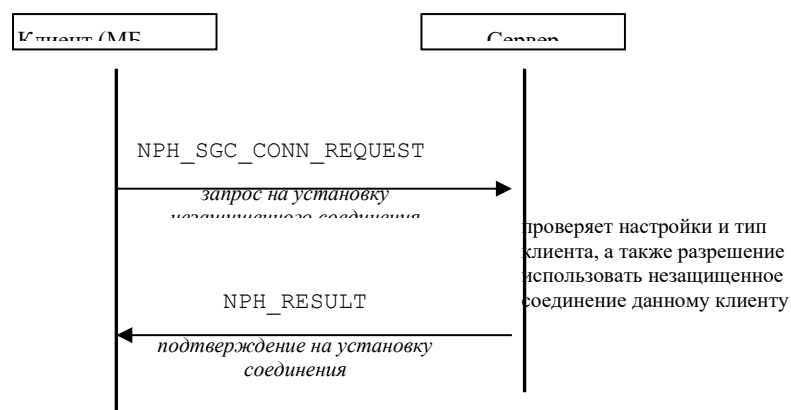
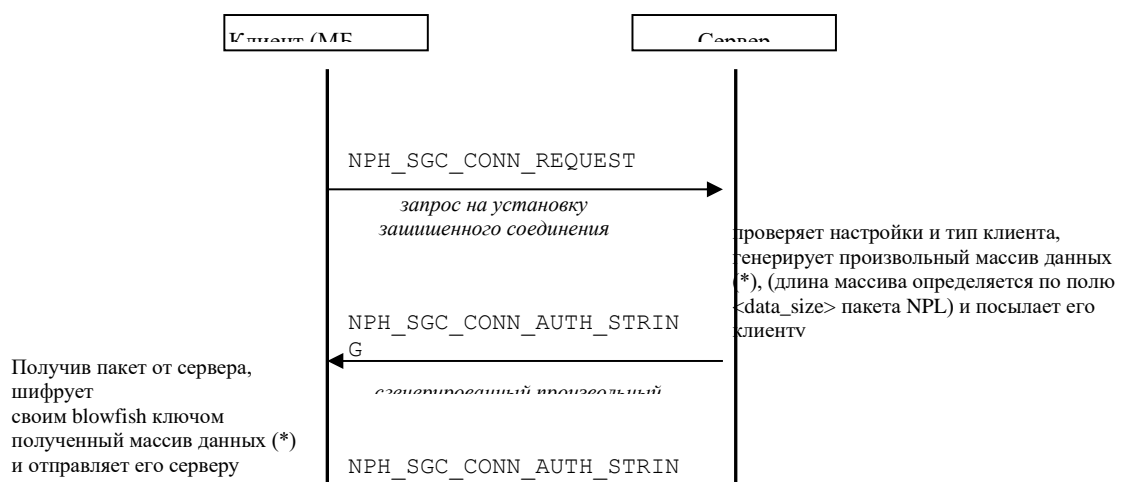


Схема установки не защищенного соединения.



### Запрос типов обслуживания

Пакет запроса типов обслуживания NPH\_SGC\_SERVICE\_REQUEST содержит следующий формат поля <data>:

поле	длина	тип	описание
<data>	2	unsigned int16	Содержит запрашиваемый тип обслуживания (NPH_SRV_CLIENT_LIST, NPH_SRV_DEBUG, NPH_SRV_SERVER_GENERIC, NPH_SRV_EXTERNAL_DEVICE, NPH_SRV_NAVDATA, NPH_SRV_GENERIC_CONTROLS), или данное поле отсутствует

При указании в запросе типа обслуживания (NPH\_SRV\_CLIENT\_LIST, NPH\_SRV\_DEBUG, NPH\_SRV\_SERVER\_GENERIC, NPH\_SRV\_EXTERNAL\_DEVICE, NPH\_SRV\_NAVDATA, NPH\_SRV\_GENERIC\_CONTROLS), пакет ответа передает значения – да, или нет;

В случае отсутствия поля <data> пакет ответа содержит список всех типов обслуживания.

Пакет ответа на запрос типа обслуживания NPH\_SGC\_SERVICES содержит следующий формат поля <data>:

поле	длина	тип	описание
<data>	var	unsigned int16[]	массив поддерживаемых типов обслуживания (NPH_SRV_CLIENT_LIST, NPH_SRV_DEBUG, NPH_SRV_SERVER_GENERIC, NPH_SRV_EXTERNAL_DEVICE, NPH_SRV_NAVDATA, NPH_SRV_GENERIC_CONTROLS)

### Схема запроса на поддержание определенного типа обслуживания

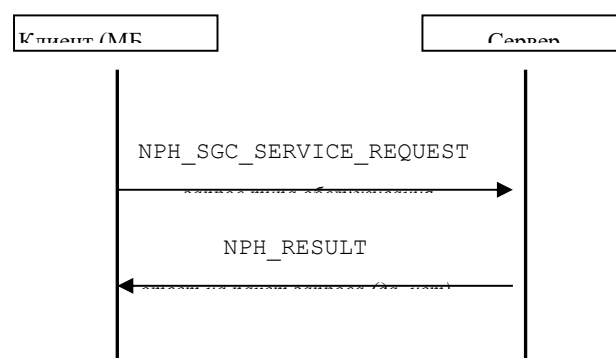
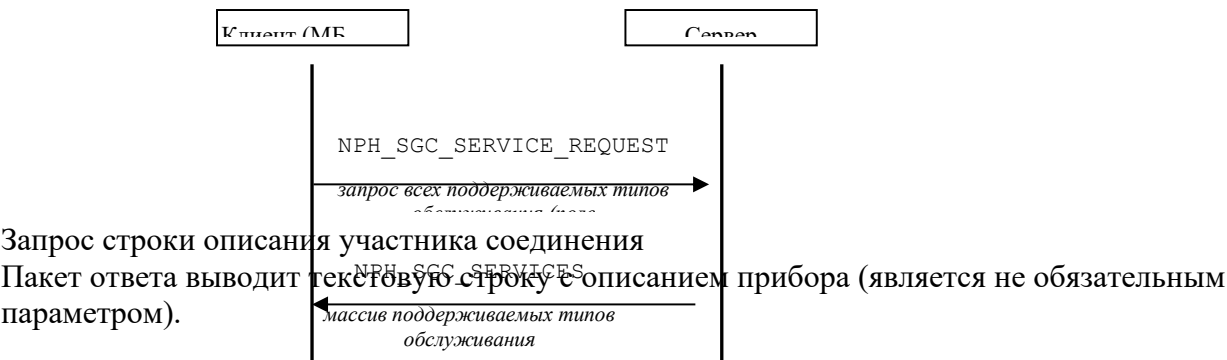


Схема запроса на поддержание типов обслуживания



Запрос строки описания участника соединения  
Пакет ответа выводит текстовую строку с описанием прибора (является не обязательным параметром).

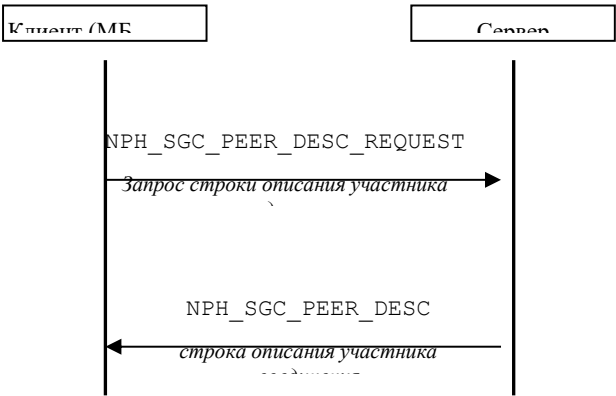
Пакет запроса строки описания участника соединения:  
NPH\_SGC\_PEER\_DESC\_REQUEST. Поле <data> отсутствует:

поле	длин а	тип	описание
<data>	-	-	отсутствует

Пакет ответа строки описания участника соединения NPH\_SGC\_PEER\_DESC содержит следующий формат поля <data>:

поле	длин а	тип	описание
<data>	var	char[]	строка описания участника соединения

Схема запроса строки описания участника соединения



Проблемы:  
в такой схеме отсутствует аутентификация сервера (можно реализовать в обратном порядке)

Команды мобильному блоку (МБ) для настройки параметров передаются пакетами, структура которых описана ниже. В фигурных скобках {} дано значение по умолчанию.

Пакет запроса параметров записи фотоснимков NPH\_GET\_PAR\_PHOTO (предназначен для обработки МБ), имеет следующий вид

поле	длина	тип	описание
< num >	1	byte	номер видеокамеры 0 – первая фотокамера 1 – вторая фотокамера .....

Ответ на прием команды - NPH\_RESULT. А сами данные прибор отправит командой NPH\_PAR\_FOTO.

Пакет установки параметров записи фотоснимков NPH\_SET\_PAR\_PHOTO (предназначен для обработки МБ), имеет следующий вид

поле	длина	тип	описание
< num >	1	byte	номер видеокамеры 0 – первая фотокамера 1 – вторая фотокамера .....
< mode >	1	byte	Режим работы камеры в навигаторе <i>Режим записи на SD карту</i> Bit1..bit0 0 – фотокамеру выключить (нет записи фотоснимков с камеры) 1 – фотокамеру включить работать с нижележащими настройками 3 – не изменять режим сохранения на SD <i>Режим передачи снимка по GPRS</i> Bit3..bit2 0 – выключить передачу данных по GPRS 1 – активировать передачу данных по GPRS. Передача снимков идет с нижележащими параметрами. 3 – не изменять режим передачи фото по GPRS
<res>	1	byte	разрешение: 0 - QVGA 320x240 1 - VGA 640x480
<frame_period >	1	byte	частота кадров в секундах * 10 (1 - 200). То есть 10 – это период формирования фотографии 100 секунд. Значение 0 – ВЫКЛ.
<num_photo>	1	byte	Количество фотоснимков для передачи. Если равно 0xFF, то это значит, пока не придет команда остановить серию фотоснимков, прибор будет отправлять фото с заданной частотой и расширением
<add_ARM>	4	Unsigned int32	Адрес куда необходимо отправлять фотоснимки. Данный адрес либо равен адресу АРМа, либо указывать NPL_ADDRESS_BROAD_DISPATERS.

Ответ на прием команды - NPH\_RESULT.

Примечание. Данной командой можно задавать настройку сохранения снимков на SD карту и передачу их по GPRS параллельно. То есть при настройке команды сохранения снимков на SD карту параметры применяются для сохранения, если задаете команду для

передачи, то параметры применяются для передачи, а если указываете сохранение и передачу снимков, то параметры применяются для обеих функций.

При запросе фотоснимка с указанной видеокамеры, он передается в формате файла с расширением \*.jpg с помощью сервиса EXTERNAL\_DEVICE. В случае запроса последовательности снимков, МБ передает с максимально доступной скоростью последовательность файлов \*.jpg.

Пакет **чтения параметров записи фотоснимков** NPH\_PAR\_PHOTO\_SD (отправляет прибор – ответ на пакет NPH\_GET\_PAR\_PHOTO), имеет следующий вид

Пакет параметров сохранения на SD карту фотоснимков

поле	длина	тип	описание
< num >	1	byte	номер видеокамеры 0 – первая фотокамера 1 – вторая фотокамера .....
< mode >	1	byte	<i>Режим записи на SD карту</i> 0 – фотокамера выключена (нет записи фотоснимков с камеры) 1 – фотокамера включена и работает с нижележащими настройками
<res>	1	byte	разрешение: 0 - QVGA 320x240 1 - VGA 640x480
<frame_period >	1	byte	частота кадров в секундах * 10 (1 - 200). То есть 10 – это период формирования фотографии 100 секунд. Значение 0 – ВЫКЛ.

Пакет **чтения параметров записи фотоснимков** NPH\_PAR\_PHOTO\_GPRS (отправляет прибор – ответ на пакет NPH\_GET\_PAR\_PHOTO), имеет следующий вид

Пакет параметров передачи по GPRS фотоснимков

поле	длина	тип	описание
< num >	1	byte	номер видеокамеры 0 – первая фотокамера 1 – вторая фотокамера .....
< mode >	1	byte	<i>Режим передачи снимка по GPRS</i> Bit3..bit2 0 – передача данных по GPRS выключена 1 – активирована передача данных по GPRS. Количество и адрес, куда передавать снимки, определяются параметрами <num_photo> и <add_ARM>
<res>	1	byte	разрешение: 0 - QVGA 320x240 1 - VGA 640x480
<frame_period >	1	byte	частота кадров в секундах * 10 (1 - 200). То есть 10 – это период формирования фотографии 100 секунд. Значение 0 – ВЫКЛ.
<num_photo>	1	byte	Количество фотоснимков для передачи. Если равно 0xFF, то это значит, пока не придет команда

			остановить серию фотоснимков, прибор будет отправлять фото с заданной частотой и расширением
<add_ARM>	4	Unsigned int32	Адрес куда необходимо отправлять фотоснимки. Данный адрес либо равен адресу АРМа, либо указывать NPL_ADDRESS_BROAD_DISPATCHERS.

