

МОДУЛИ ЦИФРОВОЙ ОБРАБОТКИ СИГНАЛОВ, ПРЕДНАЗНАЧЕННЫЕ ДЛЯ
ПРИЕМА, ДЕМОДУЛЯЦИИ И ДЕКОДИРОВАНИЯ ЦИФРОВЫХ СИГНАЛОВ ФМ,
АФМ И КАМ.

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

Руководство программиста

Версия 02

Листов 56

2018-2020

СОДЕРЖАНИЕ

	Лист
1. УДАЛЕННОЕ УПРАВЛЕНИЕ	3
1.1. Введение	3
1.2. Описание портов	3
1.3. Формат команд	4
1.4. Формат сообщений сервера	5
1.5. Описание команд и параметров	6
1.6. Описание сообщений каналов передачи данных	36
1.7. Описание сообщений изменения состояния удаленного устройства	37
1.8. Описание типов параметров	39

1. УДАЛЕННОЕ УПРАВЛЕНИЕ

1.1. Введение

Сервер – ПО, которое запускается на ПЭВМ с установленным устройством ОСПЧ, принимает подключение клиента, принимает команды от клиента, управляет устройством.

Клиент – ПО, подключающееся по TCP/IP протоколу к серверу, отправляет серверу команды управления устройством, принимает данные, не имеет физического подключения к устройству.

Процедура запуска ПО «ОСПЧ» в режиме удаленного управления описана в соответствующем разделе руководства оператора.

1.2. Описание портов

Для удаленного управления используются TCP сокет для пяти двунаправленных каналов передачи команд и данных:

Канал	Порт	Сервер	Клиент
"commandChannel"	[n]	Прием команд/конфигураций	Прием ответов на команды
"dma3Channel"	[n]	Только запрос статуса	Прием данных (DMA3) от сервера
"iqChannel"	[n]	Только запрос статуса	Прием данных (IQ) от сервера
"dmdChannel"	[n]	Только запрос статуса	Прием данных (DMD) от сервера
"signalChannel"	[n]	Только запрос статуса	Прием сигналов изменения состояния удаленного устройства

Базовый порт, используемый по умолчанию, n=30000.

После подключения к первому сокету сервер готов к работе. При необходимости получать данные и следить за изменением состояния удаленного устройства следует подключиться к каналам данных и сигнальному каналу. Всего сервер ожидает подключения по 5 сокетам, после чего будет отклонять дальнейшие подключения до тех пор, пока не будет разорвано соединение по командному каналу (при этом будут разорваны все соединения с данным клиентом и сервер перейдет в режим ожидания подключения нового клиента).

Для определения конфигурации текущего канала можно использовать запрос статуса (см. раздел [1.3](#) и [1.5](#)).

Сигнальный канал предназначен для оповещения о событиях на удаленном устройстве. Сообщения сервера, передаваемые по сигнальному каналу, описаны в разделе [1.7](#).

1.3. Формат команд

Для передачи команд используется текстовый формат обмена данными JSON. Первые 8 байт сообщения должны содержать размер команды в байтах (64-битное целое).

Формат команды:

```
{
  "requestType": тип_запроса,
  "command": "имя_команды",
  "args": [
    {
      "valueType": "тип_параметра1"
      "value": "значение_параметра1",
    },
    {
      "valueType": "тип_параметра2"
      "value": "значение_параметра2",
    },
    ...
  ]
}
```

requestType – тип запроса к серверу, может принимать следующие значения:

- 0 – запрос статуса канала,
- 1 – без возвращения результата (для такого типа запроса сервер не возвращает результат выполнения команды, можно использовать для ускорения работы по сети, либо если нет необходимости в его получении),
- 2 – с возвращение результата выполнения команды

command – имя команды, которую необходимо выполнить, описание всех доступных команд находится в разделе [1.5](#)

args – массив параметров

Для каждого параметра передается valueType – тип параметра (все типы описаны в разделе [1.8](#)) и value – значение параметра в строковом виде.

1.4. Формат сообщений сервера

На запросы, для которых предусмотрен возврат результата выполнения команды (типы 0 и 2) и для сообщений по каналам передачи данных и сигнальному каналу, сервер отправляет ответ. Первые 8 байт сообщения сервера содержат размер ответа в байтах (64-битное целое).

Формат ответа:

```
{  
  "channel": "идентификатор_канала",  
  "command": "имя_команды",  
  "status": "статус",  
  "error": "ошибка",  
  "valueType": "тип_значения",  
  "value": "значение",  
}
```

channel – идентификатор канала, по которому идет передача (commandChannel, dma3Channel, iqChannel, dmdChannel, signalChannel), список идентификаторов и описание в разделе [1.2](#)

command – для командного канала – имя выполненной команды; Для каналов передачи данных – тип данных (dma3Data, iqData, dmdData, описание в разделе [1.6](#)); Для сигнального канала – идентификатор сигнала (описание в разделе [1.7](#))

status – статус выполнения команды (ok, error)

error – описание ошибки

valueType – тип параметра (все типы описаны в разделе [1.8](#))

value – значение результата в строковом виде

1.5. Описание команд и параметров

status	
<i>Описание</i>	Запрос готовности канала. Используется для проверки соединения, а также позволяет определить конфигурацию данного канала
<i>Параметры запроса</i>	requestType = 0 (обязателен) массив аргументов - пустой
<i>Параметры результата</i>	(string) – состояние канала ok – канал готов к работе
<i>Пример запроса</i>	{ "requestType": 0, "command": "status", "args": [] }
<i>Пример ответа</i>	{ "channel": "dma3Channel", "command": "status", "valueType": "string", "value": "ok", "status": "ok", "error": "" }

isActive	
<i>Описание</i>	Запрос готовности устройства
<i>Параметры запроса</i>	requestType = 2 массив аргументов - пустой
<i>Параметры результата</i>	(bool) – состояние устройства true – устройство готово к работе false – устройство неактивно
<i>Пример запроса</i>	{ "requestType": 2, "command": "isActive", "args": [] }
<i>Пример ответа</i>	{ "channel": "commandChannel", "command": "isActive", "valueType": "bool", "value": "true", "status": "ok", "error": "" }

dataStart	
<i>Описание</i>	Инициализация передачи данных
<i>Параметры запроса</i>	requestType = 1 или 2 (DataFormat) – формат данных (BufferSize) – размер буфера (см. раздел 1.8)
<i>Параметры результата</i>	(int) – статус выполнения команды
<i>Пример запроса</i>	<pre>{ "requestType": 2, "command": "dataStart", "args": [{ "valueType": "DataFormat", "value": "7" }, { "valueType": "BufferSize", "value": "131072" }] }</pre>
<i>Пример ответа</i>	<pre>{ "channel": "commandChannel", "command": "dataStart", "valueType": "int", "value": "0", "status": "ok", "error": "" }</pre>

getData	
<i>Описание</i>	Запрос буфера данных
<i>Параметры запроса</i>	requestType = 1 или 2 (DataFormat) – формат данных (см. раздел 1.8) (bool) – непрерывная передача данных
<i>Параметры результата</i>	(int) – статус выполнения команды
<i>Пример запроса</i>	<pre>{ "requestType": 2, "command": "dataStart", "args": [{ "valueType": "DataFormat", "value": "7" }, { "valueType": "bool", "value": "false" }] }</pre>
<i>Пример ответа</i>	<pre>{ "channel": "commandChannel", "command": "getData", "valueType": "int", "value": "0", "status": "ok", "error": "" }</pre>

dataStop	
<i>Описание</i>	Завершение передачи данных
<i>Параметры запроса</i>	requestType = 1 или 2 (DataFormat) – формат данных (см. раздел 1.8)
<i>Параметры результата</i>	(int) – статус выполнения команды
<i>Пример запроса</i>	<pre>{ "requestType": 2, "command": "dataStop", "args": [{ "valueType": "DataFormat", "value": "7" }] }</pre>
<i>Пример ответа</i>	<pre>{ "channel": "commandChannel", "command": "dataStop", "valueType": "int", "value": "0", "status": "ok", "error": "" }</pre>

getIqDataSize	
<i>Описание</i>	Размер буфера данных для IQ канала
<i>Параметры запроса</i>	requestType = 2 (bool) – для режима записи сигнала в файл (DataFormat) – формат данных (см. раздел 1.8)
<i>Параметры результата</i>	(uint) – размер буфера в байтах
<i>Пример запроса</i>	<pre>{ "requestType": 2, "command": "getIqDataSize", "args": [{ "valueType": "bool", "value": "0" }, { "valueType": "DataFormat", "value": "7" }] }</pre>
<i>Пример ответа</i>	<pre>{ "channel": "commandChannel", "command": "getIqDataSize", "valueType": "uint", "value": "1048576", "status": "ok", "error": "" }</pre>

checkDataErrors	
<i>Описание</i>	Проверка ошибок передачи данных
<i>Параметры запроса</i>	requestType = 2 (DataFormat) – формат данных (см. раздел 1.8)
<i>Параметры результата</i>	(int) – статус ошибки канала данных
<i>Пример запроса</i>	<pre>{ "requestType": 2, "command": "checkDataErrors", "args": [{ "valueType": "DataFormat", "value": "7" }] }</pre>

<i>Пример ответа</i>	<pre>{ "channel": "commandChannel", "command": "checkDataErrors", "valueType": "int", "value": "0", "status": "ok", "error": "" }</pre>
----------------------	---

getUncorruptedDataSize	
<i>Описание</i>	Размер неповрежденного блока данных
<i>Параметры запроса</i>	requestType = 2 (DataFormat) – формат данных (см. раздел 1.8)
<i>Параметры результата</i>	(int64) – размер блока данных
<i>Пример запроса</i>	<pre>{ "requestType": 2, "command": "getUncorruptedDataSize", "args": [{ "valueType": "DataFormat", "value": "7" }] }</pre>
<i>Пример ответа</i>	<pre>{ "channel": "commandChannel", "command": "getUncorruptedDataSize", "valueType": "int64", "value": "0", "status": "ok", "error": "" }</pre>

getDmdDataExParam	
<i>Описание</i>	Чтение параметров данных канала DMD
<i>Параметры запроса</i>	requestType = 2 массив аргументов - пустой
<i>Параметры результата</i>	(DmdDataExParam) – (см. раздел 1.8)
<i>Пример запроса</i>	<pre>{ "requestType": 2, "command": "getDmdDataExParam", "args": [] }</pre>
<i>Пример ответа</i>	<pre>{ "channel": "commandChannel", "command": "getDmdDataExParam", "valueType": "DmdDataExParam", "value": " <см. раздел 1.8> ", "status": "ok", "error": "" }</pre>

setDmdDataExParam	
<i>Описание</i>	Установка параметров данных канала DMD
<i>Параметры запроса</i>	requestType = 1 (DmdDataExParam) – (см. раздел 1.8)
<i>Параметры результата</i>	Без результата
<i>Пример запроса</i>	<pre>{ "requestType": 1, "command": "setDmdDataExParam", "args": [{ "valueType": "DmdDataExParam", "value": " <см. раздел 1.8> " }] }</pre>

