

Программный пакет управления

(от 25 марта 2015 года)

Оглавление

Введение	3
Состав пакета разработчика	3
Программный интерфейс между модулем управления и внешним программным обеспечением	4
Протоколы обмена устройств с «Модуль управления», способы подключения	6
Методы модуля опроса	8
Описание параметров основных процедур	10
Описание параметров ответа сервера на асинхронные и синхронные методы	11
Параметры метода SetSubscribe:	12
Параметры метода CloseScribe:	20
Параметры метода OnCloseScribe:	20
Настройка модуля управления на работу с RS интерфейсом с помощью метода SetConfigurationHwSrv	23
Параметры метода SetConfigurationHwSrv:	24
Параметры метода GetConfigurationhwServer:	27
Параметры метода GetDevice:	33
Параметры метода GetStateElements:	37
Параметры метода GetListDevice:	56
Параметры метода ControlObjects:	84
Параметры метода OnRsEvent:	96
Параметры метода OnNewDevice:	99
Параметры метода OnChangeStateElements:	104
Параметры метода OnInitiative:	105
Параметры метода TranslateRightsObjects:	125
Параметры метода TranslateStatusObjects:	129
Параметры метода TranslateRsEvent:	132
Параметры метода RsCommand:	133
Параметры метода GetListMethods:	152
Параметры метода GetListError:	162
Параметры метода GetListRsCommand:	171
Параметры метода GetDeviceListAsync:	171
Параметры метода GetDeviceList:	176
Параметры метода LoadConfData (LoadAccessData):	177
Параметры метода SynchronizeOneKey:	214
Параметры метода SynchronizeKeys:	215
Параметры метода GetKeyFullInformation:	218
Параметры метода GetPasswordListWithStatus:	225
Параметры метода ReadConfiguration:	228
Параметры метода ReadDeviceKeyList:	231
Параметры метода GetDeviceConfigurationStatus:	235
Параметры метода ControlAccess:	236
Параметры метода ReadKeyCodeFromReader:	240

Ответ на вызов ReadKeyCodeFromReader:	243
Параметры метода WriteConfToPKU:	244
Параметры метода ArmDisarmShleif:	248
Параметры метода ControlRelay:	252
Параметры метода DeleteKeys:	257
Параметры метода RefreshTablesData:	260
Конфигурирование СКУД и управления взятием/снятием ШС.....	262
Приложение А. Коды типов приборов, состояний и событий.	263

Введение

Специально для фирм разрабатывающих системы безопасности, заинтересованных в поддержке оборудования выпускаемого ЗАО НВП «Болид» был разработан программный пакет управления интегрированной системой безопасности «Орион». В состав данного пакета входит модуль-драйвер «Управления ИСО Орион», который предоставляет внешний интерфейс управления охранно-пожарной системой, системой контроля и управления доступом, а также системой пожаротушения, построенных на базе оборудования ЗАО НВП «Болид».

Контроллеры, управление которым поддерживает модуль управления, приведены в таблице 1 приложения А.

Модуль выполнен в виде Web-сервиса, а интерфейс управления создан по технологии XML-RPC. Модуль может быть запущен как Windows сервис, так и обыкновенное приложение. Сервис работает под управлением операционных систем MS Window 2000/XP/2003. Для проведения интеграции необходимы базовые знания основ технологии XML-RPC, при этом программный комплекс, взаимодействующий с модулем, может быть разработан на любом языке программирования, позволяющий реализовать данную технологию.. При удаленном сетевом взаимодействии с модулем, клиентские программы могут работать и не под Microsoft совместимыми операционными системами.

Состав пакета разработчика

CoreOrion.exe – «Модуль управления ИСО Орион», xml-rpc сервер управления оборудованием «Орион», ядро логики для АРМ «Орион Про» работающее с оборудованием. Ядро логики позволяет интегрировать «Модуль управления» в АРМ и увидеть работу его совместно с выпускаемым коммерческим программным обеспечением.

Settings.exe – программа настроек временных задержек для протоколов обмена с устройствами.

RegParser.exe – программа регистрации Microsoft XML Parser 4 SP 2, XML Parser находится в каталоге «..\DLLXML».

rpcToaster.exe – программа, демонстрирующая передачу данных по XML-RPC.

hwMonitor.exe – программа «Монитор модуля опроса», демонстрирующая работу с модулем управления.

xml-rpc-test.exe – программа, демонстрирующая работу взятия/снятия шлейфов и получения событий.

AccessControlDemo.exe – программа, демонстрирующая конфигурирование СКУД.

В каталоге «..\Programs\Sources\ClhServer» хранятся исходные коды для программы «Монитор модуля опроса» (hwMonitor.exe) для Borland Delphi 7.