**Пакет запроса фотоснимков из памяти прибора NPH\_GET\_PHOTO** (запрос фотоснимка/ фотоснимков из памяти прибора - SD карты), имеет следующий вид  
**Пакет параметров передачи по GPRS фотоснимков**

поле	длина	тип	описание
< num >	1	byte	номер видеокамеры 0 – первая фотокамера 1 – вторая фотокамера .....
< UTC_time>	4	Unsigned int32	UTC время. Прибор ищет ближайшее время к данному запросу и при нахождении снимков отправляет их в АРМ.
<num_photo>	1	byte	Количество фотоснимков для передачи. Если равно 0xFF, то это значит, пока не придет команда остановить серию фотоснимков, прибор будет отправлять фото с заданной частотой и расширением
<add_ARM>	4	Unsigned int32	Адрес куда необходимо отправлять фотоснимки. Данный адрес либо равен адресу АРМа, либо указывать NPL_ADDRESS_BROAD_DISPATCHERS.

**Ответ** на прием команды - NPH\_RESULT.

При запросе указывайте время и прибор будет искать ближайший снимок в памяти прибора. Если снимок будет найден, то он отправится по сервису EXTERNAL\_DEVICE. В случае запроса последовательности снимков, МБ передает с максимально доступной скоростью последовательность файлов \*.jpg.

**Пакет запроса параметров чтения данных из тахографа**

**NPH\_GET\_PAR\_TACHOGRAPH** (предназначен для обработки МБ), имеет следующий вид

поле	длина	тип	описание
< num >	1	byte	номер тахографа 0 – первый тахограф 1 – второй .....

Ответ на прием команды - NPH\_RESULT. А сами данные прибор отправит командой NPH\_PAR\_TACHOGRAPH.

**Пакет установки параметров записи отчетов от тахографа**

**NPH\_SET\_PAR\_TACHOGRAPH** (предназначен для обработки МБ), имеет следующий вид

поле	длина	тип	описание
< num >	1	byte	номер тахографа



			0 – первый тахограф 1 – второй .....
< mode >	1	byte	Режим работы считывания данных с тахографа Bit1..bit0 0 – выключить передачу данных по GPRS 1 – активировать передачу данных по GPRS. Передача снимков идет с нижележащими параметрами.
<frame_delay >	1	byte	Через какой промежуток времени делать отчет от тахографа в часах (1-168). То есть 2 – это время 2 часа. 168 — неделя (0 это минимальное значение, то есть час)
<num_report>	1	byte	Количество отчетов для передачи. Если равно 0xFF, то это значит, пока не придет команда остановить серию отчетов от тахографа, прибор будет отправлять отчеты с заданной частотой
<add_ARM>	4	Unsigned int32	Адрес куда необходимо отправлять отчеты. Данный адрес либо равен адресу АРМа, либо указывать NPL_ADDRESS_BROAD_DISPATHERS.

**Ответ** на прием команды - NPH\_RESULT.

Примечание. Данной командой можно задавать настройку как сохранения отчетов от тахографа, так и передачу их по сети. То есть при настройке создания отчетов, они сохраняются на SD и передаются.

По запросу отчета с указанного тахографа, он передается в формате файла с расширением \*.bin с помощью сервиса EXTERNAL\_DEVICE. В случае запроса последовательности отчетов, МБ передает с максимально доступной скоростью последовательность файлов \*.bin.

**Пакет чтения параметров записи отчетов от тахографа NPH\_PAR\_TACHOGRAPH** (отправляет прибор – ответ на пакет NPH\_GET\_PAR\_TACHOGRAPH), имеет следующий вид

**Пакет параметров сохранения на SD карту фотоснимков**

поле	длина	тип	описание
< num >	1	byte	номер тахографа 0 – первый тахограф 1 – второй .....
< mode >	1	byte	Режим работы считывания данных с тахографа Bit1..bit0 0 – выключить передачу данных по GPRS 1 – активировать передачу данных по GPRS. Передача снимков идет с нижележащими параметрами.
<frame_delay >	1	byte	Через какой промежуток времени делать отчет от тахографа в часах (1-168). То есть 2 – это время 2 часа. 168 — неделя (0 это минимальное значение, то есть час)

**Пакет запроса отчетов от тахографа из памяти прибора NPH\_GET\_TACHOGRAPH** (SD карты), имеет следующий вид

поле	длина	тип	описание
< num >	1	byte	номер тахографа 0 – первый тахограф 1 – второй .....
< UTC_time>	4	Unsigned int32	UTC время. Прибор ищет ближайшее время к данному запросу и при нахождении отчетов отправляет их в АРМ.
<num_report>	1	byte	Количество отчетов для передачи.
<add_ARM>	4	Unsigned int32	Адрес куда необходимо отправлять отчеты. Данный адрес либо равен адресу АРМа, либо указывать NPL_ADDRESS_BROAD_DISPATHERS.

**Ответ** на прием команды - NPH\_RESULT.

При запросе указывайте время и прибор будет искать ближайший отчет в памяти прибора. Если отчет будет найден, то он отправится по сервису EXTERNAL\_DEVICE. В случае запроса последовательности отчетов, МБ передает с максимально доступной скоростью последовательность файлов \*.bin.

Тип данных при передаче отчета по сервису EXTERNAL\_DEVICE  
NPH\_SED\_TYPE\_TACHOGRAPH\_ZIP

**Пакет программирования дискретных выходов NPH\_SET\_PRDO**, имеет следующий формат поля <data>:

поле	длина	тип	описание
<output_id>	1	unsigned byte	номер выхода ( 0-7)
< mode >	1	unsigned byte	состояние {0}– выключить, 1 – включить

**Ответ:** NPH\_RESULT

**Пакет запроса дискретных выходов NPH\_GET\_PRDO**. Поле <data> отсутствует.

поле	длина	тип	описание
<output_id>	1	unsigned byte	номер выхода ( 0-7)

**Ответ:** NPH\_RESULT и потом прибор отправит пакет NPH\_PRDO.

**Пакет ответа дискретных выходов NPH\_PRDO**, имеет следующий формат поля <data>:

поле	длина	тип	описание
<output_id>	1	unsigned byte	номер выхода ( 0-7)
< mode >	1	unsigned byte	состояние {0}– выключено, 1 – включено

Данный пакет отправляет навигатор и он не требует подтверждения.

**Пакет программирования режима охраны NPH\_SET\_MODALARM**, имеет следующий формат поля <data>:

поле	длина	тип	описание
< mode >	1	unsigned byte	состояние {0}– выключить охрану, 1 – включить

**Пакет запроса режима охраны NPH\_GET\_MODALARM**, данный пакет не имеет данных.

**Пакет запроса режима охраны NPH\_MODALARM**, данный пакет не имеет данных.

поле	длина	тип	описание
< mode >	1	unsigned byte	состояние {0} – выключена охранка, 1 – включена

**Пакет программирования параметров GPRS NPH\_SET\_PRIА**, имеет следующий формат поля <data>:

поле	длина	тип	описание
<ip_address>	4	unsigned int32	IP адрес ССД
<port>	2	unsigned int16	порт ССД
<URL>	40	char[] null-terminated	url адрес ССД
<APN>	40	char[] null-terminated	имя точки доступа ( 3-40 символов ) { internet.mts.ru }
<User>	10	char[] null-terminated	имя пользователя ( 1-10 символов ) { mts }
<Password>	10	char[] null-terminated	пароль ( 1-10 символов ) { mts }

Устройство будет использовать URL адрес ССД только в случае ip\_address равно 0.

**Ответ: NPH\_RESULT**

**Пакет запроса параметров GPRS NPH\_GET\_PRIА**. Поле <data> отсутствует.

поле	длина	тип	описание
<data>	-	-	отсутствует

**Пакет ответа параметров GPRS NPH\_PRIА**, содержит поле <data> аналогичное пакету NPH\_SET\_PRIА

поле	длина	тип	описание
<ip_address>	4	unsigned int32	IP адрес ССД
<port>	2	unsigned int16	порт ССД
<URL>	40	char[] null-terminated	url адрес ССД
<APN>	40	char[] null-terminated	имя точки доступа ( 3-40 символов ) { internet.mts.ru }
<User>	10	char[] null-terminated	имя пользователя ( 1-10 символов ) { mts }
<Password>	10	char[] null-terminated	пароль ( 1-10 символов ) { mts }

**Пакет программирования параметров GPRS NPH\_SET\_PRIА\_EXT**, имеет следующий формат поля <data>:

поле	длина	тип	описание
<ip_address>	4	unsigned int32	IP адрес ССД
<port>	2	unsigned int16	порт ССД
<URL>	40	char[] null-terminated	url адрес ССД

<APN>	40	char[] null-terminated	имя точки доступа ( 3-40 символов ) { internet.mts.ru }
<User>	10	char[] null-terminated	имя пользователя ( 1-10 символов ) { mts }
<Password>	10	char[] null-terminated	пароль ( 1-10 символов ) { mts }
<Num_server>	2	unsigned int16	Номер сервера

Устройство будет использовать URL адрес ССД только в случае ip\_address равно 0.

**Ответ:** NPH\_RESULT

**Пакет запроса параметров GPRS NPH\_GET\_PRIAX\_EXT.** Поле <data>.

поле	длина	тип	описание
<Num_server> >	2	unsigned int16	Номер сервера

**Пакет ответа параметров GPRS NPH\_PRIAX\_EXT,** содержит поле <data> аналогичное пакету NPH\_SET\_PRIAX\_EXT

поле	длина	тип	описание
<ip_address>	4	unsigned int32	IP адрес ССД
<port>	2	unsigned int16	порт ССД
<URL>	40	char[] null-terminated	url адрес ССД
<APN>	40	char[] null-terminated	имя точки доступа ( 3-40 символов ) { internet.mts.ru }
<User>	10	char[] null-terminated	имя пользователя ( 1-10 символов ) { mts }
<Password>	10	char[] null-terminated	пароль ( 1-10 символов ) { mts }
<Num_server>	2	unsigned int16	Номер сервера

**Пакет программирования навигационных параметров NPH\_SET\_PRNAV,** имеет следующий формат поля <data>:

поле	длина	тип	описание
<SendDataInterval>	4	unsigned int32	Интервал в секундах для генерации нав. отметки в режиме движения
<SendDataInterval_stop>	4	unsigned int32	Интервал в минутах для генерации нав. отметки в режиме стоянки
<Angle>	4	unsigned int32	Угол поворота, для генерирования нав. отметки
<Dist>	4	unsigned int32	Дистанция, после прохождения которой прибор генерирует нав. отметку

**Ответ:** NPH\_RESULT

**Пакет запроса навигационных параметров NPH\_GET\_PRNAV.** Поле <data> отсутствует.

поле	длина	тип	описание
<data>	-	-	отсутствует

**Ответ:** NPH\_RESULT и потом прибор отправит пакет NPH\_PRNAV.

**Пакет ответа параметров GPRS NPH\_PRNAV**, содержит поле <data> аналогичное пакету NPH\_SET\_NPH\_PRNAV

поле	длина	тип	описание
<SendDataInterval>	4	unsigned int32	Интервал в секундах для генерации нав. отметки в режиме движения
<SendDataInterval_stop>	4	unsigned int32	Интервал в минутах для генерации нав. отметки в режиме стоянки
<Angle>	4	unsigned int32	Угол поворота, для генерирования нав. отметки
<Dist>	4	unsigned int32	Дистанция, после прохождения которой прибор генерирует нав. отметку

**Пакет программирования параметров загрузки прошивки по FTP NPH\_SET\_PRIА**, имеет следующий формат поля <data>:

поле	длина	тип	описание
<ip_address>	4	unsigned int32	IP адрес FTP
<port>	2	unsigned int16	порт FTP
<URL>	70	char[] null-terminated	url адрес FTP
<dir_file>	100	char[] null-terminated	Путь к папке прошивки { Firmware/Nav07 }
<file_name>	30	char[] null-terminated	Имя файла прошивки { blazebase_090914.blz }
<User>	20	char[] null-terminated	имя пользователя ( 1-20 символов ) {mts}
<Password>	20	char[] null-terminated	пароль ( 1-20 символов ) {mts}

Устройство будет использовать URL адрес только в случае ip\_address равно 0.