<i>Пример ответа</i>	нет
----------------------	-----

deviceType	
<i>Описание</i>	Тип устройства
<i>Параметры запроса</i>	requestType = 2 массив аргументов - пустой
<i>Параметры результата</i>	(DevType) – тип устройства (см. раздел 1.8)
<i>Пример запроса</i>	{ "requestType": 2, "command": "deviceType", "args": [] }
<i>Пример ответа</i>	{ "channel": "commandChannel", "command": "deviceType", "valueType": "DevType", "value": "1e", "status": "ok", "error": "" }

decoderType	
<i>Описание</i>	Версия декодера
<i>Параметры запроса</i>	requestType = 2 массив аргументов - пустой
<i>Параметры результата</i>	(DecoderVersion) – версия декодера (см. раздел 1.8)
<i>Пример запроса</i>	{ "requestType": 2, "command": "decoderType", "args": [] }
<i>Пример ответа</i>	{ "channel": "commandChannel", "command": "decoderType", "valueType": "DecoderVersion", "value": "3", "status": "ok", "error": "" }

signalType	
<i>Описание</i>	Тип сигнала
<i>Параметры запроса</i>	requestType = 2 массив аргументов - пустой
<i>Параметры результата</i>	(SignalType) – тип сигнала (см. раздел 1.8)
<i>Пример запроса</i>	{ "requestType": 2, "command": "signalType", "args": [] }
<i>Пример ответа</i>	{ "channel": "commandChannel", "command": "signalType", "valueType": "SignalType", "value": "3", "status": "ok", "error": "" }

symbolRate	
<i>Описание</i>	Вид скорости кодирования сигнала
<i>Параметры запроса</i>	requestType = 2 массив аргументов - пустой
<i>Параметры результата</i>	(symbolRate) – вид скорости кодирования сигнала (см. раздел 1.8)
<i>Пример запроса</i>	{ "requestType": 2, "command": "symbolRate", "args": [] }
<i>Пример ответа</i>	{ "channel": "commandChannel", "command": "symbolRate", "valueType": "SymbolRate", "value": "3", "status": "ok", "error": "" }

constellationType	
<i>Описание</i>	Тип кодирования сигнального созвездия
<i>Параметры запроса</i>	requestType = 2 массив аргументов - пустой
<i>Параметры результата</i>	(ScType) – тип кодирования созвездия (см. раздел 1.8)
<i>Пример запроса</i>	{ "requestType": 2, "command": "constellationType", "args": [] }
<i>Пример ответа</i>	{ "channel": "commandChannel", "command": "constellationType", "valueType": "ScType", "value": "2", "status": "ok", "error": "" }

clockFrequency	
<i>Описание</i>	Запрос тактовой частоты
<i>Параметры запроса</i>	requestType = 2 массив аргументов - пустой
<i>Параметры результата</i>	(double) – значение тактовой частоты
<i>Пример запроса</i>	{ "requestType": 2, "command": "clockFrequency", "args": [] }
<i>Пример ответа</i>	{ "channel": "commandChannel", "command": "clockFrequency", "valueType": "double", "value": "1000000", "status": "ok", "error": "" }

setClockFrequency	
<i>Описание</i>	Установка тактовой частоты
<i>Параметры запроса</i>	requestType = 1 или 2 (double) – значение частоты в Гц
<i>Параметры результата</i>	(int) – статус выполнения команды
<i>Пример запроса</i>	{ "requestType": 2, "command": "setClockFrequency", "args": [{ "valueType": "double", "value": "2000000" }] }
<i>Пример ответа</i>	{ "channel": "commandChannel", "command": "setClockFrequency", "valueType": "int", "value": "0", "status": "ok", "error": "" }

setModulation	
<i>Описание</i>	Установка типа модуляции
<i>Параметры запроса</i>	requestType = 1 или 2 (SignalType) – тип сигнала (см. раздел 1.8) (SymbolRate) – вид скорости кодирования сигнала (см. раздел 1.8) (ScType) – тип кодирования созвездия (см. раздел 1.8)
<i>Параметры результата</i>	(int) – статус выполнения команды
<i>Пример запроса</i>	{ "requestType": 2, "command": "setModulation", "args": [{ "valueType": "SignalType", "value": "3" }, { "valueType": "SymbolRate", "value": "3" }, { "valueType": "ScType", "value": "0" }] }
<i>Пример ответа</i>	{ "channel": "commandChannel", "command": "setModulation", "valueType": "int", "value": "0", "status": "ok", "error": "" }

clockMin	
<i>Описание</i>	Минимальная тактовая частота
<i>Параметры запроса</i>	requestType = 2 массив аргументов - пустой
<i>Параметры результата</i>	(double) – значение минимально тактовой частоты
<i>Пример запроса</i>	{ "requestType": 2, "command": "clockMin", "args": [] }
<i>Пример ответа</i>	{ "channel": "commandChannel", "command": "clockMin", "valueType": "double", "value": "2000", "status": "ok", "error": "" }

clockMax	
<i>Описание</i>	Максимальная тактовая частота
<i>Параметры запроса</i>	requestType = 2 массив аргументов - пустой
<i>Параметры результата</i>	(double) – значение максимальной тактовой частоты
<i>Пример запроса</i>	{ "requestType": 2, "command": "clockMax", "args": [] }
<i>Пример ответа</i>	{ "channel": "commandChannel", "command": "clockMax", "valueType": "double", "value": "85000000", "status": "ok", "error": "" }

carrierFrequency	
<i>Описание</i>	Запрос несущей частоты
<i>Параметры запроса</i>	requestType = 2 массив аргументов - пустой
<i>Параметры результата</i>	(double) – значение несущей частоты
<i>Пример запроса</i>	{ "requestType": 2, "command": "carrierFrequency", "args": [] }
<i>Пример ответа</i>	{ "channel": "commandChannel", "command": "carrierFrequency", "valueType": "double", "value": "1000000000", "status": "ok", "error": "" }

setCarrierFrequency	
<i>Описание</i>	Установка несущей частоты
<i>Параметры запроса</i>	requestType = 1 или 2 (double) – значение частоты в Гц
<i>Параметры результата</i>	(int) – статус выполнения команды
<i>Пример запроса</i>	{ "requestType": 2, "command": "setCarrierFrequency", "args": [{ "valueType": "double", "value": "3500000000" }] }
<i>Пример ответа</i>	{ "channel": "commandChannel", "command": "setCarrierFrequency", "valueType": "int", "value": "0", "status": "ok", "error": "" }

carrierMin	
<i>Описание</i>	Минимальная несущая частота
<i>Параметры запроса</i>	requestType = 2 массив аргументов - пустой
<i>Параметры результата</i>	(double) – значение минимально несущей частоты
<i>Пример запроса</i>	{ "requestType": 2, "command": "carrierMin", "args": [] }
<i>Пример ответа</i>	{ "channel": "commandChannel", "command": "carrierMin", "valueType": "double", "value": "950000000", "status": "ok", "error": "" }

carrierMax	
<i>Описание</i>	Максимальная несущая частота
<i>Параметры запроса</i>	requestType = 2 массив аргументов - пустой
<i>Параметры результата</i>	(double) – значение максимальной несущей частоты
<i>Пример запроса</i>	{ "requestType": 2, "command": "carrierMax", "args": [] }
<i>Пример ответа</i>	{ "channel": "commandChannel", "command": "carrierMax", "valueType": "double", "value": "2150000000", "status": "ok", "error": "" }

panoramaMaxViewBand	
<i>Описание</i>	Максимальная полоса обзора панорамы
<i>Параметры запроса</i>	requestType = 2 массив аргументов - пустой
<i>Параметры результата</i>	(double) – максимальная полоса обзора панорамы в Гц
<i>Пример запроса</i>	{ "requestType": 2, "command": "panoramaMaxViewBand", "args": [] }
<i>Пример ответа</i>	{ "channel": "commandChannel", "command": "panoramaMaxViewBand", "valueType": "double", "value": "90000000", "status": "ok", "error": "" }

sampleFrequency	
<i>Описание</i>	Частота дискретизации
<i>Параметры запроса</i>	requestType = 2 массив аргументов - пустой
<i>Параметры результата</i>	(double) – частота дискретизации в Гц
<i>Пример запроса</i>	{ "requestType": 2, "command": "sampleFrequency", "args": [] }
<i>Пример ответа</i>	{ "channel": "commandChannel", "command": "sampleFrequency", "valueType": "double", "value": "280000000", "status": "ok", "error": "" }

IConvertorType	
<i>Описание</i>	Тип L-конвертора
<i>Параметры запроса</i>	requestType = 2 массив аргументов - пустой
<i>Параметры результата</i>	(int) – тип L-конвертора
<i>Пример запроса</i>	{ "requestType": 2, "command": "IConvertorType", "args": [] }
<i>Пример ответа</i>	{ "channel": "commandChannel", "command": "IConvertorType", "valueType": "int", "value": "4", "status": "ok", "error": "" }

filterType	
<i>Описание</i>	Тип фильтра
<i>Параметры запроса</i>	requestType = 2 массив аргументов - пустой
<i>Параметры результата</i>	(int) – тип фильтра
<i>Пример запроса</i>	{ "requestType": 2, "command": "filterType", "args": [] }
<i>Пример ответа</i>	{ "channel": "commandChannel", "command": "filterType", "valueType": "int", "value": "0", "status": "ok", "error": "" }

setFilterType	
<i>Описание</i>	Установка типа фильтра
<i>Параметры запроса</i>	requestType = 1 или 2 (int) – тип фильтра
<i>Параметры результата</i>	(int) – статус выполнения команды
<i>Пример запроса</i>	{ "requestType": 2, "command": "setFilterType", "args": [{ "valueType": "int", "value": "0" }] }
<i>Пример ответа</i>	{ "channel": "commandChannel", "command": " setFilterType ", "valueType": "int", "value": "0", "status": "ok", "error": "" }

pllBand	
<i>Описание</i>	Полоса ФАПЧ: 0 - 1/50, 1 - 1/100, 2 - 1/200, 3 - 1/400
<i>Параметры запроса</i>	requestType = 2 массив аргументов - пустой
<i>Параметры результата</i>	(int) – полоса ФАПЧ
<i>Пример запроса</i>	{ "requestType": 2, "command": "pllBand", "args": [] }
<i>Пример ответа</i>	{ "channel": "commandChannel", "command": "pllBand", "valueType": "int", "value": "2", "status": "ok", "error": "" }

setPllBand	
<i>Описание</i>	Установка полосы ФАПЧ
<i>Параметры запроса</i>	requestType = 1 или 2 (int) – полоса ФАПЧ (0 - 1/50, 1 - 1/100, 2 - 1/200, 3 - 1/400)
<i>Параметры результата</i>	(int) – статус выполнения команды
<i>Пример запроса</i>	{ "requestType": 2, "command": "setPllBand", "args": [{ "valueType": "int", "value": "0" }] }
<i>Пример ответа</i>	{ "channel": "commandChannel", "command": "setPllBand", "valueType": "int", "value": "0", "status": "ok", "error": "" }