**Ответ:** Нет. Устройство самостоятельно перезагрузится при приеме данной команды.

**Пакет запроса идентификационной информации от устройства NPH\_GET\_INFO.** Поле <data> отсутствует.

поле	длина	тип	описание
<data>	-	-	отсутствует

**Ответ:** NPH\_RESULT и потом прибор отправит пакет NPH\_INFO

**Пакет идентификационная информация NPH\_INFO.** Поле <data> имеет следующий формат:

поле	длина	тип	описание
<Data_Ver>	15	char[] null-terminated	Дата прошивки
<Ver_navig>	10	char[] null-terminated	Тип навигатора
<lat>	4	unsigned int32	Широта
<lon>	4	unsigned int32	Долгота
<Num_sat>	1	unsigned int8	Количество видимых спутников
<speed>	2	unsigned int16	Скорость движения

**Пакет запроса баланса на SIM карте прибора NPH\_GET\_BALANCE.** Поле <data> отсутствует.

поле	длина	тип	описание
<cmd_req>	20	char[] null-terminated	Команда которая используется для запроса баланса. Если данная строка нулевой длины, то навигатор возьмет данную команду из параметров.

**Ответ:** NPH\_RESULT и потом прибор отправит пакет NPH\_BALANCE

**Пакет строки ответа оператора на запрос баланса NPH\_BALANCE .** Поле <data> имеет следующий формат:

поле	длина	тип	описание
<Balance>	70	char[] null-terminated	Строка, которую вернул оператор на запрос баланса.

**Пакет команда на установку текущего времени в устройстве NPH\_SET\_CURTIME .** Поле <data> имеет следующий формат:

поле	длина	тип	описание
<cur_time>	4	Unsigned int32	Текущее время в формате UTC

**Ответ:** NPH\_RESULT

**Пакет установки маршрута и рейса для автоинформатора навигатора NPH\_SET\_ROUTE\_AUTOINFORMER,** имеет следующий формат поля <data>:

поле	длина	тип	описание
<route>	30	char[] null-terminated	Имя маршрута
<run>	30	char[] null-terminated	Имя рейса

**Ответ:** NPH\_RESULT и потом прибор отправит пакет NPH\_RUN\_AUTOINFORMER

**Пакет запроса маршрута и рейса для автоинформатора навигатора NPH\_GET\_ROUTE\_AUTOINFORMER,** имеет следующий формат поля <data>:

поле	длина	тип	описание
<data>	-	-	отсутствует

Имя маршрута и рейса — это маршрут и рейс в файле автоинформатора. Если они оба равны NULL, то есть имеют 0 длину, то это значит выключение автоинформатора.

**Ответ:** NPH\_RESULT и потом уходит пакет NPH\_ROUTE\_AUTOINFORMER.

**Пакет текущего маршрута и рейса автоинформатора навигатора NPH\_ROUTE\_AUTOINFORMER,** имеет следующий формат поля <data>:

поле	длина	тип	описание
<route>	30	char[] null-terminated	Имя маршрута
<run>	30	char[] null-terminated	Имя рейса

Если маршрут и рейс равны NULL, то есть не содержат строк, то это значит что автоинформатор в навигаторе выключен.

**Пакет сброса внутренних состояний прибора NPH\_RESET\_INT\_STATE**, имеет следующий формат поля <data>:

поле	длина	тип	описание
<state>	4	Unsigned int32	Битовая маска для сброса внутренних состояний прибора. При установке соответствующего бита в единицу прибор сбросит внутреннее состояние. 0 бит — сброс голосового вызова 1 бит — сброс сигнала SOS

**Ответ:** NPH\_RESULT.

**Пакет запроса IMSI кода у вставленной SIM карты в навигатор NPH\_GET\_SIM\_IMSI.**  
Поле <data> отсутствует.

поле	длина	тип	описание
<data>	-	-	отсутствует

**Ответ:** NPH\_RESULT и потом прибор отправит пакет NPH\_SIM\_IMSI.

**Пакет IMSI кода у вставленной SIM карты в навигатор NPH\_SIM\_IMSI**, имеет следующий формат поля <data>:

поле	длина	тип	описание
<IMSI>	20	char[] null-terminated	IMSI код

Если IMSI код пустая строка, то это означает что код сик карты не был прочитан корректно.

**Пакет задания настройки активации дополнительного сервера NPH\_SET\_SEC\_SERVER**, имеет следующий формат поля <data>:

поле	длина	тип	описание
<enable>	1	Unsigned char	Вкл (1) или Выключение (0) работы дополнительного сервера

*Тип обслуживания: NPH\_SRV\_NAVDATA*

Навигационные данные передаются в типе обслуживания NPH\_SRV\_NAVDATA.

Существует два типа пакетов:

NPH\_SND\_REALTIME – передача навигационных данных в реальном времени;

поле	длина	тип	описание
	<b>26</b>	<b>struct</b>	
<time_stamp>	4	unsigned int32	содержит значение реального времени
<longitude>	4	unsigned int32	Содержит долготу в градусах, умноженную на 10 000 000
<latitude>	4	unsigned int32	Содержит широту в градусах, умноженную на 10 000 000
<extra_dop>	1	unsigned int8	bit7 - достоверность навигационных данных ( 1- достоверны, 0 – нет ); bit6 - полушарие долготы ( 1 – E, 0 – W ); bit5 - полушарие широты ( 1 – N, 0 – S ); bit4 - флаг работы от встроенного аккумулятора; bit3 – флаг первоначального включения; bit2 – состояние SOS (1 – SOS, 0 – нет SOS) bit1 – флаг тревожной информации ( один из параметров находится в диапазоне тревоги) bit0 – состояние вызова на голосовую связь (1 – есть запрос на голосовую связь. 0 – нет запроса на голосовую связь)
<bat_voltage>	1	unsigned int8	Напряжение батареи, 1бит = 20мВ
<speed_avg>	2	unsigned int16	Средняя скорость за период в км/ч
<speed_max>	2	unsigned int16	Максимальная скорость за период в км/ч
<course>	2	unsigned int16	Направление движения (0 - 360)
<track>	2	unsigned int16	Пройденный путь, м
<altitude>	2	signed int16	Высота над уровнем моря в метрах ( -18000 - +18000)
<nsat>	1	unsigned int8	Количество видимых спутников
<pdop>	1	unsigned int8	PDOP

NPH\_SND\_HISTORY – передача навигационных данных сохраненной в памяти устройства («ретроспективы»).

Пакеты NPH\_SND\_HISTORY или NPH\_SND\_REALTIME сохраняются сервером в БД. Так же пакет NPH\_SND\_REALTIME ретранслируется диспетчерам той же префиксной группы, что и устройство, отправившее этот пакет. Пакет NPH\_SND\_REALTIME не требует подтверждения от диспетчера, т.к. выставляется бит подтверждения.

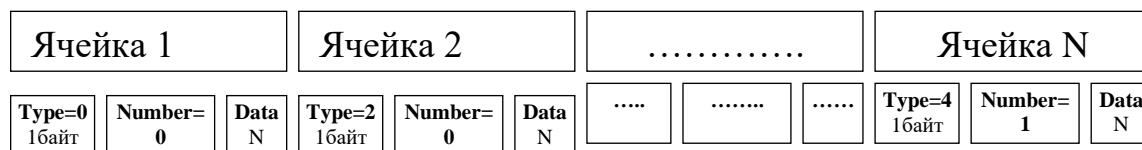
Пакеты передачи навигационных данных NPH\_SND\_HISTORY, NPH\_SND\_REALTIME, имеют следующий формат поля <data>:

поле	длина	тип	описание
< Type >	1	unsigned int8	Тип ячейки (определяет длину и содержимое). Различаются следующие типы: 0 – основные навигационные данные; 1 – дополнительные навигационные данные; 2 – данные от внутренних портов; 3 - данные от устройства Корона; 4 - данные от устройства IRMA
< Number >	1	unsigned int8	Определяет навигационный приемник: <b>N=0</b> – GPS приемник, <b>N=1</b> – GLONASS приемник.
< Data >	var	char[]	Данные от датчика. Структура определяется полем <Type>.

Структура поля <Data> состоит из ячеек, каждая из которых имеет поля: <Type>, <Number> и <Data> переменной длины. Длина каждой ячейки поля <Data> определяется полем <Type>.



Порядок следования ячеек строго по увеличению поля <Type> для того, чтобы сервер смог разобрать известные пакеты, если не знает пакеты с определенного номера <Type>. Также могут идти несколько ячеек с одинаковым полем <Type>, но обязательно разными полями <Number>, которое определяет номер датчика, от которого идет данный пакет. Поле < Number > определяет навигационный приемник: **N=0** – GPS приемник, **N=1** – GLONASS приемник. Время формирования данных от внешних датчиков определяется пакетом навигационных данных.



Ячейка передачи основных навигационных данных (**Type=0, N=0(1)**)

Дополнительные навигационные данные **Type=1, N=0 (1)** (структура на данный момент не реализована).

Данные от внутренних датчиков **Type=2, N=0(1)**

поле	дли на	тип	описание
	<b>26</b>	<b>struct</b>	
<an_in0>	2	unsigned int16	Значение 0 аналогового входа в 16 битном формате
<an_in1>	2	unsigned int16	Значение 1 аналогового входа в 16 битном формате
<an_in2>	2	unsigned int16	Значение 2 аналогового входа в 16 битном формате
<an_in3>	2	unsigned int16	Значение 3 аналогового входа в 16 битном формате
<di_in>	1	unsigned int8	Значение цифровых входов
<di_out>	1	unsigned int8	Состояние дискретных выходов
<di0_counter>	2	unsigned int16	Количество импульсов на дискретном входе 0 с предыдущей нав. отметки
<di1_counter>	2	unsigned int16	Количество импульсов на дискретном входе 1 с предыдущей нав. отметки
<di2_counter>	2	unsigned int16	Количество импульсов на дискретном входе 2 с предыдущей нав. отметки
<di3_counter>	2	unsigned int16	Количество импульсов на дискретном входе 3 с предыдущей нав. отметки
<odometer>	4	unsigned int32	Длина трека с момента первого включения
<CSQ>	1	unsigned int8	Сила GSM сигнала
<GPRS_State >	1	unsigned int8	Состояние GPRS подключения
<Accel_Energ y>	1	unsigned int8	Акселерометр - энергия

<ext_volt>	1	unsigned int8	Напряжение бортовой сети (1 бит - 200мВ)
------------	---	------------------	--

16 битное значение полей <an\_inN>. Если устройство поддерживает 12 битную оцифровку, то оно должно умножить свое значение на 4. То есть сделать максимальное значение равное 0xFFFFC. У такого устройства дискретность становится не 1 (единица), а 4.

Данные от устройства Корона. **Type=3, N=0 (1)**

поле	длина	тип	описание
	<b>14</b>	<b>struct</b>	
<odometer>	4	unsigned int32	Одометр. Циклический счётчик пробега в метрах.
<zone>	2	unsigned int16	Номер зоны для позонной оплаты
<corona_door_in1>	1	unsigned int8	Количество вошедших пассажиров через первую дверь.
<corona_door_in2>	1	unsigned int8	Количество вошедших пассажиров через вторую дверь.
<corona_door_in3>	1	unsigned int8	Количество вошедших пассажиров через третью дверь.
<corona_door_in4>	1	unsigned int8	Количество вошедших пассажиров через четвертую дверь.
<corona_door_out1>	1	unsigned int8	Количество вышедших пассажиров через первую дверь.
<corona_door_out2>	1	unsigned int8	Количество вышедших пассажиров через вторую дверь.
<corona_door_out3>	1	unsigned int8	Количество вышедших пассажиров через третью дверь.
<corona_door_out4>	1	unsigned int8	Количество вышедших пассажиров через четвертую дверь.

Данные от устройства IRMA. **Type=4, N=0 (1)**

поле	длина	тип	описание
	<b>15</b>	<b>struct</b>	
<odometer>	4	unsigned int32	Одометр. Циклический счётчик пробега в метрах.
<zone>	2	unsigned int16	Номер зоны для позонной оплаты
<irma_door_in1>	1	unsigned int8	Количество вошедших пассажиров через первую дверь.
<irma_door_in2>	1	unsigned int8	Количество вошедших пассажиров через вторую дверь.
<irma_door_in3>	1	unsigned int8	Количество вошедших пассажиров через третью дверь.
<irma_door_in4>	1	unsigned int8	Количество вошедших пассажиров через четвертую дверь.
<irma_door_out1>	1	unsigned int8	Количество вышедших пассажиров через первую дверь.
<irma_door_out2>	1	unsigned int8	Количество вышедших пассажиров через вторую дверь.

<irma_door_out3>	1	unsigned int8	Количество вышедших пассажиров через третью дверь.
<irma_door_out4>	1	unsigned int8	Количество вышедших пассажиров через четвертую дверь.
<irma_present_door>	1	unsigned int8	Поле – битовая маска присутствия двери (bit3..bit0) 1 - дверь присутствует 0 – дверь не найдена битовая маска закрытия двери (bit7..bit4). То есть если бит установлен в 1, то дверь закрывалась на остановке. 1 - дверь закрывалась 0 – дверь не закрывалась

Данные от пульта КДМ. **Type=5, N=0 (1)**

поле	длина	тип	описание
	<b>6</b>	<b>struct</b>	
<PGM_enable>	1	unsigned int8	Признак работы оборудования распределителя: дискретный параметр со значениями: 1 - включено, 0 - выключено.
<PGM_width>	1	unsigned int8	Параметр: ширина посыпки/распределения ПГМ (метры). Диапазон значений 0- 15 метров. Точность: единицы метров.
<PGM_density>	2	unsigned int16	Параметр: плотность посыпки или распределения ПГМ (г/м2). Диапазон значений: 0 - 511 г./м2; Точность: единицы граммов.
<plough_state>	1	unsigned int8	Признак: плуг опущен/поднят: дискретный параметр со значениями: 1 - опущен, 0 - поднят.
<brush_state>	1	unsigned int8	Признак щетки включены/выключены (средний гребень опущен/поднят): дискретный параметр со значениями: 1 - включены, 0 - выключены.

Данные от счетчика – интеллектуальный датчик навигатора. **Type=6, N=0 (1...n)**

поле	длина	тип	описание
	<b>9</b>	<b>struct</b>	
Количество импульсов за интервал времени (например в 1 секунду)			
<num_impulse_min>	2	unsigned int16	Минимальное количество импульсов за период времени <time> между двумя навигационными отметками.
<num_impulse_max>	2	unsigned int16	Максимальное количество импульсов за период времени <time> между двумя навигационными отметками..
<time>	1	unsigned int8	Интервал времени за который счетчик считает количество импульсов.
Подсчет количества раз переполнения импульсов определенного значения (например 1000 импульсов )			
<num_overflow>	1	unsigned int8	Количество переполнения счетчика (сколько раз счетчик просчитал импульсов больше <num_impulse>).

<num_impulse>	2	unsigned int16	Количество импульсов при подсчете по кругу.
Информация об импульсах			
<pres_impulse>	1	unsigned int8	Флаг присутствия импульсов.

Данные от цифрового датчика – интеллектуальный датчик навигатора. **Type=7, N=0 (1...n)**

поле	длина	тип	описание
	<b>1</b>	<b>struct</b>	
<value>	1	unsigned int8	Состояние интеллектуального цифрового датчика. Принимает значения: 0 – Норма 1 – Тревога 2 – Обрыв 3 – Замыкание на массу 4 – Замыкание на питание

Данные от топливного датчика УЗИ-М . **Type=8, N=0 (1...n)**

поле	длина	тип	описание
	<b>6</b>	<b>struct</b>	
<det_status>	1	unsigned int8	Состояние датчика топлива.
<level_mm>	2	unsigned int16	Уровень топлива в баке в мм
<level_l>	2	unsigned int16	Уровень топлива в баке в литрах
<temperature>	1	unsigned int8	Температура бака

Данные от регистратора . **Type=9, N=0 (1...n)**

поле	длина	тип	описание
	<b>40</b>	<b>struct</b>	
<ID>	8	unsigned int64	ID регистратора
<Name>	32	unsigned int8	Массив данных считанный из регистратора размером 32 байта. <b>Не воспринимать</b> как строку символов заканчивающейся 0.

Данные от CAN модуля M333. **Type=10, N=0 (1...n)**

поле	длина	тип	описание
	<b>37</b>	<b>struct</b>	
<SecFlagStatus>	4	unsigned int32	Флаг работоспособности устройства и флаги безопасности (Security state flags)
<AllTimeEngine>	4	unsigned int32	Полное время работы двигателя ч * 100. То есть 1ч 15 мин = 125
<AllTrack>	4	unsigned int32	Полный пробег транспортного средства км * 100
<AllFuelConsum>	4	unsigned int32	Полный расход топлива л

<FuelLevel>	2	unsigned int16	bit15 – определяет единицу измерения уровня топлива. Если равен 1, то проценты, если 0 — то литры. bit0 .. bit14 – уровень топлива
<SpeedTurnEngine>	2	unsigned int16	Скорость оборотов двигателя rpm
<TEngine>	2	signed int16	Температура двигателя °C
<Speed>	1	unsigned int8	Скорость тр средства км/ч
<PressureAxis>	2	unsigned int16	Давление на ось 1 кг * 10
<PressureAxis>	2	unsigned int16	Давление на ось 2 кг * 10
<PressureAxis>	2	unsigned int16	Давление на ось 3 кг * 10
<PressureAxis>	2	unsigned int16	Давление на ось 4 кг * 10
<PressureAxis>	2	unsigned int16	Давление на ось 5 кг * 10
<FlagAlarm>	4	unsigned int32	Контроллеры аварии (см. описание ниже)

### Описание флагов.

<SecFlagStatus> - Основные флаги работы автомобиля.

Если значение равно 0xFFFFFFFF — то устройство CAN модуль не найден.

bit 0 .. bit 7

1 - автомобиль закрыт при помощи штатного брелока

2 - автомобиль открыт при помощи штатного брелока

3 - багажник открыт при помощи штатного брелока

4 - модуль выслал сигнал rearming в сигнализацию

7 - модуль перешел в режим экономии энергии „sleep mode”

bit 8 - открыта дверь водителя

bit 9 - открыта дверь пассажира

bit 10 - открыт багажник

bit 11 - открыт капот

bit 12 - затянут рычаг ручного тормоза (информация доступна только при включенном зажигании)

bit 13 - нажат ножной тормоз (информация доступна только при включенном зажигании )

bit 14 - двигатель работает (информация доступна только при включенном зажигании )

bit 16 - зажигание включено

bit 17 – штатная сигнализация поставлена на охрану (находится в режиме тревоги)

bit 18 - автомобиль закрыт при помощи штатного брелока

bit 19 - ключ находится в замке зажигания

bit 20 – включено динамичное зажигание 2

<FlagAlarm> - флаги контроллеров аварий

Устройство само решает, сколько ячеек будет в пакете. Структура пакета ограничивает только то, что ячейки следуют одна за другой и размер ячейки определяется полем

<Type>.

Данные от RFID. Type=12, N=0

поле	длина	тип	описание
------	-------	-----	----------

	<b>5</b>	<b>struct</b>	
<Key>	5	unsigned int8	Считанный RFID ключ

Данные от датчика уровня продукта в отсеке . **Type=13, N=0 (1...n)**

поле	длина	тип	описание
	<b>13</b>	<b>struct</b>	
<density>	4	float	Плотность продукта
<temperature>	4	float	Температура продукта
<level>	4	float	Уровень продукта
<level_unit>	1	unsigned int8	Единица измерения 1 - % 2 — мм 3 м3

Данные от датчика BMS Liaz. **Type=14, N=0 (1...n)**

поле	длина	тип	описание
	<b>15</b>	<b>struct</b>	
<max_temperature >	2	unsigned int16	Максимальная температура
<min_cell_voltage >	2	unsigned int16	Минимальное напряжение ячейки * 0,0015 (Вольт)
<max_cell_voltage >	2	unsigned int16	Максимальное напряжение ячейки * 0,0015 (Вольт)
<voltage>	4	unsigned int32	Общее напряжение в вольтах * 0,0015 (Вольт)
<code_error>	1	unsigned int8	Коды ошибок от устройства (если установлен бит, то) 0 бит — Зафиксировано превышение допустимого напряжения на одной из ячеек 1 бит — зафиксировано напряжение на одной из ячеек ниже допустимого уровня 2 бит — зафиксировано превышение температуры на одной из ячеек 3 бит — зафиксировано превышение максимально допустимого тока заряда или разряда
<current>	4	unsigned int32	Ток батареи в Амперах

Данные от датчика LLS Sensor **Type=15, N=0 (1...n)**

поле	длин а	тип	описание
	<b>50</b>	<b>struct</b>	
<Status>	2	unsigned int16	
<main_float_level>	4	unsigned int32	Уровень основного поплавка
<Temperature_avera ge>	4	unsigned int32	Средняя температура в продукте

<Percent_of_volume>	4	unsigned int32	Процентное заполнение по объему
<total_Volume>	4	unsigned int32	Общий объем
<Weight>	4	unsigned int32	Масса
<Density>	4	unsigned int32	Плотность
<Net_Standard_Volume>	4	unsigned int32	Объем основного продукта
<Level_of_water>	4	unsigned int32	Уровень подтоварной воды
<Pressure>	4	unsigned int32	Давление
<Vapor_temperature_average>	4	unsigned int32	Средняя температура в паровой фазе
<Vapor_Weight>	4	unsigned int32	Масса паровой фазы
<Liquid_phase_Weight>	4	unsigned int32	Масса жидкой фазы

Если поле равно 0xFFFFFFFF - это означает, что данные для этого поля не были получены от датчика или данные устарели ( время ожидания превысило установленный предел для обновления данных ).

Флаги поля Status: бит 6 - сенсор присутствует в списке слотов адаптера, бит 7 - сенсор присутствует на шине LIN и отвечает адаптеру( при установленном бите 6 поля Status ), бит 15 - неисправность сенсора - сенсор перестал отвечать на опрос адаптера, или время ожидания всех параметров превышено ( при установленных битах 6 и 7 поля Status ).

#### Данные от датчика температуры TERMO Type=16, N=0

поле	длина	тип	описание
	<b>8</b>	<b>struct</b>	
<Status>	4	unsigned int32	Если не 0 то нет связи с датчиком
<temp>	4	signed int32	Температура в градусах Цельсия

#### Данные от спиртометра Type=17, N=0

поле	длина	тип	описание
	<b>46</b>	<b>struct</b>	
Status	2	unsigned int16	(бит 0) Device Status Flag - считано текущее состояние устройства ( всегда присутствует в ячейке )  (бит 1) Last Record Flag - считана запись архива (поля присутствуют) AlcoEvent ( тип события / отсек ),

			<p>Alcohol Volume ( слитый / налитый объём ),</p> <p>Starting Time ( время начала операции )</p> <p>и</p> <p>End Time ( время завершения операции )</p> <p><b>(бит 2)</b> Record Counter Flag - считан счетчик записей архива ( если он равен 0, то чтение архива не производится )</p> <p><b>(бит 3)</b> Temperature Flag - температура считана</p> <p><b>(бит 4)</b> Readed Record Flag - считана последняя прочитанная запись архива</p> <p><b>( бит 5)</b> Current Time Flag - считано текущее время устройства</p> <p><b>(бит 6)</b> Current Volume Flag - считано значения текущей операции ( поля содержат данные )</p> <p>Current Volume (текущий слитый / налитый объём )</p> <p>Current Section ( номер отсека текущей операции )</p> <p><b>(бит 14)</b> Status Express Cell - внеочередная ячейка ( могут отсутствовать все данные, кроме Device Status)</p>
AlcoEvent	2	unsigned int16	тип события / номер отсека
Alcohol Volume	2	unsigned int16	слитый / налитый объём
Starting Time	4	unsigned int32	время начала операции
End Time	4	unsigned int32	время завершения операции
Record Counter	2	unsigned int16	счетчик записей архива
Temperature	2	unsigned int16	Температура t1, t2
Readed AlcoEvent	2	unsigned int16	тип события / номер отсека
Readed Alcohol Volume	2	unsigned int16	слитый / налитый объём
Readed Starting Time	4	unsigned int32	время начала операции
Readed End Time	4	unsigned int32	время завершения операции



Current Time	4	unsigned int32	текущее время устройства
Device Status	2	unsigned int16	текущее состояние устройства
Current Volume	2	unsigned int16	текущий слитый / налитый объём
Current Section	2	unsigned int16	номер отсека текущей операции
Reserved16	2	unsigned int16	
Reserved32	4	unsigned int32	

Данные от CAN Sensors (коробки расширителя портов для прибора гранит-навигатор 4.xx)  
**Type=18, N=0(1)**

поле	дли на	тип	описание
	<b>50</b>	<b>struct</b>	
<an_in0>	2	unsigned int16	Значение 0 аналогового входа в 16 битном формате
<an_in1>	2	unsigned int16	Значение 1 аналогового входа в 16 битном формате
<an_in2>	2	unsigned int16	Значение 2 аналогового входа в 16 битном формате
<an_in3>	2	unsigned int16	Значение 3 аналогового входа в 16 битном формате
<an_in4>	2	unsigned int16	Значение 4 аналогового входа в 16 битном формате
<an_in5>	2	unsigned int16	Значение 5 аналогового входа в 16 битном формате
<an_in6>	2	unsigned int16	Значение 6 аналогового входа в 16 битном формате
<an_in7>	2	unsigned int16	Значение 7 аналогового входа в 16 битном формате
<di_in>	1	unsigned int8	Значение цифровых входов
<di_out>	1	unsigned int8	Состояние дискретных выходов
<di0_counter>	4	unsigned int32	Количество импульсов на дискретном входе 0 с предыдущей нав. отметки
<di1_counter>	4	unsigned int32	Количество импульсов на дискретном входе 1 с предыдущей нав. отметки
<di2_counter>	4	unsigned int32	Количество импульсов на дискретном входе 2 с предыдущей нав. отметки
<di3_counter>	4	unsigned int32	Количество импульсов на дискретном входе 3 с предыдущей нав. отметки
<di4_counter>	4	unsigned int32	Количество импульсов на дискретном входе 4 с предыдущей нав. отметки
<di5_counter>	4	unsigned int32	Количество импульсов на дискретном входе 5 с предыдущей нав. отметки

<di6_counter>	4	unsigned int32	Количество импульсов на дискретном входе 6 с предыдущей нав. отметки
<di7_counter>	4	unsigned int32	Количество импульсов на дискретном входе 7 с предыдущей нав. отметки

16 битное значение полей <an\_inN>. Если устройство поддерживает 12 битную оцифровку, то оно должно умножить свое значение на 4. То есть сделать максимальное значение равное 0xFFFC. У такого устройства дискретность становится не 1 (единица), а 4.