carrierTracking	
<i>Описание</i>	Слежение за несущей
<i>Параметры запроса</i>	requestType = 2 массив аргументов - пустой
<i>Параметры результата</i>	(bool) – слежение за несущей включено(выключено)
<i>Пример запроса</i>	{ "requestType": 2, "command": "carrierTracking", "args": [] }
<i>Пример ответа</i>	{ "channel": "commandChannel", "command": "carrierTracking", "valueType": "bool", "value": "true", "status": "ok", "error": "" }

setCarrierTracking	
<i>Описание</i>	Установка слежения за несущей
<i>Параметры запроса</i>	requestType = 1 или 2 (bool) – вкл/выкл
<i>Параметры результата</i>	(int) – статус выполнения команды
<i>Пример запроса</i>	{ "requestType": 2, "command": "setCarrierTracking", "args": [{ "valueType": "bool", "value": "true" }] }
<i>Пример ответа</i>	{ "channel": "commandChannel", "command": "setCarrierTracking", "valueType": "int", "value": "0", "status": "ok", "error": "" }

clockTracking	
<i>Описание</i>	Слежение за тактовой
<i>Параметры запроса</i>	requestType = 2 массив аргументов - пустой
<i>Параметры результата</i>	(bool) – слежение за тактовой включено(выключено)
<i>Пример запроса</i>	{ "requestType": 2, "command": "clockTracking", "args": [] }
<i>Пример ответа</i>	{ "channel": "commandChannel", "command": "clockTracking", "valueType": "bool", "value": "true", "status": "ok", "error": "" }

setClockTracking	
<i>Описание</i>	Установка слежения за тактовой
<i>Параметры запроса</i>	requestType = 1 или 2 (bool) – ВКЛ/ВЫКЛ
<i>Параметры результата</i>	(int) – статус выполнения команды
<i>Пример запроса</i>	{ "requestType": 2, "command": "setClockTracking", "args": [{ "valueType": "bool", "value": "true" }] }
<i>Пример ответа</i>	{ "channel": "commandChannel", "command": "setClockTracking", "valueType": "int", "value": "0", "status": "ok", "error": "" }

afc	
<i>Описание</i>	Автоматическая подстройка частоты
<i>Параметры запроса</i>	requestType = 2 массив аргументов - пустой
<i>Параметры результата</i>	(bool) – автоматическая подстройка частоты включена(выключена)
<i>Пример запроса</i>	{ "requestType": 2, "command": "afc", "args": [] }
<i>Пример ответа</i>	{ "channel": "commandChannel", "command": "afc", "valueType": "bool", "value": "false", "status": "ok", "error": "" }

setAfc	
<i>Описание</i>	Установка авто. подстройки частоты
<i>Параметры запроса</i>	requestType = 1 или 2 (bool) – вкл/выкл
<i>Параметры результата</i>	(int) – статус выполнения команды
<i>Пример запроса</i>	{ "requestType": 2, "command": "setAfc", "args": [{ "valueType": "bool", "value": "true" }] }
<i>Пример ответа</i>	{ "channel": "commandChannel", "command": "setAfc", "valueType": "int", "value": "0", "status": "ok", "error": "" }

adaptiveCorrector	
<i>Описание</i>	Адаптивный корректор
<i>Параметры запроса</i>	requestType = 2 массив аргументов - пустой
<i>Параметры результата</i>	(int) – адаптивный корректор включен – 1 (выключен - 0)
<i>Пример запроса</i>	{ "requestType": 2, "command": "adaptiveCorrector", "args": [] }
<i>Пример ответа</i>	{ "channel": "commandChannel", "command": "adaptiveCorrector", "valueType": "int", "value": "0", "status": "ok", "error": "" }

setAdaptiveCorrector	
<i>Описание</i>	Установка авто. подстройки частоты
<i>Параметры запроса</i>	requestType = 1 или 2 (int) – вкл – 1, выкл - 0
<i>Параметры результата</i>	(int) – статус выполнения команды
<i>Пример запроса</i>	{ "requestType": 2, "command": "setAdaptiveCorrector", "args": [{ "valueType": "int", "value": "1" }] }
<i>Пример ответа</i>	{ "channel": "commandChannel", "command": "setAdaptiveCorrector", "valueType": "int", "value": "0", "status": "ok", "error": "" }

reference	
<i>Описание</i>	Тип опорного генератора
<i>Параметры запроса</i>	requestType = 2 массив аргументов - пустой
<i>Параметры результата</i>	(int) – 0 – внутренний, 1 - внешний
<i>Пример запроса</i>	{ "requestType": 2, "command": "reference", "args": [] }
<i>Пример ответа</i>	{ "channel": "commandChannel", "command": "reference", "valueType": "int", "value": "0", "status": "ok", "error": "" }

setReference	
<i>Описание</i>	Установка типа опорного генератора
<i>Параметры запроса</i>	requestType = 1 или 2 (int) – внутренний – 0, внешний - 1
<i>Параметры результата</i>	(int) – статус выполнения команды
<i>Пример запроса</i>	{ "requestType": 2, "command": "setReference", "args": [{ "valueType": "int", "value": "1" }] }
<i>Пример ответа</i>	{ "channel": "commandChannel", "command": "setReference", "valueType": "int", "value": "0", "status": "ok", "error": "" }

clockInversion	
<i>Описание</i>	Инверсия тактовой частоты
<i>Параметры запроса</i>	requestType = 2 массив аргументов - пустой
<i>Параметры результата</i>	(bool) – инверсия тактовой частоты включена(выключена)
<i>Пример запроса</i>	{ "requestType": 2, "command": "clockInversion", "args": [] }
<i>Пример ответа</i>	{ "channel": "commandChannel", "command": "clockInversion", "valueType": "bool", "value": "false", "status": "ok", "error": "" }

setClockInversion	
<i>Описание</i>	Установка инверсии тактовой частоты
<i>Параметры запроса</i>	requestType = 1 или 2 (bool) – вкл/выкл
<i>Параметры результата</i>	(int) – статус выполнения команды
<i>Пример запроса</i>	{ "requestType": 2, "command": "setClockInversion", "args": [{ "valueType": "bool", "value": "false" }] }
<i>Пример ответа</i>	{ "channel": "commandChannel", "command": "setClockInversion", "valueType": "int", "value": "0", "status": "ok", "error": "" }

setMGC1PanoramaLValue	
<i>Описание</i>	Установка значения РУ1
<i>Параметры запроса</i>	requestType = 1 (double) – значение усиления в дБ
<i>Параметры результата</i>	Без результата
<i>Пример запроса</i>	{ "requestType": 1, "command": "setMGC1PanoramaLValue", "args": [{ "valueType": "double", "value": "0" }] }
<i>Пример ответа</i>	нет

mgcAction	
<i>Описание</i>	Действие для ручного управления РУ
<i>Параметры запроса</i>	requestType = 1 или 2 (int) – номер РУ (0 или 1) (MgcAction) – увеличить – 0, уменьшить - 1
<i>Параметры результата</i>	(int) – статус выполнения команды
<i>Пример запроса</i>	{ "requestType": 2, "command": "mgcAction", "args": [{ "valueType": "int", "value": "0" }, { "valueType": "MgcAction", "value": "0" }] }
<i>Пример ответа</i>	{ "channel": "commandChannel", "command": "mgcAction", "valueType": "int", "value": "0", "status": "ok", "error": "" }

isMgcEnable	
<i>Описание</i>	Ручное управление РУ
<i>Параметры запроса</i>	requestType = 2 (int) – номер РУ (0 или 1)
<i>Параметры результата</i>	(bool) – ручное управление РУ включено(выключено)
<i>Пример запроса</i>	{ "requestType": 2, "command": "isMgcEnable", "args": [{ "valueType": "int", "value": "0" }] }
<i>Пример ответа</i>	{ "channel": "commandChannel", "command": "isMgcEnable", "valueType": "bool", "value": "false", "status": "ok", "error": "" }

setMgcEnable	
<i>Описание</i>	Установка ручного управления РУ
<i>Параметры запроса</i>	requestType = 1 или 2 (int) – номер РУ (0 или 1) (bool) – вкл/выкл
<i>Параметры результата</i>	(int) – статус выполнения команды
<i>Пример запроса</i>	{ "requestType": 2, "command": "setMgcEnable", "args": [{ "valueType": "int", "value": "0" }, { "valueType": "bool", "value": "true" }] }
<i>Пример ответа</i>	{ "channel": "commandChannel", "command": "setMgcEnable", "valueType": "int", "value": "0", "status": "ok", "error": "" }

gainControl2Mode	
<i>Описание</i>	Режим АРУ2
<i>Параметры запроса</i>	requestType = 2 массив аргументов - пустой
<i>Параметры результата</i>	(GC2Mode) – режим АРУ2 (см. раздел 1.8)
<i>Пример запроса</i>	{ "requestType": 2, "command": "gainControl2Mode", "args": [] }
<i>Пример ответа</i>	{ "channel": "commandChannel", "command": "gainControl2Mode", "valueType": "GC2Mode", "value": "1", "status": "ok", "error": "" }

setGainControl2Mode	
<i>Описание</i>	Установка режима АРУ2
<i>Параметры запроса</i>	requestType = 1 или 2 (GC2Mode) – режим АРУ2 (см. раздел 1.8)
<i>Параметры результата</i>	(int) – статус выполнения команды
<i>Пример запроса</i>	{ "requestType": 2, "command": "setGainControl2Mode", "args": [{ "valueType": "GC2Mode", "value": "1" }] }
<i>Пример ответа</i>	{ "channel": "commandChannel", "command": "setGainControl2Mode", "valueType": "int", "value": "0", "status": "ok", "error": "" }

gainControl2UserTimeHigh	
<i>Описание</i>	Постоянная времени 1 пользовательского режима АРУ2
<i>Параметры запроса</i>	requestType = 2 массив аргументов - пустой
<i>Параметры результата</i>	(int) – первый параметр пользовательского режима АРУ2
<i>Пример запроса</i>	{ "requestType": 2, "command": "gainControl2UserTimeHigh", "args": [] }
<i>Пример ответа</i>	{ "channel": "commandChannel", "command": "gainControl2UserTimeHigh", "valueType": "int", "value": "4", "status": "ok", "error": "" }

gainControl2UserTimeLow	
<i>Описание</i>	Постоянная времени 2 пользовательского режима АРУ2
<i>Параметры запроса</i>	requestType = 2 массив аргументов - пустой
<i>Параметры результата</i>	(int) – второй параметр пользовательского режима АРУ2
<i>Пример запроса</i>	{ "requestType": 2, "command": "gainControl2UserTimeLow", "args": [] }
<i>Пример ответа</i>	{ "channel": "commandChannel", "command": "gainControl2UserTimeLow", "valueType": "int", "value": "4", "status": "ok", "error": "" }