Данные о базовых станциях GSM (для прибора гранит-навигатор 4.xx) **Type=19, N=0(1)**

поле	длина	тип	описание
	<b>40</b>	<b>struct</b>	
<MCC>	2	unsigned int16	Код страны
<MNC>	1	unsigned int8	Код оператора
Данные от рабочей вышки GSM			
<LAC>	2	unsigned int16	LAC
<CID>	2	unsigned int16	CID
<RSSI>	1	unsigned int8	RSSI
<time_adv>	2	unsigned int16	timing advance
Данные от других видимых вышек GSM			
<LAC_1>	2	unsigned int16	LAC
<CID_1>	2	unsigned int16	CID
<RSSI_1>	1	unsigned int8	RSSI
<LAC_2>	2	unsigned int16	LAC
<CID_2>	2	unsigned int16	CID
<RSSI_2>	1	unsigned int8	RSSI
<LAC_3>	2	unsigned int16	LAC
<CID_3>	2	unsigned int16	CID
<RSSI_3>	1	unsigned int8	RSSI
<LAC_4>	2	unsigned int16	LAC
<CID_4>	2	unsigned int16	CID

<RSSI_4>	1	unsigned int8	RSSI
<LAC_5>	2	unsigned int16	LAC
<CID_5>	2	unsigned int16	CID
<RSSI_5>	1	unsigned int8	RSSI
<LAC_6>	2	unsigned int16	LAC
<CID_6>	2	unsigned int16	CID
<RSSI_6>	1	unsigned int8	RSSI

Данные дополнительные от CAN модуля M333. Type=20, N=0 (1...n)

поле	длин а	тип	описание
	<b>8</b>	<b>struct</b>	
<flag_high>	4	unsigned int32	Битовая маска флагов  Бит (значении 1 — вкл или активно, 0 в противном случае) 11 (0x0400) Статус включения передней гидравлики
<flag_low>	4	unsigned int32	Битовая маска флагов  Бит (значении 1 — вкл или активно, 0 в противном случае) 1 (0x0001) Состояние щеток 2 (0x0002) Подача воды 3 (0x0004) Состояние пылесоса 4 (0x0008) Выгрузка из бункера 5 (0x0010) Мойка высокого давления 6 (0x0020) Рассивание песка с солью 7 (0x0040) Низкий уровень песка в бункере 10 (0x0200) Низкий уровень жидкости в бункере 11 (0x0400) Статус включения задней гидравлики 12 (0x0800) Статус завода автономного двигателя 13 (0x1000) Джойстик вправо 14 (0x2000) Джойстик влево 15 (0x4000) Джойстик вперед 16 (0x8000) Джойстик назад

Данные от спиртометр2 Type=21, N=0

поле	длина	тип	описание
	<b>180</b>	<b>struct</b>	
Status	2	unsigned int16	<p><b>(бит 0)</b> Device Status Flag = 1: считано текущее состояние устройства ( всегда присутствует в ячейке ), действительны: Current Section ( номер отсека текущей операции )</p> <p><b>(бит 1)</b> Last Record Flag = 1: считана запись архива, действительны:</p> <p>AlcoEvent_Section ( тип события / отсек ), Sensor_Serial_Number ( серийный номер датчика ), Counter_Start_Value ( счетчик в начале операции ), Counter_Stop_Value ( счетчик в конце операции ), Starting Time ( время начала операции ), End_Time ( время завершения операции ), Final_Temperatur ( температура в конце операции ).</p> <p><b>(бит 2)</b> Record Counter Flag = 1: считан счетчик записей архива ( если он равен 0, то чтение архива не производится )</p> <p><b>(бит 3)</b> Temperature Flag = 1: температура считана</p> <p><b>(бит 4)</b> Readed Record Flag = 1: считана последняя прочитанная запись архива</p> <p><b>( бит 5)</b> Current Time Flag = 1: считано текущее время устройства</p> <p><b>(бит 6)</b> Current Volume Flag = 1: считан текущий слитый / налитый объём</p> <p><b>(бит 7)</b> Product Type Flag = 1: считан код вида продукции</p> <p><b>(бит 8)</b> Product Code Flag = 1: считан код продукции</p>

			<p><b>(бит 9)</b> Organization Code Flag = 1: считан код организации</p> <p><b>(бит 12)</b> Installation Date &amp; Time Flag = 1: считана дата и время установки средства измерения</p> <p><b>(бит 13)</b> Error Description Flag = 1: считано описание ошибки</p> <p><b>(бит 14)</b> Status Express Cell = 1: внеочередная ячейка ( могут быть недействительны ( ...Flag = 0 ) любые данные, кроме Device_Status_Section )</p>
RecordCounter	2	unsigned int16	количество записей в архиве
Device_Status_Section	2	unsigned int16	текущее состояние устройства: тип события, номер отсека
Current_Time	4	unsigned int32	текущее время устройства
Current_Volume	2	unsigned int16	текущий слитый / налитый объем
AlcoEvent_Section	2	unsigned int16	<b>запись:</b> тип события / номер отсека
Sensor_Serial_Number	8	unsigned int64	<b>запись:</b> серийный номер датчика
Counter_Start_Value	4	unsigned int32	<b>запись:</b> показания счетчика в начале операции
Counter_Stop_Value	4	unsigned int32	<b>запись:</b> показания счетчика в конце операции
Starting_Time	4	unsigned int32	<b>запись:</b> время начала операции
End_Time	4	unsigned int32	<b>запись:</b> время завершения операции
Final_Temperature	2	unsigned int16	<b>запись:</b> температура жидкости в конце операции
Temperature	2	unsigned int16	Температура t1, t2
Reserved16	2	unsigned int16	Reserved
Readed_AlcoEvent_Section	2	unsigned int16	<b>последняя прочитанная запись:</b> тип события / номер отсека
Readed_Sensor_Serial_Number	8	unsigned int64	<b>последняя прочитанная запись:</b> серийный номер датчика
Readed_Counter_Start_Value	4	unsigned int32	<b>последняя прочитанная запись:</b> показания счетчика в начале операции
Readed_Counter_Stop_Value	4	unsigned int32	<b>последняя прочитанная запись:</b> показания счетчика в конце операции
Readed_Starting_Time	4	unsigned int32	<b>последняя прочитанная запись:</b> время начала операции

Readed_End_Time	4	unsigned int32	<b>последняя прочитанная запись:</b> время завершения операции
Readed_Final_Temperature	2	unsigned int16	<b>последняя прочитанная запись:</b> температура жидкости в конце операции
Product_Type_identifier	4	unsigned int8	код вида продукции
Product_Code	20	unsigned int8	код продукции
Organization_Code	18	unsigned int8	код организации
Installation_Date_Time	16	unsigned int8	дата и время установки средства измерения
Error_Description	50	unsigned int8	описание ошибки

**Статистические данные о качестве связи с сервером Type=22, N=0 (1...n)**

поле	длина	тип	описание
	<b>24</b>	<b>struct</b>	
<id_max>	4	unsigned int32	Максимальное значение идентификатора сквозной нумерации навигационных данных в памяти навигатора.
<id_min>	4	unsigned int32	Минимальное значение идентификатора сквозной нумерации навигационных данных в памяти навигатора.
<tm_oldest>	4	unsigned int32	Время самой ранней отметки в файловом хранилище.
<tm_oldest_unack>	4	unsigned int32	Время самой ранней неподтвержденной отметки в файловом хранилище.
<cnt_unack>	4	unsigned int32	Счетчик неподтвержденных пакетов.
<cnt_unack_losted>	4	unsigned int32	Счетчик неподтвержденных утерянных пакетов.

**Статистические данные от трекера Type=23, N=0 (1...n)**

поле	длина	тип	описание
	<b>16</b>	<b>struct</b>	
<cnt_ack>	4	unsigned int32	Количество подтвержденных отметок с начала работы прибора
<cnt_ack_realtime>	4	unsigned int32	Количество отметок, подтвержденных как realtime (с начала работы прибора)
<cnt_noack>	4	unsigned int32	Количество отметок, которые не получилось отправить (с начала работы прибора)
<cnt_connect>	4	unsigned int32	Количество соединений с сервером (с начала работы прибора)

**Данные статуса отметки Type==24**

Поле	длина	тип	описание
------	-------	-----	----------

	<b>4</b>	<b>struct</b>	
<status_flags>	4	unsigned int32	флаги состояния. Расшифровка кодов состояния смотри ниже.

Номера битов и значения:

- 0 – диагностика ( внутренние тесты не обнаружили ошибок, 1 – есть ошибки оборудования )
- 1 - сброс устройства ( первая запись после старта устройства )
- 2 – состояние зажигания ( 0 – выключено, 1 – включено )
- 3 – состояние движения ( 0 – ТС не движется, 1 – ТС движется )
- 4 – кнопка SOS ( 0 – не нажата, 1 – нажата )
- 5 – кнопка SERVICE ( 0 – не нажата, 1 – нажата )
- 6 – датчик вскрытия корпуса ( 0 – корпус закрыт, 1 – корпус открыт )
- 7-8 – состояние внешней антенны GSM ( 00 – нет данных, 01 – подключена, 10 – отключена, 11 – КЗ на землю )
- 9-10 – состояние внешней антенны GNSS ( 00 – нет данных, 01 – подключена, 10 – отключена, 11 – КЗ на землю )
- 11 – jammer GSM ( 0 – отсутствует, 1 – детектируется сигнал джаммера )
- 12 – jammer GNSS ( 0 – отсутствует, 1 – детектируется сигнал джаммера )
- 13 – работа от резервной батареи ( 0 – прибор работает от внешнего питания, 1 – прибор работает от встроенной батареи )
- 14 – разряд резервной батареи ( 0 – оставшаяся ёмкость резервной батареи более 20%, 1 – менее 20% )
- 15-16 – состояние резервной батареи ( 00 – батарея заряжается/заряжена, 01 – батарея не заряжается  $t > T_{high}$ , 10 – батарея не заряжается  $t < T_{low}$ , 11 – батарея не заряжается, причина неизвестна )
- 17-18 – состояние сети GSM ( 00 – нет сети, 01 – есть сеть, 10 – есть GPRS соединение, 11 – есть IP соединение с сервером )
- 19-20 – SIM карта №1 ( 00 – отсутствует, 01 – присутствует, 10 – активна, 11 активна и в роуминге )
- 21-22 – SIM карта №2 ( 00 – отсутствует, 01 – присутствует, 10 – активна, 11 активна и в роуминге )
- 23 – ошибки ECU ( 0 – нет ошибок, 1 – на диагностической шине ТС есть индикация ошибок ECU )
- 24-25 – режим работы прибора ( 00 – постоянный, 01 – спящий, 10 – ждущий, 11 – дежурный )
- 26-27 – текущий профиль ( 00 – HOME, 01 – ROAM, 10 – ECO, 11 – USER )
- 28 – присутствие радиометки ( 0 – нет зарегистрированных радиометок в радиусе действия, 1 – присутствует хотя бы одна зарегистрированная радиометка в радиусе действия )
- 29-30 – источник навигационных данных ( 00 – данные невалидны, 01 – данные DR, 10 – данные LBS, 11 – данные GNSS )

#### Данные калибровки акселерометра для диагностики превышения ускорений Type=25

Поле	длина	тип	описание
	<b>41</b>	<b>struct</b>	
<is_valid>	1	unsigned int8	флаги валидности элементов матрицы кватернионов (произведена калибровка прибора (1) или нет ( 0 ))
<q_0>	8	Double 64	Элемент матрицы кватернионов. Действителен если прибор скалиброван

<q_1>	8	Double 64	Элемент матрицы кватерионов. Действителен если прибор скалиброван
<q_2>	8	Double 64	Элемент матрицы кватерионов. Действителен если прибор скалиброван
<w_0>	8	Double 64	Элемент матрицы кватерионов. Действителен если прибор скалиброван
<w_3>	8	Double 64	Элемент матрицы кватерионов. Действителен если прибор скалиброван

#### Запакованные данные от датчика **Type=100, N=0 (1...n)**

поле	длина	тип	описание
	<b>N</b>	<b>struct</b>	
<id_sensor>	1	unsigned int8	Код типа упакованных данных датчика. Структуры датчика и соответствующие коды описаны ниже
<flag>	1	unsigned int8	Bit0 - флаг упаковки данных <data> 1 - упакованные данные 0 — данные не упакованы
<len_data>	2	unsigned int16	Длина данных поля <data>.
<data>	<len_data>	int8[]	Данные датчика. Распакованные данные должны быть кратны структуре данных датчика.

#### Данные от датчика **MARWIS <id\_sensor>=1**

поле	длина	тип	описание
	<b>40</b>	<b>struct</b>	
<Road_Surface_Temperature>	4	unsigned int32	
<Dewpoint_Temperature>	4	unsigned int32	
<Relative_Humidity_over_RST>	4	unsigned int32	
<Water_Film_Height>	4	unsigned int32	
<Road_Condition>	4	unsigned int32	
<Ice_Percentage>	4	unsigned int32	
<Friction>	4	unsigned int32	
<RxTime>	4	unsigned int32	
<longitude>	4	unsigned int32	
<latitude>	4	unsigned int32	

#### Данные с CAN шины **<id\_sensor>=2**



Поле	длина	тип	описание
	<b>12</b>	<b>struct</b>	
<address>	4	unsigned int32	
<data>	8	unsigned int8	

#### Данные с OBD. Diagnostic Trouble Codes. <id\_sensor>=3

Поле	длина	тип	описание
< ID _DTC >	2	unsigned int16	Идентификатор ошибки.

#### Данные от датчика стиля вождения <id\_sensor>=4

Поле	длина	тип	описание
		<b>struct</b>	
<data>	<len_data>	protobuf	Данные закодированы protobuf по описанию, расположенному ниже. Длина данных надо брать из поля <len_data>, описанного выше

enum MileageEnum

```
{
  MILEAGE_CAN = 0;
  MILEAGE_GPS = 1;
}
```

enum IgnitionEnum

```
{
  IGNITION_OFF = 0;
  IGNITION_ON = 1;
}
```

enum InsuranceSideFlag\_v2

```
{
  INSURANCE2_SIDE_XF = 0; // F - forward, ускорение вперед.
  INSURANCE2_SIDE_XB = 1; // B - backward, ускорение назад.
  INSURANCE2_SIDE_YF = 2;
  INSURANCE2_SIDE_YB = 3;
  INSURANCE2_SIDE_ZF = 4;
  INSURANCE2_SIDE_ZB = 5;
  INSURANCE2_SIDE_LAST = 6;
}
```

message PositionPoint

```
{
  required uint32 ts = 1; // Время в секундах UTC, начиная с 01.01.1970.
  required int32 lat = 2; // Широта.
  required int32 lon = 3; // Долгота.
  optional uint32 speed = 5; // Скорость в км/ч, макс 256 км/ч.
  optional uint32 heading = 6; // Направление в градусах.
```

```

optional uint32    hdop          = 7; // Фактор геометрической точности в гор. плоскости
(HDOP) * 10.
optional uint32    sat_track_gps = 9; // Кол-во отслеживаемых спутников GPS.
optional uint32    sat_track_gln = 11; // Кол-во отслеживаемых спутников GLONAS.
optional MileageEnum mileage_type = 12; // Тип пробега can или gps.
optional uint32    mileage       = 13; // Пробег.
optional uint32    partial_dst   = 14; // Расстояние до предыдущей точки трека.
optional int32     altitude      = 15; // Высота, метры.
optional IgnitionEnum ignition    = 16; // Статус зажигания.
}

```

```

message InsuranceData_v3

```

```

{
    required int32          low_bound = 1; // Значение порога, равен минимуму
выборки ( по умолчанию 0.2g ).
    required InsuranceSideFlag_v2 side = 2; // Сторона сработки порога.
    required uint32         start_sec  = 3; // Начальное время захвата события в сек
от 01.01.1970.
    required uint32         dur_ms     = 4; // Длительность превышения порога, мсек.
    required int32          mean       = 5; // Среднее значение выборки.
    optional PositionPoint  position   = 6; // Данные о местоположении объекта.
    optional uint32         arr_type   = 7; // Тип выборки.
    repeated uint32         arr_tm     = 8; // Значения выборки. Время в миллисекундах
от начала интервала захвата.
    repeated int32          arr_acc    = 9; // Значения выборки. Ускорения.
}

```

```

message insurance_events

```

```

{
    repeated InsuranceData_v3 insurance_events = 1; // Массив событий.
}

```

Значение ускорения считается + это значения сонаправленно с той по которой произошло превышение порога. То есть для стороны -X значение будет положительным если совпадает с направлением -X. Среднее значение сичтается для всего преиода захвата. Так же в нем ищется и минимальное.

#### Данные от датчика видимых блютуз меток <id\_sensor>=5

Поле	длина	тип	описание
		<b>struct</b>	
<data>	<len_data>	protobuf	Данные запкованы protobuf по описанию, рпсположенному ниже. Длину данных надо брать из поля <len_data>, описанного выше

```

enum label_type_t

```

```

{
    BUTTONS_SOS_AND_SERVICE = 1;
}

```

```

enum label_connection_state_t

```

```

{

```

```

CONNECTION_ESTABLISHED    = 1;
CONNECTION_NOT_ESTABLISHED = 2;
}

```

message bluetooth\_visible\_label\_t

```

{
    required label_type_t          label_type          = 1;
    required label_connection_state_t label_connection_state = 2;
    required uint32                label_addr_high      = 3;
    required uint32                label_addr_low       = 4;
    required uint32                label_rssi           = 5;
    required uint32                label_present_in_white_list = 6;
}

```

message bluetooth\_visible\_labels\_t

```

{
    required uint32                bluetooth_visible_labels_count = 1;
    repeated bluetooth_visible_label_t bluetooth_visible_label     = 2;
}

```

#### Данные о причине формирования отметки <id\_sensor>=20

Поле	длин а	тип	описание
	<b>1</b>	<b>struct</b>	
<record_reason>	1	unsigned int8	Причина формирования отметки. Список кодов представлен ниже

```

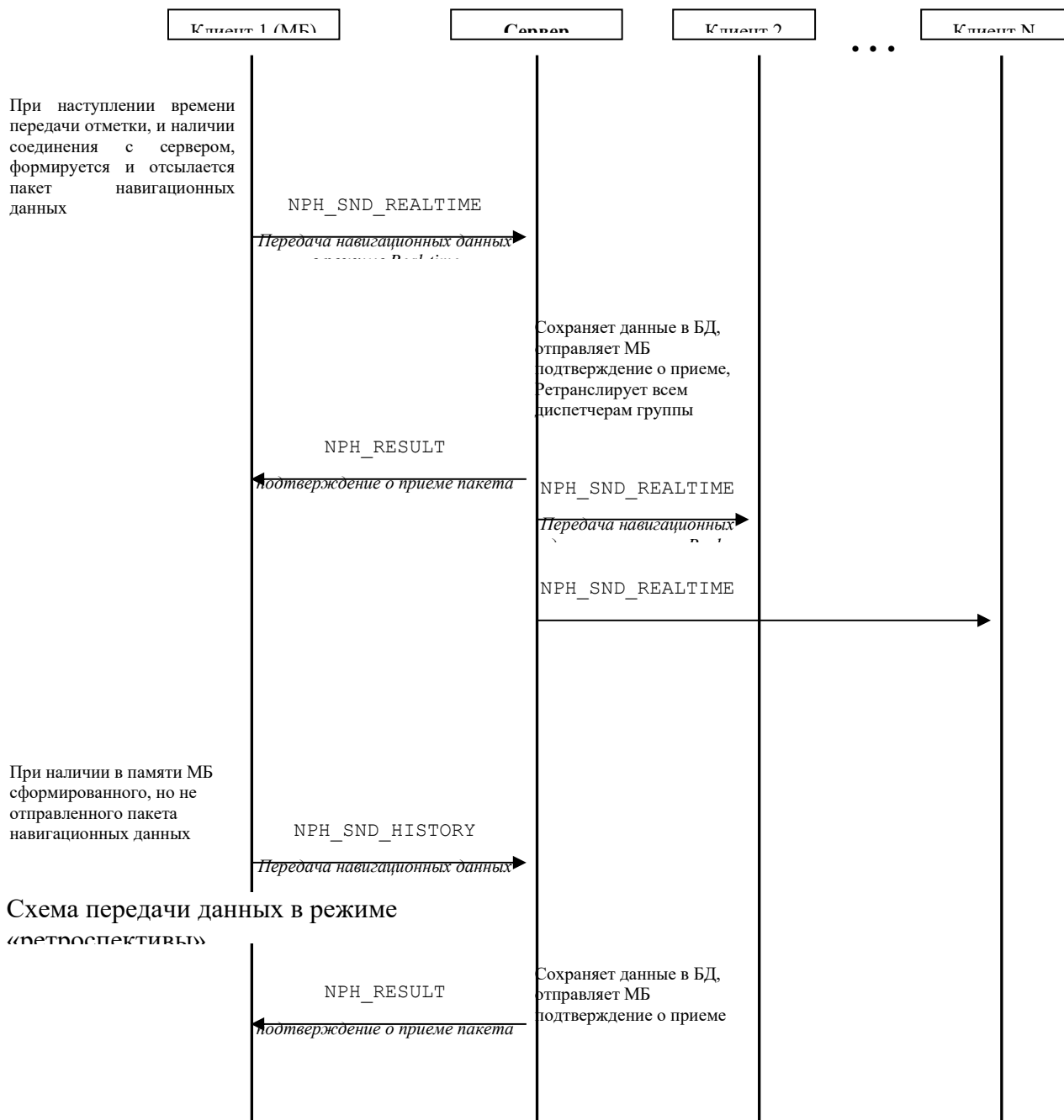
nv_em_storage_record_reason_DEVICE_RESET = 1,
nv_em_storage_record_reason_PROFILE_CHANGE = 2,
nv_em_storage_record_reason_IGNITION_ON = 3,
nv_em_storage_record_reason_IGNITION_OFF = 4,
nv_em_storage_record_reason_TRIP_BEGIN = 5,
nv_em_storage_record_reason_TRIP_END = 6,
nv_em_storage_record_reason_STOP = 7,
nv_em_storage_record_reason_MOVE = 8,
nv_em_storage_record_reason_TOWING_BEGIN = 9,
nv_em_storage_record_reason_TOWING_END = 10,
nv_em_storage_record_reason_TIMER_MOVE = 11,
nv_em_storage_record_reason_TIMER_STOP = 12,
nv_em_storage_record_reason_ANGLE = 13,
nv_em_storage_record_reason_DISTANCE = 14,
nv_em_storage_record_reason_SOS_BUTTON = 15,
nv_em_storage_record_reason_SERVICE_BUTTON = 16,
nv_em_storage_record_reason_TAMPER = 17,
nv_em_storage_record_reason_ANTENNA_SWITCH = 18,
nv_em_storage_record_reason_GSM_LOST = 19,
nv_em_storage_record_reason_GSM_RECONNECT = 20,
nv_em_storage_record_reason_GNSS_LOST = 21,
nv_em_storage_record_reason_GNSS_REAQUISITION = 22,
nv_em_storage_record_reason_JAMMER_GSM = 23,
nv_em_storage_record_reason_JAMMER_GNSS = 24,
nv_em_storage_record_reason_OVER_SPEED = 25,
nv_em_storage_record_reason_OVER_RPM = 26,

```

nv\_em\_storage\_record\_reason\_OVER\_TEMPERATURE = 27,  
 nv\_em\_storage\_record\_reason\_DANGEROUS\_DRIVING = 28,  
 nv\_em\_storage\_record\_reason\_ACCIDENT = 29,  
 nv\_em\_storage\_record\_reason\_OVERTHROW = 30,  
 nv\_em\_storage\_record\_reason\_ON\_BATTERY = 31,  
 nv\_em\_storage\_record\_reason\_BATTERY\_DISCHARGE = 32,  
 nv\_em\_storage\_record\_reason\_RADIO\_TAG\_REG = 33,  
 nv\_em\_storage\_record\_reason\_RADIO\_TAG\_UNREG = 34,  
 nv\_em\_storage\_record\_reason\_MOVE\_WITHOUT\_RADIO\_TAG = 35,  
 nv\_em\_storage\_record\_reason\_ECU\_ERROR\_CODE = 36,  
 nv\_em\_storage\_record\_reason\_EXTERNAL\_REQUEST = 37,  
 nv\_em\_storage\_record\_reason\_DEVICE\_TEST = 38,  
 nv\_em\_storage\_record\_reason\_OTHER\_REASON = 99

Инициатором передачи данных является клиент (МБ), схема взаимодействия клиент – сервер при обработке полученных пакетов навигационных данных описана ниже:

### Схема передачи навигационных данных в режиме Real-time



### Тип обслуживания NPH\_SRV\_EXTERNAL\_DEVICE

Тип обслуживания NPH\_SRV\_EXTERNAL\_DEVICE предназначен для передачи данных от дополнительного устройства МБ (текстовые сообщения, картинки от камеры, звук и др.) на сервер или в АРМ или наоборот. Атомарные данные называются сообщениями.