gainControl2UserCoeff	
<i>Описание</i>	Коэффициент пользовательского режима АРУ2
<i>Параметры запроса</i>	requestType = 2 массив аргументов - пустой
<i>Параметры результата</i>	(double) – коэффициент пользовательского режима АРУ2
<i>Пример запроса</i>	{ "requestType": 2, "command": "gainControl2UserCoef", "args": [] }
<i>Пример ответа</i>	{ "channel": "commandChannel", "command": "gainControl2UserCoef", "valueType": "double", "value": "1", "status": "ok", "error": "" }

setGainControl2UserParameters	
<i>Описание</i>	Установка параметров для пользовательского режима АРУ2
<i>Параметры запроса</i>	requestType = 1 или 2 (int) – постоянная времени 1 (int) – постоянная времени 2 (double) – коэффициент АРУ2
<i>Параметры результата</i>	(int) – статус выполнения команды
<i>Пример запроса</i>	{ "requestType": 2, "command": "setGainControl2UserParameters", "args": [{ "valueType": "int", "value": "4" }, { "valueType": "int", "value": "4" }, { "valueType": "double", "value": "1" }] }
<i>Пример ответа</i>	{ "channel": "commandChannel", "command": "setGainControl2UserParameters", "valueType": "int", "value": "0", "status": "ok", "error": "" }

displayedSnrType	
<i>Описание</i>	Тип ОСШ
<i>Параметры запроса</i>	requestType = 2 массив аргументов - пустой
<i>Параметры результата</i>	(int) – тип ОСШ (0 – Ps/Pn, 1 – Eb/No)
<i>Пример запроса</i>	{ "requestType": 2, "command": "displayedSnrType", "args": [] }
<i>Пример ответа</i>	{ "channel": "commandChannel", "command": "displayedSnrType", "valueType": "int", "value": "0", "status": "ok", "error": "" }

setDisplayeдSnrType	
<i>Описание</i>	Установка режима АРУ2
<i>Параметры запроса</i>	requestType = 1 или 2 (int) – тип ОСШ (0 – Ps/Pn, 1 – Eb/No)
<i>Параметры результата</i>	(int) – статус выполнения команды
<i>Пример запроса</i>	{ "requestType": 2, "command": "setDisplayeдSnrType", "args": [{ "valueType": "GC2Mode", "value": "1" }] }
<i>Пример ответа</i>	{ "channel": "commandChannel", "command": "setDisplayeдSnrType", "valueType": "int", "value": "0", "status": "ok", "error": "" }

getBoardStatus	
<i>Описание</i>	Чтение статуса платы
<i>Параметры запроса</i>	requestType = 2 массив аргументов - пустой
<i>Параметры результата</i>	(BoardStatus) – (см. раздел 1.8)
<i>Пример запроса</i>	{ "requestType": 2, "command": "getBoardStatus", "args": [] }
<i>Пример ответа</i>	{ "channel": "commandChannel", "command": "getBoardStatus", "valueType": "BoardStatus", "value": "<см. раздел 1.8>", "status": "ok", "error": "" }

getBoardValues	
<i>Описание</i>	Чтение значений параметров платы
<i>Параметры запроса</i>	requestType = 2 массив аргументов - пустой
<i>Параметры результата</i>	(BoardValues) – (см. раздел 1.8)
<i>Пример запроса</i>	{ "requestType": 2, "command": "getBoardValues", "args": [] }
<i>Пример ответа</i>	{ "channel": "commandChannel", "command": "getBoardValues", "valueType": "BoardValues", "value": "<см. раздел 1.8>", "status": "ok", "error": "" }

getBoardTemperature	
<i>Описание</i>	Чтение температурных показателей платы
<i>Параметры запроса</i>	requestType = 2 массив аргументов - пустой
<i>Параметры результата</i>	(BoardTemperature) – (см. раздел 1.8)
<i>Пример запроса</i>	{ "requestType": 2, "command": "getBoardTemperature", "args": [] }
<i>Пример ответа</i>	{ "channel": "commandChannel", "command": "getBoardTemperature", "valueType": "BoardTemperature", "value": "<см. раздел 1.8>", "status": "ok", "error": "" }

getSnrCoeff	
<i>Описание</i>	Чтение коэффициентов ОСШ
<i>Параметры запроса</i>	requestType = 2 массив аргументов - пустой
<i>Параметры результата</i>	(SnrCoeff) – (см. раздел 1.8)
<i>Пример запроса</i>	{ "requestType": 2, "command": "getSnrCoeff", "args": [] }
<i>Пример ответа</i>	{ "channel": "commandChannel", "command": "getSnrCoeff", "valueType": "SnrCoeff", "value": "<см. раздел 1.8>", "status": "ok", "error": "" }

getDeviceConfiguration	
<i>Описание</i>	Чтение конфигурации устройства
<i>Параметры запроса</i>	requestType = 2 массив аргументов - пустой
<i>Параметры результата</i>	(DeviceConfiguration) – (см. раздел 1.8)
<i>Пример запроса</i>	{ "requestType": 2, "command": "getDeviceConfiguration", "args": [] }
<i>Пример ответа</i>	{ "channel": "commandChannel", "command": "getDeviceConfiguration", "valueType": "DeviceConfiguration", "value": "<см. раздел 1.8>", "status": "ok", "error": "" }

loadDeviceConfiguration	
<i>Описание</i>	Установка конфигурации устройства
<i>Параметры запроса</i>	requestType = 1 или 2 (DeviceConfiguration) – (см. раздел 1.8) (bool) – force load
<i>Параметры результата</i>	(int) – статус выполнения команды
<i>Пример запроса</i>	{ "requestType": 2, "command": "loadDeviceConfiguration", "args": [{ "valueType": "DeviceConfiguration", "value": "<см. раздел 1.8>" }, { "valueType": "bool", "value": "false" }] }
<i>Пример ответа</i>	{ "channel": "commandChannel", "command": "loadDeviceConfiguration", "valueType": "int", "value": "0", "status": "ok", "error": "" }

testSignalEnable	
<i>Описание</i>	Режим тестового сигнала в каналах данных
<i>Параметры запроса</i>	requestType = 2 массив аргументов - пустой
<i>Параметры результата</i>	(bool) – тестовый сигнал в каналах данных включен(выключен)
<i>Пример запроса</i>	{ "requestType": 2, "command": "testSignalEnable", "args": [] }
<i>Пример ответа</i>	{ "channel": "commandChannel", "command": "testSignalEnable", "valueType": "bool", "value": "false", "status": "ok", "error": "" }

setTestSignalEnable	
<i>Описание</i>	Установка режима тестового сигнала в каналах данных
<i>Параметры запроса</i>	requestType = 1 (bool) – вкл/выкл
<i>Параметры результата</i>	Без результата
<i>Пример запроса</i>	{ "requestType": 1, "command": "setTestSignalEnable", "args": [{ "valueType": "bool", "value": "true" }] }
<i>Пример ответа</i>	нет

getDecoderConfiguration	
<i>Описание</i>	Чтение конфигурации декодера
<i>Параметры запроса</i>	requestType = 2 массив аргументов - пустой
<i>Параметры результата</i>	(DecoderConfiguration) – (см. раздел 1.8)
<i>Пример запроса</i>	{ "requestType": 2, "command": "getDecoderConfiguration", "args": [] }
<i>Пример ответа</i>	{ "channel": "commandChannel", "command": "getDecoderConfiguration", "valueType": "DecoderConfiguration", "value": "<см. раздел 1.8>", "status": "ok", "error": "" }

loadDecoderConfiguration	
<i>Описание</i>	Установка конфигурации декодера
<i>Параметры запроса</i>	requestType = 1 или 2 (DecoderConfiguration) – (см. раздел 1.8) (bool) – force load
<i>Параметры результата</i>	(int) – статус выполнения команды
<i>Пример запроса</i>	{ "requestType": 2, "command": "loadDecoderConfiguration", "args": [{ "valueType": "DecoderConfiguration", "value": "<см. раздел 1.8>" }, { "valueType": "bool", "value": "false" }] }
<i>Пример ответа</i>	{ "channel": "commandChannel", "command": "loadDecoderConfiguration", "valueType": "int", "value": "0", "status": "ok", "error": "" }

getOspchDeviceBaseParameters	
<i>Описание</i>	Чтение основных параметров устройства
<i>Параметры запроса</i>	requestType = 2 массив аргументов - пустой
<i>Параметры результата</i>	(DeviceBaseParameters) – (см. раздел 1.8)
<i>Пример запроса</i>	{ "requestType": 2, "command": "getOspchDeviceBaseParameters", "args": [] }
<i>Пример ответа</i>	{ "channel": "commandChannel", "command": "getOspchDeviceBaseParameters", "valueType": "DeviceBaseParameters", "value": "<см. раздел 1.8>", "status": "ok", "error": "" }

getISOCVRTLoading	
<i>Описание</i>	Загрузка внутреннего буфера изохронного преобразователя
<i>Параметры запроса</i>	requestType = 2 массив аргументов - пустой
<i>Параметры результата</i>	(ISOCVRTLoading) – (см. раздел 1.8)
<i>Пример запроса</i>	{ "requestType": 2, "command": "getISOCVRTLoading", "args": [] }
<i>Пример ответа</i>	{ "channel": "commandChannel", "command": "getISOCVRTLoading", "valueType": "ISOCVRTLoading", "value": " <см. раздел 1.8> ", "status": "ok", "error": "" }

getPCIEFreq	
<i>Описание</i>	Частота шины PCIe
<i>Параметры запроса</i>	requestType = 2 массив аргументов - пустой
<i>Параметры результата</i>	(double) – частота шины
<i>Пример запроса</i>	{ "requestType": 2, "command": "getPCIEFreq", "args": [] }
<i>Пример ответа</i>	{ "channel": "commandChannel", "command": "getPCIEFreq", "valueType": "double", "value": "", "status": "ok", "error": "" }

getContinuousDataSpeed	
<i>Описание</i>	Чтение скоростей обмена данными
<i>Параметры запроса</i>	requestType = 2 массив аргументов - пустой
<i>Параметры результата</i>	(ContinuousDataSpeed) – (см. раздел 1.8)
<i>Пример запроса</i>	{ "requestType": 2, "command": "getContinuousDataSpeed", "args": [] }
<i>Пример ответа</i>	{ "channel": "commandChannel", "command": "getContinuousDataSpeed", "valueType": "ContinuousDataSpeed", "value": " <см. раздел 1.8> ", "status": "ok", "error": "" }

readReg	
<i>Описание</i>	Чтение параметров устройства из реестра
<i>Параметры запроса</i>	requestType = 2 (uint) – адрес
<i>Параметры результата</i>	(uint) – данные
<i>Пример запроса</i>	{ "requestType": 2, "command": "readReg", "args": [{ "valueType": "uint", "value": "0" }] }
<i>Пример ответа</i>	{ "channel": "commandChannel", "command": "readReg", "valueType": "uint", "value": "0", "status": "ok", "error": "" }