Данный тип обслуживания имеет несколько типов пакетов данных:

NPH\_SED\_DEVICE\_TITLE\_DATA - Первый пакет (заголовок)

NPH\_SED\_DEVICE\_DATA - Второй и последующие пакеты

NPH\_SED\_DEVICE\_RESULT - Пакет ответа на успешно принятое сообщение

Пакет NPH\_SED\_DEVICE\_TITLE\_DATA имеет следующий формат поля <data>:

поле	длина	тип	описание
<message_id>	2	unsigned int16	Содержит ID сообщения. Для всех пакетов одного сообщения данное поле одно.
<num_packet>	2	unsigned int16	Содержит номер пакета. Для последнего пакета старший бит равен 1. Для всех других пакетов сообщения старший бит равен 0. Если передача осуществляется одним пакетом поле равно 0x8000.
<address_from>	2	unsigned int16	Адрес, от какого устройства данные bit15...bit4 – тип периферийного устройства bit3...bit0 – порядковый номер устройства
<address_to>	2	unsigned int16	Адрес, для какого устройства данные. bit15...bit4 – тип периферийного устройства bit3...bit0 – порядковый номер устройства
<type_data>	4	unsigned int32	Содержит тип структуры данных.
<data_packet>	var	struct	Данные пакета

Каждое сообщение может передаваться не одним пакетом. Для этого используется старший бит в поле <num\_packet>.

Первый пакет имеет номер 0. Таким образом, можно организовать посылку сразу всех пакетов на устройство, так как устройство всегда «знает» сколько пакетов должно прийти.

**Поле < address\_from > и < address\_to > содержит адрес периферии устройства, и определяет «кому» предназначается данное сообщение. Данный адрес разбит на две части:**

**Тип периферийного устройства (bit15...bit4)**

**Порядковый номер данного периферийного устройства (если присутствует более одного периферийного устройства одного типа) (bit3...bit0)**

**Тип периферийного устройства может принимать следующие значения:**

**NPH\_SED\_ADDR\_DISPLAY\_DEVICE**

**NPH\_SED\_ADDR\_VOICE\_DRIVER\_DEVICE**

**NPH\_SED\_ADDR\_VOICE\_DRIVER\_PASSANGERS**

**NPH\_SED\_ADDR\_INTERNAL\_DISPLAY**

**NPH\_SED\_ADDR\_EXTERNAL\_DISPLAY**

**NPH\_SED\_ADDR\_IRMA\_DEVICE**

**NPH\_SED\_ADDR BPKRD\_DEVICE**

**NPH\_SED\_ADDR\_TEMPERATURE\_ON\_BOARD**

Поле **<type\_data>** определяет тип передаваемых данных. Для каждого периферийного устройства можно посылать только те данные, которые он обрабатывает. Если сообщение не может быть обработано периферийным модулем, то на экране прибора отобразится ошибка – принято сообщение с неизвестным форматом.

Следующие значения для поля **<type\_data>**:

NPH\_SED\_TYPE\_MSG\_SCRIPT\_DISPLAY – скрипт сообщения для дисплея прибора

NPH\_SED\_TYPE\_JPG\_FOTO – картинка в формате jpg

NPH\_SED\_TYPE\_VECTOR\_MAP – векторная карта, в собственном формате

NPH\_SED\_TYPE\_VOICE\_LPC10 - голосовое сообщение, сжатое по алгоритму LPC10

NPH\_SED\_TYPE\_VOICE\_GSM630 – голосовое сообщение, сжатое по алгоритму GSM630

NPH\_SED\_TYPE\_VOICE\_SPEEX – голосовое сообщение, сжатое по алгоритму Speex

Пакет NPH\_SED\_DEVICE\_DATA имеет следующий формат поля <data>:

поле	длина	Тип	описание
<message_id>	2	unsigned int16	Содержит ID сообщения. Для всех пакетов одного сообщения данное поле одно.
<num_packet>	2	unsigned int16	Содержит номер пакета. Для последнего пакета старший бит равен 1. Для всех других пакетов сообщения старший бит равен 0. Если передача осуществляется одним пакетом поле равно 0x8000.
<data_packet>	var	Struct	Данные пакета

Пакет ответа, отсылаемый принимающей стороной. В случае успешной передачи NPH\_SED\_DEVICE\_RESULT имеет следующий формат поля <data>:

поле	длина	тип	описание
<num_packet>	2	unsigned int16	Содержит номер пакета. Для последнего пакета старший бит равен 1. Для всех других пакетов сообщения старший бит равен 0.
<error_packet>	4	unsigned int32	Содержит код ошибки (см. п. «Общий пакет подтверждения: NPH_RESULT»). При успешном приеме пакета = 0.
<message_id>	2	unsigned int16	<b>Содержит ID сообщения. Для всех пакетов одного сообщения данное поле одно.</b>

Схема передачи сообщений по схеме КЛИЕНТ – СЕРВЕР аналогична схеме КЛИЕНТ – КЛИЕНТ, только сервер принимает сообщение, и отправляет подтверждение о приеме.

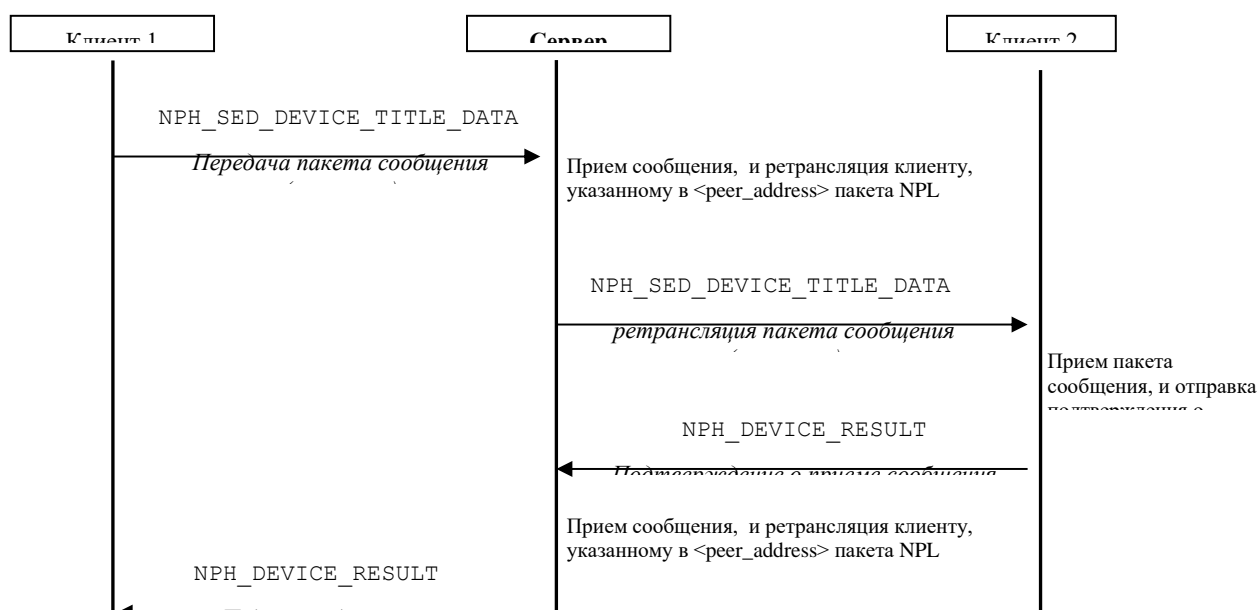


Схема передачи сообщений по схеме КЛИЕНТ - КЛИЕНТ

*Тип обслуживания NPH\_SRV\_DEBUG*

Тип обслуживания NPH\_SRV\_DEBUG, предназначен для передачи отладочной информации протокола NPH, существует следующий тип пакетов:

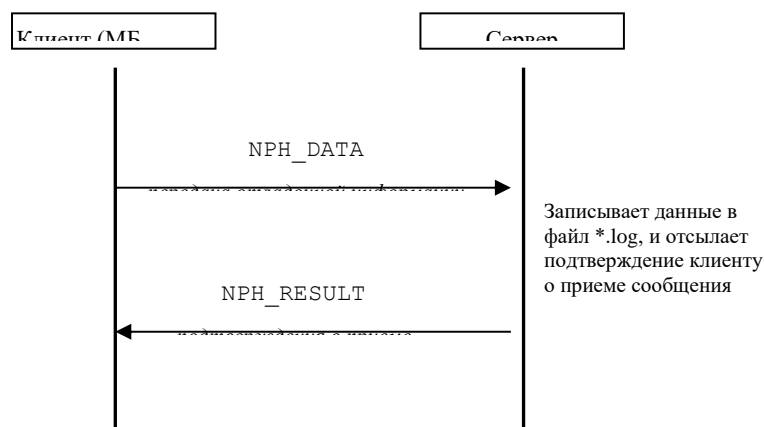
NPH\_EXT\_DEVICE\_DATA – отладочные данные.

Пакет передачи данных от дополнительного устройства МБ на сервер

NPH\_EXT\_DEVICE\_DATA имеет следующий формат поля <data>:

поле	длина	тип	описание
<time_stamp>	4	unsigned int32	содержит значение реального времени
<debug_data>	var	struct	Отладочные данные, которые ССД пишет в *.log файл с именем, соответствующем номеру устройства (если это разрешено в настройках ССД). Например, 41859log.txt

Схема передачи отладочных данных:



*Тип обслуживания NPH\_SRV\_CLIENT\_LIST*

Данный тип обслуживания позволяет клиенту (АРМ) осуществлять запросы списка и состояния активности МБ (зарегистрирован МБ, или нет на сервере в момент запроса), имеет следующие типы пакетов:

NPH\_SCL\_CLIENT\_LIST\_REQUEST – пакет запроса списка активных МБ;

NPH\_SCL\_CLIENT\_LIST – пакет списка активных клиентов;

NPH\_SCL\_CLIENT\_STATUS\_REQUEST – пакет запроса статуса клиента (активен, не активен).

Запрос списка активных МБ (ТС).

В пакете запроса списка активных МБ NPH\_SCL\_CLIENT\_LIST\_REQUEST поле <data> отсутствует:

поле	длина	тип	описание
<data>	-	-	отсутствует

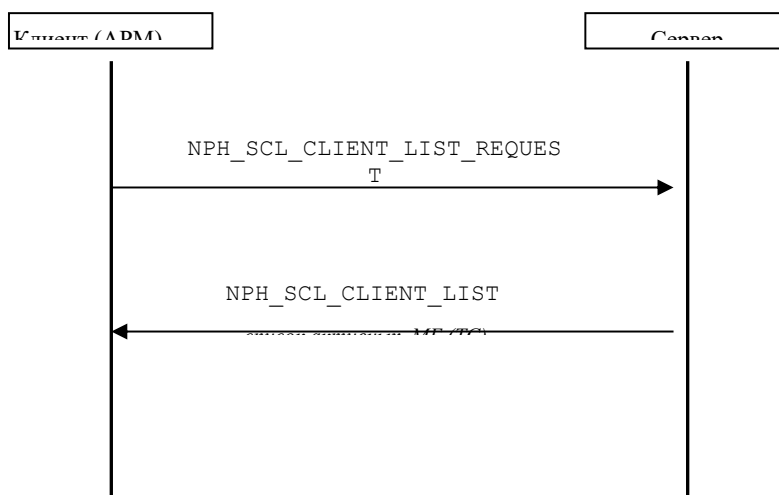
Пакет ответа NPH\_SCL\_CLIENT\_LIST имеет следующий формат поля <data>:

поле	длина	тип	описание
<data>	var	unsigned int32[]	список номеров активных клиентов

Количество номеров определяется по размеру поля <data>, указанному в заголовке NPL.