writeReg	
<i>Описание</i>	Запись параметров устройства в реестр
<i>Параметры запроса</i>	requestType = 1 (uint) – адрес (uint) – данные
<i>Параметры результата</i>	Без результата
<i>Пример запроса</i>	{ "requestType": 1, "command": "writeReg", "args": [{ "valueType": "uint", "value": "0" }, { "valueType": "uint", "value": "42" }] }
<i>Пример ответа</i>	нет

getDecoderApiVersion	
<i>Описание</i>	Версия библиотеки декодера
<i>Параметры запроса</i>	requestType = 2 массив аргументов - пустой
<i>Параметры результата</i>	(string) – версия библиотеки декодера
<i>Пример запроса</i>	{ "requestType": 2, "command": "getDecoderApiVersion", "args": [] }
<i>Пример ответа</i>	{ "channel": "commandChannel", "command": "getDecoderApiVersion", "valueType": "string", "value": "", "status": "ok", "error": "" }

readModuleVerEEPROM	
<i>Описание</i>	Чтение версии модуля из EEPROM
<i>Параметры запроса</i>	requestType = 2 массив аргументов - пустой
<i>Параметры результата</i>	(EEPROMData) – данные из EEPROM (см. раздел 1.8)
<i>Пример запроса</i>	{ "requestType": 2, "command": "readModuleVerEEPROM", "args": [] }
<i>Пример ответа</i>	{ "channel": "commandChannel", "command": "readModuleVerEEPROM", "valueType": "EEPROMData", "value": { "data": "DLE5g==", "status": 0 }, "status": "ok", "error": "" }

readUserEEPROMFull	
<i>Описание</i>	Чтение всех данных из EEPROM
<i>Параметры запроса</i>	requestType = 2 массив аргументов - пустой
<i>Параметры результата</i>	(EEPROMData) – данные из EEPROM (см. раздел 1.8)
<i>Пример запроса</i>	{ "requestType": 2, "command": "readUserEEPROMFull", "args": [] }
<i>Пример ответа</i>	{ "channel": "commandChannel", "command": "readUserEEPROMFull", "valueType": "EEPROMData", "value": { "data": "TMz4GfAyFOZMzdwY7DLE5g==", "status": 0 }, "status": "ok", "error": "" }

writeUserEEPROMFull	
<i>Описание</i>	Запись данных в EEPROM
<i>Параметры запроса</i>	requestType = 1 или 2 (base64) – данные
<i>Параметры результата</i>	(int) – статус выполнения команды
<i>Пример запроса</i>	{ "requestType": 2, "command": "writeUserEEPROMFull", "args": [{ "valueType": "base64", "value": "KioqKio=" }] }
<i>Пример ответа</i>	{ "channel": "commandChannel", "command": "writeUserEEPROMFull", "valueType": "int", "value": 0, "status": "ok", "error": "" }

readUserEEPROM	
<i>Описание</i>	Чтение данных из EEPROM по адресу
<i>Параметры запроса</i>	requestType = 2 (short) – адрес в EEPROM
<i>Параметры результата</i>	(EEPROMData) – данные из EEPROM (см. раздел 1.8)
<i>Пример запроса</i>	<pre>{ "requestType": 2, "command": "readUserEEPROM", "args": [{ "valueType": "short", "value": "0" }] }</pre>
<i>Пример ответа</i>	<pre>{ "channel": "commandChannel", "command": "readUserEEPROM", "valueType": "EEPROMData", "value": { "data": "GfAy==", "status": 0 }, "status": "ok", "error": "" }</pre>

writeUserEEPROM	
<i>Описание</i>	Чтение данных из EEPROM по адресу
<i>Параметры запроса</i>	requestType = 1 или 2 (short) – адрес в EEPROM (int) – значение (1 байт)
<i>Параметры результата</i>	(int) – статус выполнения команды
<i>Пример запроса</i>	<pre>{ "requestType": 2, "command": "writeUserEEPROM", "args": [{ "valueType": "short", "value": "0" }, { "valueType": "int", "value": "13" }] }</pre>
<i>Пример ответа</i>	<pre>{ "channel": "commandChannel", "command": "writeUserEEPROM", "valueType": "int", "value": 0, "status": "ok", "error": "" }</pre>

getLastErrorDescript	
<i>Описание</i>	Считывание описание последней ошибки
<i>Параметры запроса</i>	requestType = 2 массив аргументов - пустой
<i>Параметры результата</i>	(string) – текст описания ошибки
<i>Пример запроса</i>	<pre>{ "requestType": 2, "command": "getLastErrorDescript", "args": [] }</pre>

<i>Пример ответа</i>	{ "channel": "commandChannel", "command": "getLastErrorDescript", "valueType": "string", "value": "Open file error", "status": "ok", "error": "" }
----------------------	---

getDNA	
<i>Описание</i>	Считывание DNA кода
<i>Параметры запроса</i>	requestType = 2 массив аргументов - пустой
<i>Параметры результата</i>	(string) – DNA-код в текстовом виде
<i>Пример запроса</i>	{ "requestType": 2, "command": "getDNA", "args": [] }
<i>Пример ответа</i>	{ "channel": "commandChannel", "command": "getDNA", "valueType": "string", "value": "0000000000000000", "status": "ok", "error": "" }

getSnr	
<i>Описание</i>	Считывание DNA кода
<i>Параметры запроса</i>	requestType = 2 массив аргументов - пустой
<i>Параметры результата</i>	(double) – значение ОСШ
<i>Пример запроса</i>	{ "requestType": 2, "command": "getSnr", "args": [] }
<i>Пример ответа</i>	{ "channel": "commandChannel", "command": "getSnr", "valueType": "double", "value": "0.0", "status": "ok", "error": "" }

getIoCounters	
<i>Описание</i>	Считывание значений счетчиков данных
<i>Параметры запроса</i>	requestType = 2 массив аргументов - пустой
<i>Параметры результата</i>	(IoCounters) – значение счетчиков данных
<i>Пример запроса</i>	{ "requestType": 2, "command": "getIoCounters", "args": [] }
<i>Пример ответа</i>	{ "channel": "commandChannel", "command": "getIoCounters", "valueType": "IoCounters", "value": { "irqCnt": 0, "threadCnt": 0 }, "status": "ok",

	"error": "" }
--	---------------

setImitHwParameters	
<i>Описание</i>	Установка параметров аппаратного имитатора
<i>Параметры запроса</i>	requestType = 2 (HwImitParam) – параметры аппаратного имитатора (bool) - включение аппаратного имитатора
<i>Параметры результата</i>	(int) – статус выполнения команды
<i>Пример запроса</i>	{ "requestType": 2, "command": "setImitHwParameters", "args": [{ "valueType": "HwImitParam", "value": "<см. раздел 1.8>" }, { "valueType": "bool", "value": "true" }]}
<i>Пример ответа</i>	{ "channel": "commandChannel", "command": "setImitHwParameters", "valueType": "IoCounters", "value": { "irqCnt": 0, "threadCnt": 0 }, "status": "ok", "error": "" }

isImitHwStarted	
<i>Описание</i>	Считывание статуса аппаратного имитатора
<i>Параметры запроса</i>	requestType = 2 массив аргументов - пустой
<i>Параметры результата</i>	(bool) – статус аппаратного имитатора
<i>Пример запроса</i>	{ "requestType": 2, "command": "isImitHwStarted", "args": [] }
<i>Пример ответа</i>	{ "channel": "commandChannel", "command": "isImitHwStarted", "valueType": "bool", "value": "true", "status": "ok", "error": "" }

1.6. Описание сообщений каналов передачи данных

Для запроса данных необходимо использовать команды dataStart, getData, dataStop (см. раздел [1.5](#)). Данные кодируются в Base64 и отправляются по соответствующему каналу данных.

Канал dma3Channel

dma3Data	
<i>Описание</i>	Данные канала DMA3
<i>Параметры результата</i>	(base64) – массив двоичных данных
<i>Пример ответа</i>	<pre>{ "channel": "dma3Channel", "command": "dma3Data", "valueType": "base64", "value": "pDbM3sDIgCE0NmTeqMkQIVQ2fN54yZQhBDYs3ijJDCE8NujejMn ... +TMz4GfAyFOZMzdwY7DLE5g==", "status": "ok", "error": "" }</pre>

Канал iqChannel

iqData	
<i>Описание</i>	Данные канала IQ
<i>Параметры результата</i>	(base64) – массив двоичных данных
<i>Пример ответа</i>	<pre>{ "channel": "iqChannel", "command": "iqData", "valueType": "base64", "value": "MLCPsM3TchMv0Ihg1aNyQynwjtDXU3DjKDCNMNWzcmIAJG ... +BVo5YDqKBUyFEDldOxIGcQSQOYQ7w==", "status": "ok", "error": "" }</pre>

Канал dmdChannel

dmdData	
<i>Описание</i>	Данные канала DMD
<i>Параметры результата</i>	(base64) – массив двоичных данных
<i>Пример ответа</i>	<pre>{ "channel": "dmdChannel", "command": "dmdData", "valueType": "base64", "value": "1Nd8CZAnAPZw2FQJhCWA9yjaQAi4I1T38NykB ... +rDe9PUoIcQI9N1s96Ai9AX83UT5DCPwBHjcVPy4Iw==", "status": "ok", "error": "" }</pre>

1.7. Описание сообщений изменения состояния удаленного устройства

deviceChanged	
<i>Описание</i>	Сообщение об изменении источника данных
<i>Параметры результата</i>	нет
<i>Пример ответа</i>	{ "channel": "signalChannel", "command": "deviceChanged", "valueType": "", "value": "", "status": "ok", "error": "" }

clockChanged	
<i>Описание</i>	Сообщение об изменении тактовой частоты
<i>Параметры результата</i>	(double) – значение частоты в Гц
<i>Пример ответа</i>	{ "channel": "signalChannel", "command": "clockChanged", "valueType": "double", "value": "1100000", "status": "ok", "error": "" }

modulationChanged	
<i>Описание</i>	Сообщение об изменении вида модуляции
<i>Параметры результата</i>	нет
<i>Пример ответа</i>	{ "channel": "signalChannel", "command": "modulationChanged", "valueType": "", "value": "", "status": "ok", "error": "" }

carrierChanged	
<i>Описание</i>	Сообщение об изменении несущей частоты
<i>Параметры результата</i>	(double) – значение частоты в Гц
<i>Пример ответа</i>	{ "channel": "signalChannel", "command": "carrierChanged", "valueType": "double", "value": "999900000", "status": "ok", "error": "" }

freqLimitsChanged	
<i>Описание</i>	Сообщение об изменении значений максимальной и минимальной частоты
<i>Параметры результата</i>	нет

<i>Пример ответа</i>	{ "channel": "signalChannel", "command": "freqLimitsChanged", "valueType": "", "value": "", "status": "ok", "error": "" }
----------------------	--

freqTrackingParametersChanged	
<i>Описание</i>	Сообщение об изменении параметров слежения за частотами
<i>Параметры результата</i>	нет
<i>Пример ответа</i>	{ "channel": "signalChannel", "command": "freqTrackingParametersChanged", "valueType": "", "value": "", "status": "ok", "error": "" }

displayedSnrTypeChanged	
<i>Описание</i>	Сообщение об изменении типа отображения ОСШ
<i>Параметры результата</i>	нет
<i>Пример ответа</i>	{ "channel": "signalChannel", "command": "displayedSnrTypeChanged", "valueType": "", "value": "", "status": "ok", "error": "" }

progressImitHwChanged	
<i>Описание</i>	Сообщение об изменении переданных имитатором данных
<i>Параметры результата</i>	(double) – часто файла, переданная в имитатор (1.0 = 100%)
<i>Пример ответа</i>	{ "channel": "signalChannel", "command": "progressImitHwChanged", "valueType": "double", "value": "0.5", "status": "ok", "error": "" }

configurationChanged	
<i>Описание</i>	Сообщение об изменении параметров сигнала
<i>Параметры результата</i>	нет
<i>Пример ответа</i>	{ "channel": "signalChannel", "command": "configurationChanged", "valueType": "", "value": "", "status": "ok", "error": "" }

1.8. Описание типов параметров

(int) – целое число (4 байта)

(short) – целое число (2 байта)

(uint) – целое беззнаковое число

(int64) – 64-битное целое число

(double) – числа с плавающей запятой

(bool) – целое число, принимает значения true и false

(string) – строка

(base64) – двоичные данные закодированные по стандарту Base64

(DataFormat) – формат данных. Принимает значения:

- 0 – ADC (АЦП)
- 1 – IQ8 (8-разрядные выборки IQ)
- 2 – IQ16 (16-разрядные выборки IQ)
- 3 – DMD8 (8-разрядные выборки DMD)
- 4 – DMDPACK (демодулятор - упаковка)
- 5 – DECODER (данные декодера)
- 6 – IQ Clck (данные IQ для анализа тактовой)
- 7 – DMA3 (данные для построения векторной диаграммы)

(BufferSize) – размер буфера данных (байт). Принимает значения:

- 131072
- 65536
- 32768
- 16384
- 8192
- 4096
- 2048
- 1024
- 512
- 1048576 - максимальный размер буфера (подходит только для канала IQ, равен значению, возвращаемому с помощью *getIqDataSize*)

(DevType) – тип устройства. Принимает значения:

- 0x00 – неизвестное устройство
- 0x10 – имитатор
- 0x16 – M
- 0x18 – E
- 0x19 – M1
- 0x1C – E1
- 0x1D – E2
- 0x1E – E3
- 0x21 – E4
- 0xFF – демо устройство

(DecoderVersion) – версия декодера. Принимает значения:

- 0 – M1
- 1 – M2
- 2 – M3
- 3 – E4
- 5 – не определен

(SignalType) – тип сигнала. Принимает значения:

- 0 – ФМ2
- 1 – ФМ4
- 2 – ФМ4С
- 3 – ФМ8
- 4 – КАМ8
- 5 – КАМ16
- 6 – КАМ32
- 7 – КАМ64
- 8 – $\pi/2$ – ФМ2
- 9 – $\pi/2$ – ФМ4
- 10 – зарезервировано (недоступно)
- 11 – ПСП-2047
- 12 – зарезервировано (недоступно)
- 13 – ФМ2-135
- 14 – АФМ16
- 15 – АФМ32
- 16 – зарезервировано (недоступно)
- 17 – зарезервировано (недоступно)
- 18 – 8QAM модема PSM-500L
- 19 – 8QAM модема PD-60
- 20 – DVB-S2
- 21 – DVB-S2X
- 22 – Пользовательский
- 23 – DVB-S2X-Jp
- 24 – DVB-S2X-CDM
- 25 – DVB-S2X-NT

(SymbolRate) – виды скорости кодирования сигналов. Принимает значения:

- 0 – 0
- 1 – 1/2
- 2 – 2/3
- 3 – 3/4
- 4 – 4/5
- 5 – 5/6
- 6 – 7/8
- 7 – 8/9
- 8 – 9/10

(ScType) – тип кодирования сигнального созвездия. G - отношение внешнего радиуса к внутреннему для АФМ16/АФМ32. Принимает значения:

- 0 – неопределенный
- 1 – ФМ8 Intelsat
- 2 – ФМ8 Gray 1
- 3 – ФМ8 Gray 2
- 4 – ФМ8 Gray 3
- 5 – ФМ8 Natural
- 6 – АФМ16 G = 3.15
- 7 – АФМ16 G = 2.85
- 8 – АФМ16 G = 2.75
- 9 – АФМ16 G = 2.70
- 10 – АФМ16 G = 2.60
- 11 – АФМ16 G = 2.57
- 12 – АФМ32 G = 2.84
- 13 – АФМ32 G = 2.72
- 14 – АФМ32 G = 2.64
- 15 – АФМ32 G = 2.54
- 16 – АФМ32 G = 2.53

(MgcAction) – действие для ручного РУ. Принимает значения:

- 0 – увеличить
- 1 – уменьшить

(GC2Mode) – режим РУ2. Принимает значения:

- 0 – медленный
- 1 – нормальный
- 2 – быстрый
- 3 – пульсирующий
- 4 - пользовательский

(EEPROMData) – объект данных считанных из EEPROM:

```
{
  "data": "rDe9PUoIcQI9N1s96Ai9AX83UT5DCPwBHjcVPy4Iw==",
  // (base64) данные
  "status": 0
  // (int) статус выполнения чтения данных
}
```

(DmdDataExParam) – объект параметров данных канала DMD:

```
{
  "dmdDataReverseBitsConstelD7": false,
  // (bool) изменяет упаковку бит в сигнальном созвездии на обратную (ранний – младший,
  на ранний - старший) в режиме записи "Демодулятор-упаковка"
  "dmdDataReverseBytesD6": false,
  // (bool) изменяет упаковку бит в байты на обратную (ранний – младший, на ранний -
  старший)
```

```

"dmdDataUserPackEnable": false,
  // (bool) упаковка канала DMD
"dmdDataUserPackFile": "",
  // (string) файл пользовательской упановки
"dmdDataUserPackMode": 0
  // (int) режим упаковки
}

```

(SyncState) – флаг синхронизации. Принимает значения:

- 0 – NO (синхронизация отсутствует / лицензия не введена)
- 1 – YES (есть синхронизация / лицензия введена)
- 2 – BAD (есть синхронизация, но сигнал содержит ошибки / лицензия введена неверно)
- 3 – NA (модуль обработки данных не загружен или недоступен / лицензия не требуется)

(BoardStatus) – объект статуса платы:

```

{
  "carrierSync": 0
  // (SyncState) синхронизация несущей частоты
  "clockSync": 0,
  // (SyncState) синхронизация тактовой частоты
  "decLicenseState": 0,
  // (SyncState) состояние ввода лицензии декодера
  "decOverheat": true,
  // (bool) флаг перегрева микросхемы декодера
  "decRS_0": 0,
  // (SyncState) статус синхронизации декодера Рида-Соломона
  "decRS_1": 0,
  // (SyncState) статус синхронизации декодера Рида-Соломона
  "decSync": 0,
  // (SyncState) синхронизация декодера
  "demLicenseState": 0,
  // (SyncState) состояние ввода лицензии демодулятора
  "demOverheat": true,
  // (bool) флаг перегрева микросхемы демодулятора
  "dmxTop0": 0,
  // (SyncState) синхронизация демультиплексора
  "dmxTop1": 0,
  // (SyncState) синхронизация демультиплексора
  "gc1InRange": 0,
  // (SyncState) статус АРУ1 (false - перегрузка)
  "errCode": 0,
  // (int) статус выполнения чтения данных с платы
  "ready": true
  // (bool) флаг корректности инициализации данных объекта
}

```

(BoardValues) – объект значений параметров платы:

```

{
  "ber": 0,
  // (double) вероятность битовой ошибки на выходе декодера
  "berIsSupported": false,
  // (bool) true – значение ber инициализировано и корректно
}

```

```

"carrier": 2.1219957904712067e-314,
  // (double) считанное значение несущей частоты
"clock": null,
  // (double) считанное значение тактовой частоты
"dvbs2Stat": "CAAAAAAAAAAIAPtyAgAAAAgABAwIAAAAVAAAAAAAAABoB24QAA
AAAAAAAAAAAAAAAAAAhuEAAAAAAAAAAAAAAAAAAQAAAAAAAAAgduEAgAAA
BrB24QAAAAAAAAgAAAAAAAAAAAAAAAAAAAAAAAAABJXVmAAAAAAkABAYAAAAAQM
FuEAAAAAA=",
  // (base64) статистика пакетов DVB-S2
"dvbs2StatCount": 0,
  // (int) размер таблицы статистики DVB-S2
"dvbs2StatReady": false,
  // (bool) таблица статистики пакетов DVB-S2 инициализирована и корректна
"eccStatus": {
  "isCCMMMode": true,
    // (bool)
  "isSISMode": true,
    // (bool)
  "isi": 36700160,
    // (int)
  "issy": true,
    // (bool)
  "mode": "0",
    // (int) DVBS2ECC_MODE принимает значения:
    // 0 - GENERIC_PACKETIZED
    // 1 - GENERIC_CONTINUOUS
    // 2 - MODE_RESERVED
    // 3 - MODE_TRANSPORT_STREAM
  "npd": false,
    // (bool)
  "rollOff": "0"
    // (int) DVBS2ECC_ROLLOFF принимает значения:
    // 0 - 035
    // 1 - 025
    // 2 - 020
    // 3 - RESERVED
  },
"eccStatusModcode": "",
  // (string)
"eccStatusReady": true,
  // (bool) параметры eccStatus инициализированы и корректны
"gc1Hard": 8.4927584940968277e-315,
  // (double)
"gc1PanoramaLValue": 1.3613081647942701e-315,
  // (double)
"gc1Soft": 8.4584305906942373e-315,
  // (double)
"gc2": 0,
  // (double)
"snrIsSupported": false,
  // (bool) считывание ОСШ поддерживается на данном устройстве и выполнено без ошибок
"snr": 8.4584412427495616e-315,
  // (double) значение ОСШ

```

```

"errCode": 0,
  // (int) статус выполнения чтения данных с платы
"ready": true
  // (bool) флаг корректности инициализации данных объекта
}