*проблема: нельзя передать большое число клиентов одним пакетом*

Схема запроса списка активных МБ (ТС).



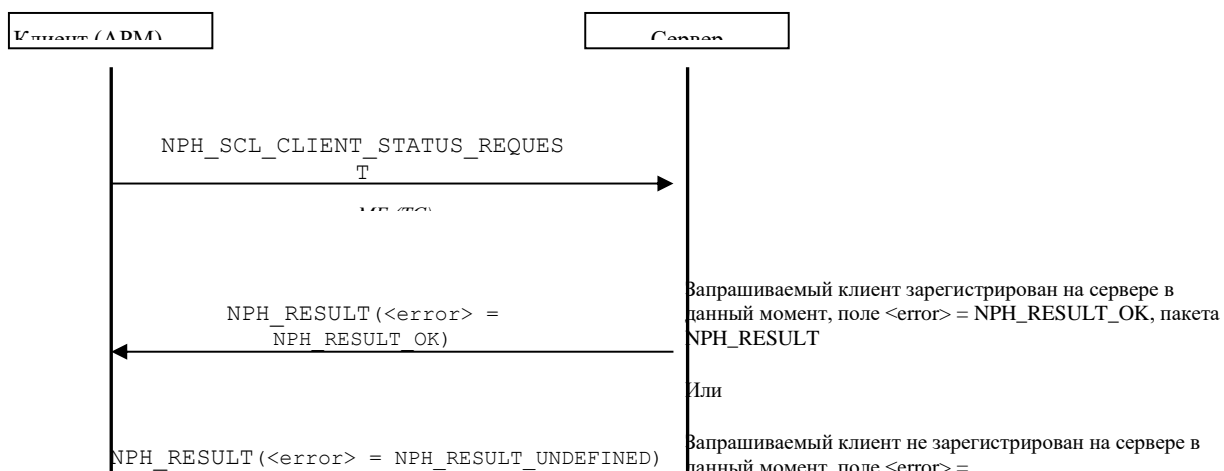
Запрос статуса клиента

Пакет запроса статуса клиента NPH\_SCL\_CLIENT\_STATUS\_REQUEST имеет следующий формат поля <data>:

поле	длина	тип	описание
<data>	4	unsigned int32	номер (NPL адрес) клиента

На запрос сервер должен прислать пакет NPH\_RESULT с кодом NPH\_RESULT\_OK если запрошенный клиент зарегистрирован на сервере в данный момент, в противном случае NPH\_RESULT\_UNDEFINED (если в данный момент клиент не зарегистрирован на сервере) или код ошибки.

Схема запроса состояния активности МБ (ТС).





Коды ошибок протокола NDTP.

Конкретные коды ошибок приведены в файле ndtp.h.

Коды ошибок протокола NPL.

Общие ошибки:

NPL\_ERR\_OK - запрос выполнен успешно

NPL\_ERR\_UNDEFINED - код для ошибок не имеющих описания

Ошибки маршрутизации пакетов:

NPL\_ERR\_INVALID\_PEER\_ADDRESS - недопустимый адрес участника соединения

NPL\_ERR\_PEER\_NOT\_AVAILABLE - участника соединения недоступен

NPL\_ERR\_PEER\_PERM\_DENIED - доступ запрещен

Коды ошибок протокола NPH.

Конкретные коды ошибок приведены в файле ndtp.h.

Общие ошибки:

NPH\_RESULT\_OK - запрос выполнен успешно

NPH\_RESULT\_UNDEFINED - код для ошибок не имеющих описания

NPH\_RESULT\_BUSY - участник соединения не может обработать пакет в данный момент.

NPH\_RESULT\_SERVICE\_NOT\_SUPPORTED - тип обслуживания не поддерживается

NPH\_RESULT\_SERVICE\_NOT\_ALLOWED - тип обслуживания запрещен для данного участника соединения

NPH\_RESULT\_SERVICE\_NOT\_AVAILABLE - тип обслуживания не доступен в данный момент

NPH\_RESULT\_PACKET\_NOT\_SUPPORTED - неизвестный тип пакета, либо тип пакет не поддерживается

NPH\_RESULT\_PACKET\_INVALID\_FORMAT - неверный формат пакета

NPH\_RESULT\_PACKET\_INVALID\_PARAMETER - неверный параметр пакета

Ошибки установки соединения:

NPH\_RESULT\_PROTO\_VER\_NOT\_SUPPORTED - версия протокола не поддерживается

NPH\_RESULT\_CLIENT\_NOT\_REGISTERED - клиент не зарегистрирован на сервере (в БД)

NPH\_RESULT\_CLIENT\_TYPE\_NOT\_SUPPORTED - тип клиента не поддерживается

NPH\_RESULT\_CLIENT\_AUTH\_FAILED - ошибка аутентификации клиента

Формат SMS для мобильных блоков.

Допускается передача нескольких СМС команд в одной СМСке. Единственное ограничение — длина такой СМСки не должна превышать 140 символов (физически одна СМС), иначе прибор правильно не разберет ее. Начинать вторую СМС команду можно сразу после окончания первой не вставляя никаких дополнительных символов.

*Пример* - BB+PRNAV=10,2,0,BB+GETINF

Команда	Описание
BB+PRSMS= <AddServer>, <port>, <APN>, <User>,	Программирование параметров GPRS  < AddServer > - IP адрес компьютера или URL адрес диспетчерского пункта (205.040.101.003 или www.graybox.ru) <port> - порт компьютера диспетчерского пункта ( 5555 )

<p>&lt;Password&gt;, &lt;Num_server&gt;</p>	<p>&lt;APN&gt; - имя точки доступа ( 3-30 символов ) { internet.mts.ru}          &lt;User&gt; - имя пользователя ( 1-8 символов ) {mts}          &lt;Password&gt; - пароль ( 1-8 символов ) {mts}          &lt;Num_server&gt; - номер сервера для установки параметров (значение 1 или 2). Если параметр опущен, то данный параметр принимается за значение 1</p> <p>При отправке SMS, используя сервисы сотовых операторов, желательно после кода SMS ставить Enter, так как таким образом программа отделит код сообщения от рекламы.          Прибор после выполнения команды отправит на телефон ответ.</p>
<p>BB+PRNAV= &lt;MoveTime&gt;, &lt;ParkTime&gt;, &lt;Distance&gt;, &lt;Angle&gt;</p>	<p>Программирование параметров навигационного модуля прибора</p> <p>&lt;MoveTime&gt; - период получения прибором навигационных отметок при движении (в секундах)          &lt;ParkTime&gt;- период получения прибором навигационных отметок при стоянке (в минутах)          &lt;Distance&gt;- расстояние, которое проходит прибор для получения навигационной отметки (в метрах)          &lt;Angle&gt;- угол, на которых повернет прибор для получения навигационной отметки (в градусах)</p> <p>Прибор после выполнения команды отправит на телефон ответ.</p>
<p>BB+PRLFIRM= &lt;AddServer&gt;, &lt;port&gt;, &lt;DirName&gt;, &lt;FileName&gt;, &lt;User&gt;, &lt;Password&gt;</p>	<p>Программирование параметров загрузки прошивки по FTP.</p> <p>&lt; AddServer &gt; - IP адрес сервера или URL адрес сервера (205.040.101.003 или ftp.glorient.ru)          &lt;port&gt; - порт сервера (обычно 21 )          &lt;DirName&gt; - директория на ftp ( 70 символов ) { Navigator07/firmware}          &lt;FileName&gt; - имя файла для скачивания ( 30 символов ) { blazebase_091003.blz}          &lt;User&gt; - имя пользователя для подключения к ftp сервера ( 20 символов )          {LoadFirmware}          &lt;Password&gt; - пароль для подключения к ftp( 20 символов )          {load220903}</p> <p>Алгоритм работы прибора при получении смс.          Получив смс прибор перезагружается и начинает коннектиться к серверу и скачивать файл.          Скачав файл, прибор перепрограммируется и загрузится..</p> <p>После прошивки прибор соединится и отправит ответное смс о результате. Если прибор не сможет скачать файл, то также сообщит об этом.</p>
<p>BB+GETINF</p>	<p>Запрос информационной строки в приборе.</p> <p>Навигатор примет запрос и ответит SMS формата</p>

	<p>#=номер прибора, Ver=дата, Lat=широта, Lon=долгота, S=количество видимых приемником спутников, V=скорость, GPRS=состояние соединения к серверу</p>
BB+GETBAL=<StrRequest>	<p>Запрос баланса на SIM карте навигатора. &lt;StrRequest&gt; - это строка запроса параметров баланса на SIM карте навигатора. Данная строка может быть опущена, в этом случае навигатор возьмет строку, которая записана в параметре для запроса баланса.</p> <p>После получения ответа навигатор отправит ответную SMS на телефон, по которому запрашивали баланс.</p>
BB+RESET	<p>Команда для сброса устройства. После приема данной команды устройство сразу сбрасывается и не посылается ответная СМС.</p>
BB+TELADD=<TelNum>	<p>Команда для добавления телефона в телефонную книгу навигатора. Телефонная книга позволяет фильтровать sms только с тех номеров, которые есть в телефонной книге. &lt;TelNum&gt; - номер телефона, который необходимо добавить в книгу. ПРИМЕЧАНИЕ номер телефона должен быть всегда начинаться с +7.</p> <p>После приема данной sms прибор попытается добавить телефон и отправит ответ с результатом выполнения данного действия.</p>
BB+TELDEL=<TelNum>	<p>Команда для удаления телефона из телефонной книжки. &lt;TelNum&gt; - номер телефона, который необходимо добавить в книгу. ПРИМЕЧАНИЕ номер телефона должен быть всегда начинаться с +7.</p> <p>После приема данной команды навигатор попытается найти телефон в книге и удалить и отправит ответ с результатом выполнения данной команды.</p>
BB+GPRSROMING=<enable>,<AddServer_roming>,<port_roming>,<APN_roming>,<User_roming>,<Password_roming>,<Num_server>	<p>Команда для управления соединения с сервером в роуминге. Данные настройки применяются только когда устройство находится в роуминге.</p> <p>&lt;enable&gt; - активация соединения с сервером в роуминге. Если установить 1, то прибор начнет коннектиться к серверу в роуминге, 0 — прибор не будет предпринимать попыток соединиться с сервером в роуминге. Если параметр опущен, то навигатор воспримет &lt;enable&gt; = 0.</p> <p>&lt;AddServer_roming&gt; - ip адрес сервера, к которому прибор должен коннектиться в роуминге, если параметр опущен, то параметр не изменится</p> <p>&lt;port_roming&gt; - порт сервера в роуминге. Если параметр опущен, то значение не изменится</p> <p>&lt;APN_roming&gt; - APN точки доступа, когда прибор находится в роуминге. Если параметр опущен, то значение не изменится в приборе.</p> <p>&lt;User_roming&gt; - user параметр точки доступа в роуминге. Если параметр опущен, то значение не изменится.</p> <p>&lt;Password_roming&gt; - psw параметр точки доступа в роуминге. Если параметр опущен, то значение не изменится.</p>

	<p>&lt;Num_server&gt; - номер сервера для установки параметров (значение 1 или 2). Если параметр опущен, то данный параметр принимается за значение 1</p>
BB+GETNAVINFO	<p>Команда для запроса минимальной навигации от модуля</p> <p>Навигатор примет запрос и ответит SMS формата  # = номер прибора, D = дата, Lat = широта, Lon = долгота,  NS = количество функций, V = валидность, S = скорость, C = курс  примечание  формат поля даты — год месяц число час минута секунда (2011 10 20 16 32 30)</p>
BB+SECSERVER= <enable>	<p>Команда по управлению включению или выключению дополнительного сервера.</p> <p>&lt;enable&gt; - включение или выключение дополнительного сервера. Если установить в 1 — то это включит передачу навигационных данных на второй сервер. Если параметр опущен, то параметр принимает значение 0.</p>