```

(BoardTemperature) – объект температурных показателей платы:

```

{
  "ahaDecTemperature": 5.7311614917584599e-322,
    // (double) температура микросхемы АНА4541 декодера (если установлена)
  "decOverheat": false,
    // (bool) флаг перегрева микросхемы декодера
  "decTemperatureReady": true,
    // (bool) температурные параметры декодера инициализированы и корректны
  "demOverheat": false,
    // (bool) флаг перегрева микросхемы демодулятора
  "eccDecTemperature": 0,
    // (double) температура микросхемы ECC3100 декодера (если установлена)
  "fpgaDecTemperature": 1.0037516711414099e-317,
    // (double) температура ПЛИС декодера
  "fpgaDemTemperature": 0,
    // (double) температура ПЛИС демодулятора
  "errCode": 0,
    // (int) статус выполнения чтения данных с платы
  "ready": false
    // (bool) флаг корректности инициализации данных объекта
}

```

(SnrCoeff) – объект коэффициентов для программного расчета ОСШ:

```

{
  "cod1": 1,
    // (double)
  "cod2": 1,
    // (double)
  "mod1": 1,
    // (double)
  "errCode": 0,
    // (int) статус выполнения чтения данных с платы
  "ready": true
    // (bool) флаг корректности инициализации данных объекта
}

```

(DESCRAMBLER) – объект параметров дескремблера:

```

{
  "Type": "1",
    // (int) тип, принимает значения:
    // 0 – зарезервировано
    // 1 – V.35
    // 2 – 3.20C
    // 3 - нестандартный

```

```

"V35Type": "0",
  // (int) тип v.35, принимает значения:
  // 0 – тип IESS
  // 1 – обычный
"Polynome": "0"
  // (int) полином для нестандартного дескремблера
}

```

(Modulation) – вид модуляции. Принимает значения:

- 0 – ФМ2
- 1 – ФМ4
- 2 – ФМ4С
- 3 – ФМ8
- 4 – КАМ8
- 5 – КАМ16
- 6 – КАМ32
- 7 – КАМ64
- 8 – $\pi/2$ – ФМ2
- 9 – $\pi/2$ – ФМ4
- 10 – ФМ2-135
- 11 – АФМ16
- 12 – АФМ32
- 13 – КАМ128
- 14 – КАМ256
- 15 – 8QAM модема PSM-500L
- 16 – 8QAM модема PD-60

(DecoderConfiguration) – объект конфигурации декодера:

```

{
  "DescramblerParam": <DESCRAMBLER, см. выше>,
  // (DESCRAMBLER) – объект параметров дескремблера
  "channelOrder": false,
  // (bool) порядок каналов
  "decRsCParam": {
    "BlockLen": 126,
    // (int) размер блока
    "StrCount": 4,
    // (int) кол-во строк
    "Syncs": 50
    // (int) сдвиг строк
  },
  "decRsDvbParam": {
    "Scrambler": "1",
    // (int) аддитивный скремблер вкл/выкл (0 – off, 1 – on)
    "Type": "0"
    // (int) тип RS DVB, принимает значения:
    // 0 – 204_188
    // 1 – 160_146
  },
  "decRsEnable": false,

```

```

// (bool)
"decRsIessParam": {
    "BlockLen": 225,
        // (int) размер блока
    "InfCount": 205,
        // (int) кол-во информационных символов
    "Scrambler": "0",
        // (int) аддитивный скремблер вкл/выкл (0 – off, 1 – on)
    "StrCount": 4,
        // (int) кол-во строк
    "StrShift": 112
        // (int) сдвиг строк
    },
"decRsMode": 0,
    // (int) режим, принимает значения:
    // 0 – IESS
    // 1 – C
    // 2 – DVB
"decoderEnable": false,
    // (bool) состояние декодера (вкл/выкл)
"descramblerEnable": false,
    // (bool) состояние дескремблера (вкл/выкл)
"descramblerRsEnable": false,
    // (bool) состояние дескремблера RS (вкл/выкл)
"descramblerRsParam": <DESCRAMBLER, см. выше>,
    // (DESCRAMBLER) – объект параметров дескремблера
"diffDecoderEnable": "0",
    // (int) диф.декодер вкл/выкл (0 – off, 1 – on)
"diffDecoderRsEnable": "0",
    // (int) диф.декодер RS вкл/выкл (0 – off, 1 – on)
"dmxAddEnable": false,
    // (bool)
"dmxAddParam": {
    "EDMACDescrambler": "0",
        // (int) дескремблер вкл/выкл (0 – off, 1 – on)
    "EDMACPeriod": 1008,
        // (int) период
    "IBSDescrambler": "1",
        // (int) дескремблер вкл/выкл (0 – off, 1 – on)
    "IBSFrame": 256,
        // (int) кадр
    "IBSPeriod": 512,
        // (int) период
    "IDRSpeed": "0",
        // (int) скорость IDR, принимает значения:
        // 0 – 1544
        // 1 – 2048
        // 2 – 6144
        // 3 – 8448
    "Inversion": "0",
        // (int) инверсия вкл/выкл (0 – off, 1 – on)
    "K2944Frame": 64,
        // (int) кадр

```

```

"K2944SuperFrame": 2,
  // (int) супер кадр
"Type": "3"
  // (int) тип демультимплексора дополнительного уровня, принимает значения:
  // 0 – выключен
  // 1- IBS
  // 2 – IDR
  // 3 – EDMAC
  // 4 – K2944
  // 5 – K128
},
"dmxEnable": false,
  // (bool)
"dmxqam8Param": {
  "BlockLen": "0",
  "ClkF": 0,
  // (double) текущая тактовая в кГц
  "ColCount": "0",
  // (int) кол-во столбцов
  "Dintrlv": "0",
  // (int) депережежитель вкл/выкл (0 – off, 1 – on)
  "InfBlockLen": "0",
  // (int) длина информационного блока
  "Modulation": "0",
  // (int) (Modulation) текущий вид модуляции
  "ROrder": "0",
  // (int) порядок чтения вкл/выкл (0 – прямой, 1 – обратный)
  "RowCount": "0",
  // (int) кол-во строк
  "ScrPoly": "0",
  // (int) аддитивный скремблер полином
  "ScrSet": "0",
  // (int) аддитивный скремблер установка
  "UW1": "0",
  // (int) уникальное слово [31..0]
  "UW2": "0",
  // (int) уникальное слово [63..32]
  "UW3": "0",
  // (int) уникальное слово [95..64]
  "UWLength": 0
  // (int) длина уникального слова
},
"dvbs2ccmParam": {
  "bbHeader": false,
  // (bool)
  "multichannel": false,
  // (bool)
  "syncDelete": false
  // (bool)
},
"dvbs2s2xParam": {
  "adaptiveCorrector": 0,
  // (int)

```

```

"bbHeader": false,
  // (bool)
"df1": false,
  // (bool)
"filterType": 0,
  // (int)
"modCode": false,
  // (bool)
"multichannel": false,
  // (bool)
"pvSize": 0,
  // (int)
"spectrInversion": false,
  // (bool)
"syncDelete": false,
  // (bool)
"upl": false
  // (bool)
},
"fildpcParam": {
  "FLMode": "0",
  // (int) режим, принимает значения:
  // 0 – LowLatency
  // 1 – Balanced
  // 2 – LowBER
  "FLScrambler": 0,
  // (int) вкл/выкл (0 – off, 1 – on)
  "FLSpeed": "0"
  // (int) скорость, принимает значения:
  // 0 – 0.499
  // 1 – 0.532
  // 2 – 0.639
  // 3 – 0.710
  // 4 – 0.726
  // 5 – 0.778
  // 6 – 0.798
  // 7 – 0.828
  // 8 – 0.851
},
"fxldpcParam": {
  "FXScrambler": "0",
  // (int) вкл/выкл (0 – off, 1 – on)
  "FXSize": "0",
  // (int) размер, принимает значения:
  // 0 – 256
  // 1 – 512
  // 2 – 1K
  // 3 – 2K
  // 4 – 4K
  // 5 – 8K
  // 6 – 16K
  "FXSpeed": "0"
  // (int) скорость, принимает значения:

```



```
// 0 – 1/2
// 1 – 2/3
// 2 – 3/4
// 3 – 14/17
// 4 – 7/8
// 5 – 10/11
// 6 – 16/17
},
"invertorEnable": "0",
// (int) инвертор вкл/выкл (0 – off, 1 – on)
"invertorRsEnable": "0",
// (int) инвертор RS вкл/выкл (0 – off, 1 – on)
"ldpcCodeRate": "0",
// (int) кодовая скорость, принимает значения:
// 0 – 1/2
// 1 – 3/4
// 2 – 7/8
// 3 – 2/3
// 4 – 5/6
"licenseKeyFLdpc": "",
// (string) лицензионный ключ декодера FL LDPC
"licenseKeyFxDpc": "",
// (string) лицензионный ключ декодера FX LDPC
"licenseKeyLdpc": "",
// (string) лицензионный ключ декодера LDPC
"licenseKeyTcc": "",
// (string) лицензионный ключ декодера TCC SB
"licenseKeyVfLdpc": "",
// (string) лицензионный ключ декодера VF LDPC
"mode": 0,
// (int) текущий режим декодера, принимает значения:
// 0 – NSK K7
// 1 – NSK K7 S32
// 2 – NSK K8
// 3 – SSK
// 4 – TPC
// 5 – SYNCHR
// 6 – DVBS2
// 7 – LDPC
// 8 – TCCSB
// 9 – VFLDPC
// 10 – FLLDPC
// 11 - FXLDPC
"nskk7Mode": 0,
// (int) режим декодера NSK K7, принимает значения:
// 0 – 1/2
// 1 – 3/4
// 2 – 7/8
// 3 – 2/3 DVB
// 4 – 3/4 DVB
// 5 – 5/6 DVB
// 6 – 7/8 DVB
// 7 – 2/3 TCM IESS
```

```

// 8 –2/3 TCM DSNG
// 9 –5/6 TCM DSNG
"nskk7s32Param": {
    "DMXEnable": false,
        // (bool)
    "frameLength": 1,
        // (int)
    "mode": 0,
        // (int) режим, принимает значения:
        // 0 – 1/2
        // 1 – 3/4
        // 2 – 7/8
    "pspFile": "",
        // (string)
    "sync": 1
        // (int)
},
"spectrInversion": false,
    // (bool) инверсия спектра
"sskParam": {
    "mode": 0,
        // (int) режим декодера Sequential, принимает значения:
        // 0 – 1/2 K36
        // 1 – 3/4 K63
        // 2 – 7/8 K89
        // 3 – 1/2 K41
        // 4 – 3/4 K57
    "puncturing": 0
        // (int) выкалывание (1- вкл, 0- выкл)
},
"tccParam": {
    "codeRate": 0,
        // (int)
    "interleaver": false,
        // (bool) включение деперемежителя
    "threshold": 0
        // (int)
},
"tpcDatumParam": {
    "Modulation": "0",
        // (int) (Modulation) текущий вид модуляции
    "TPCDatumMode": "0",
        // (int) режим, принимает значения:
        // 0 – ADVANCED 4K 1/2
        // 1 – ADVANCED 4K 3/4
        // 2 – ADVANCED 4K 7/8
        // 3 – ADVANCED 4K 0.950
        // 4 – ADVANCED 16K 0.453
        // 5 – ADVANCED 16K 1/2
        // 6 – ADVANCED 16K 3/4
        // 7 – ADVANCED 16K 7/8
        // 8 – ADVANCED 16K 0.922
        // 9 – M5_LEGACY 3/4

```

```

// 10 – M5_LEGACY 7/8
// 11 – M5_SHORT 3/4
// 12 – M5_SHORT 7/8
// 13 – M5_FULL 3/4
"descramblerEn": false
// (bool) включение аддитивного дескремблера
},
"trcHamParam": {
  "CheckGain": 0,
  // (int) порог синхронизации при проверке
  "ClkF": -6.2774385622041925e+66,
  // (double) текущая тактовая в кГц
  "DelLastBits": 0,
  // (int) удаление последних бит
  "Dintrlv": "0",
  // (int) деперемежитель вкл/выкл (0 – off, 1 – on)
  "Dopos": "0",
  // (int) дополнительная четность вкл/выкл (0 – off, 1 – on)
  "FindGain": 0,
  // (int) порог синхронизации при поиске
  "Modulation": "0",
  // (int) (Modulation) текущий вид модуляции
  "NumBlocks": 0,
  // (int) кол-во блоков
  "Reduce1": 0,
  // (int) сокращение 1й оси
  "Reduce2": 0,
  // (int) сокращение 1й плоскости
  "ScrPoly": "0",
  // (int) аддитивный скремблер полином
  "ScrSet": "0",
  // (int) аддитивный скремблер установка
  "Scrsw": "0",
  // (int) включение аддитивного скремблера (порядок: 0 – перед, 1 – после)
  "Scrsync": "0",
  // (int) синхронизация скремблера (0 – кадр, 1 – суперкадр)
  "UW1": "0",
  // (int) уникальное слово [31..0]
  "UW2": "0",
  // (int) уникальное слово [63..32]
  "UW3": "0",
  // (int) уникальное слово [95..64]
  "UWDel": "0",
  // (int) удаление уникального слова (порядок: 0 – перед, 1 – после)
  "UWLength": 0,
  // (int) длина уникального слова
  "kX": 57,
  // (int) ПУ код
  "kY": 39,
  // (int) ПУ код
  "kZ": 0,
  // (int) ПУ код
  "nX": 64,

```

```

        // (int) ПУ код
        "nY": 46,
        // (int) ПУ код
        "nZ": 0
        // (int) ПУ код
    },
    "tpcMode": "0",
    // (int) режим ТРС, принимает значения:
    // 0 – Hamming
    // 1 – N
    // 2 - Standard
    // 3 – Datum (только М3)
    // 4 – Paradise (только М3)
    // 5 – Other
    "tpcNParam": {
        "ClkF": -6.2774385622041925e+66,
        // (double) текущая тактовая в кГц
        "Modulation": "0",
        // (int) \(Modulation\) текущий вид модуляции
        "NumBlocks": 0,
        // (int) кол-во блоков
        "TPCNMode": "2",
        // (int) режим ТРС*N, принимает значения:
        // 0 – (128, 120) x (128, 120)
        // 1 – (64, 57) x (64, 57)
        // 2 – (32, 26) x (32, 26) x (4, 3)
        // 3 – (32, 26) x (32, 21)
    },
    "tpcStdParam": {
        "ClkF": -6.2774385622041925e+66,
        // (double) текущая тактовая в кГц
        "Modulation": "0",
        // (int) \(Modulation\) текущий вид модуляции
        "TPCSTDMode": "2",
        // (int) режим ТРС Standard, принимает значения:
        // 0 – 5/16 (М3)
        // 1 – 21/44 (М3)
        // 2 – 3/4 (М1, М2, М3) - то же, что и (64, 57) x (46, 39)
        // 3 – 7/8 (М1, М2, М3) - то же, что и (128, 120) x (128, 120)
        // 4 – 0.95 (М3)
        "descramblerEn": false
        // (bool) включение аддитивного дескремблера
    },
    "tpcOtherParam": {
        "ClkF": -6.2774385622041925e+66,
        // (double) текущая тактовая в кГц
        "Modulation": "0",
        // (int) \(Modulation\) текущий вид модуляции
        "TPCOtherMode": "0"
        // (int) режим ТРС, принимает значения:
        // 0 – 64_57_32_26_x8_ADD
    },
    "tpcParadiseParam": {

```

```

    "Modulation": "0",
    // (int) (Modulation) текущий вид модуляции
    "TPCParadiseMode": "0",
    // (int) режим TPC Paradise, принимает значения:
    // 0 – 0.477
    // 1 – 0.493
    // 2 – 0.666
    // 3 – 0.789
    // 4 – 0.875
    // 5 – 0.93
    "descramblerEn": false
    // (bool) включение аддитивного дескремблера
  },
  "vfldpcType": "1"
  // (int) параметры VFLDPC, принимает значения:
  // 0 – BPSK 0.488
  // 1 – QPSK 0.533
  // 2 – QPSK 0.631
  // 3 – QPSK 0.706
  // 4 – QPSK 0.803
  // 5 – QAM8 0.576
  // 6 – QAM8 0.642
  // 7 – QAM8 0.711
  // 8 – QAM8 0.780
  // 9 – QAM16 0.644
  // 10 – QAM16 0.731
  // 11 – QAM16 0.780
  // 12 – QAM16 0.829
  // 13 – QAM16 0.853
  // 14 – BPSK 0.493
  // 15 – QPSK 0.493
  // 16 – QPSK 0.654
  // 17 – QPSK 0.734
}

```

(DemodulatorConfiguration) – объект конфигурации демодулятора:

```

{
  "adCorrEnable": 0,
  // (int) режим адаптивного корректора
  "adCorrSpeed": 0,
  // (int) скорость адаптивного корректора
  "afcEnable": false,
  // (bool) включение автоматической подстройки частоты
  "carrier": 1000000000,
  // (double) несущая частота (Гц)
  "carrierTracking": true,
  // (bool) слежение за несущей частотой
  "clock": 1000000,
  // (double) тактовая частота
  "clockInversion": false,
  // (bool) инверсия тактовой частоты
  "clockTracking": true,

```

```

// (bool) слежение за тактовой частотой
"constellationType": "0",
// (string) (ScType) – тип кодирования сигнального созвездия (см. выше)
"filterType": 0,
// (int)
"gc2Limit": 4032,
// (int)
"gc2Mode": "1",
// (string) (GC2Mode) – режим РУ2 (см. выше)
"gc2UserCoeff": 1,
// (double)
"gc2UserTimeHigh": 4,
// (int)
"gc2UserTimeLow": 4,
// (int)
"inputSignalMode": 0,
// (int) (InputBoardMode) режим работы демодулятора (0: L-конвертор, 1: ПЧ-140)
"licenseKeyAdaptiveCorrector": "",
// (string) лицензионный ключ адаптивного корректора
"licenseKeyB2347": "",
// (string) лицензионный ключ расширенного частотного диапазона 23-47 МГц
"licenseKeyB4760": "",
// (string) лицензионный ключ расширенного частотного диапазона 47-60 МГц
"licenseKeyDvbS2": "",
// (string) лицензионный ключ режима DVB-S2
"licenseKeyDvbS2x": "",
// (string) лицензионный ключ режима DVB-S2X
"mgc1Enable": false,
// (bool) включение ручного управления схемы РУ1
"mgc2Enable": false,
// (bool) включение ручного управления схемы РУ2
"pllBand": 2,
// (int) полоса ФАПЧ (0 - 1/50, 1 - 1/100, 2 - 1/200, 3 - 1/400)
"reference": 0,
// (int) тип опорного генератора
"signalType": "1",
// (int) (SignalType) – тип сигнала (см. выше)
"snrType": 0,
// (int) тип ОСШ (0 - Ps/Pn, 1 - Eb/No)
"symbolRate": "3"
// (int) (SymbolRate) – виды скорости кодирования сигналов (см. выше)
}

```

(DeviceConfiguration) – объект конфигурации устройства:

```

{
  "dec": <DecoderConfiguration см. выше>,
  // (DecoderConfiguration) – объект конфигурации декодера
  "dem": <DemodulatorConfiguration см. выше>,
  // (DemodulatorConfiguration) – объект конфигурации демодулятора
  "dmdExParam": <DmdDataExParam см. выше>
  // (DmdDataExParam) объект параметров данных канала DMD
}

```

(DeviceBaseParameters) – объект основных параметров устройства:

```
{
  "decType": "0",
    // (int) (DecoderVersion) – версия декодера (см. выше)
  "fullName": "",
    // (string) имя устройства в системе
  "interfaceVer": "",
    // (string) версия интерфейса модуля ОСПЧ
  "isReady": true,
    // (bool) статус устройства
  "lcSerial": "",
    // (string) серийный номер L-конвертора
  "lcType": 2462392,
    // (int) тип L-конвертора
  "lcVer": 109054240,
    // (int) версия L-конвертора
  "moduleVersion": 0,
    // (int) версия модуля ОСПЧ
  "number": 17,
    // (int) номер устройства ОСПЧ в системе
  "serial": "",
    // (string) серийный номер устройства ОСПЧ
  "slot": 0,
    // (int) слот устройства ОСПЧ в системе
  "type": "c7a9380",
    // (int) (DevType) – тип устройства ОСПЧ (см. выше)
  "errCode": 0,
    // (int) статус выполнения чтения данных с платы
  "ready": false
    // (bool) флаг корректности инициализации данных объекта
}
```

(ISOCVRTLoading) – объект содержащий значение загрузки внутреннего буфера изохронного преобразователя:

```
{
  "loading": 209150624,
    // (int) загрузка буфера изохронного преобразователя
  "errCode": 0,
    // (int) статус выполнения чтения данных с платы
  "ready": false
    // (bool) флаг корректности инициализации данных объекта
}
```

(ContinuousDataSpeed) – объект скоростей обмена данными:

```
{
  "hwSpeed": 0,
    // (double) Hardware - скорость обмена данных по шине PCIe
  "swSpeed": 0,
    // (double) Software – вычисленная скорость поступления данных
}
```

```
"errCode": 0,  
  // (int) статус выполнения чтения данных с платы  
"ready": true  
  // (bool) флаг корректности инициализации данных объекта  
}
```

(HwImitParam) – объект управления аппаратным имитатором:

```
{  
  "filePath": 0,  
    // (string) путь к файлу имитационного сигнала  
  "digits": 0,  
    // (int) разрядность данных (0 – 8 бит; 1 – 16 бит)  
  "dataType": 0,  
    // (int) тип данных (0 – комплексные, 1 – вещественные)  
  "zeroPadding": false,  
    // (bool) заполнение нулями  
  "reverseBits": false,  
    // (bool) порядок бит  
  "reverseBytes": false,  
    // (bool) порядок байт  
  "reverseWords": false,  
    // (bool) Software – порядок слов  
  "errCode": 0,  
    // (int) статус выполнения чтения данных с платы  
  "ready": true  
    // (bool) флаг корректности инициализации данных объекта  
}
```