В каталоге «..\Programs\Sources\Toaster» хранятся исходные коды для программы демонстрирующее работу xml-rpc (rpctoaster.exe) для Borland Delphi 7.

В каталоге «..\Doc» находится описание на пакет разработчика, инструкция установки компонентов XML-RPC, инструкция интеграции модуля в Орион Про и вопросы и ответы по модулю опроса.

В каталоге «..\Library» находятся реализации XML-RPC компонентов для наиболее популярных языков программирования, таких как: C++, .NET, Delphi. Дополнительные и более расширенные версии реализации XML-RPC для различных языков программирования можно скачать с сайта <http://www.xmlrpc.com/directory/1568/implementations>.

Программный интерфейс между модулем управления и внешним программным обеспечением

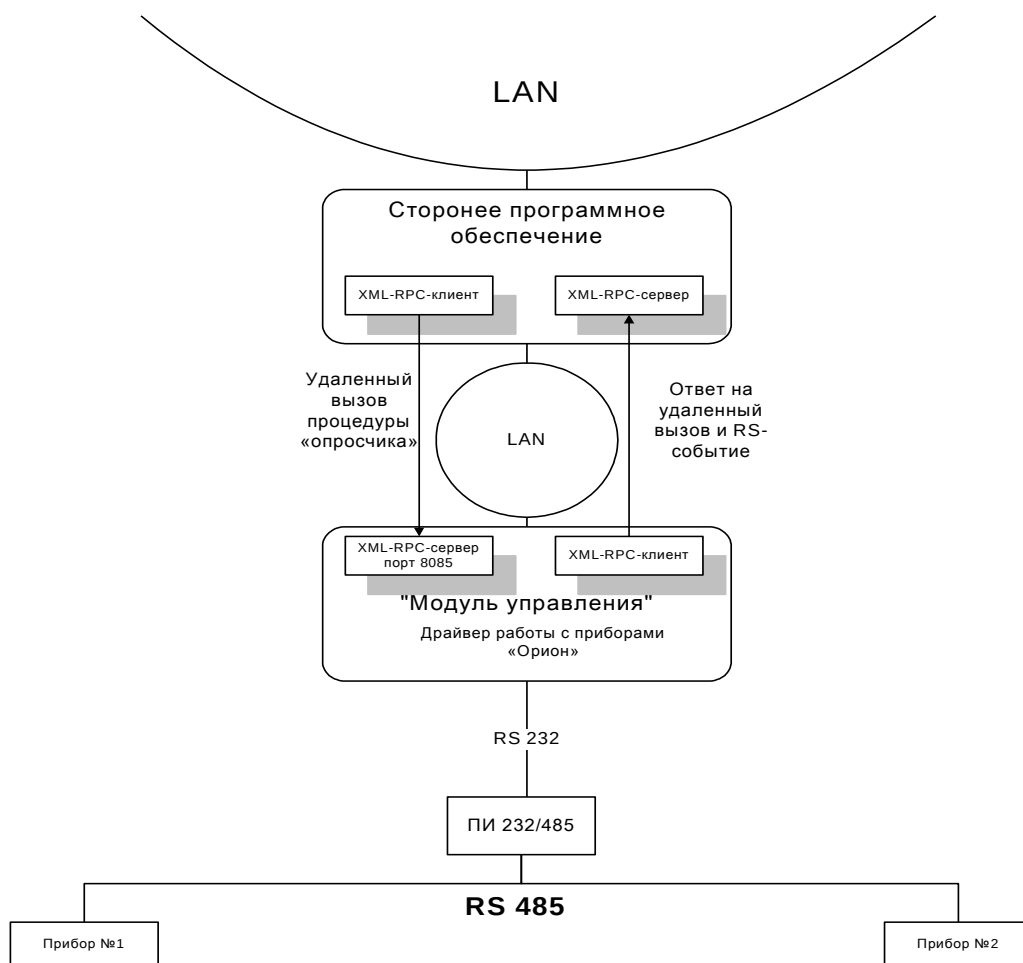
Межзадачное взаимодействие модуля управления и внешнего программного обеспечения осуществляется через протокол TCP/IP. При этом для возможности вызова удаленных RPC методов в модуле создается серверный TCP/IP сокет, а для возврата результатов на асинхронные входящие вызовы и для трансляции событий вычитанных модулем с RS интерфейса создается клиентский TCP/IP сокет.

Серверный сокет по умолчанию открывается на порту 8080, если необходимо указать другой порт, то необходимо установить в разделе реестра компьютера соответствующее значение для параметра XMLIPPORT в HKEY_LOCAL_MACHINE\SOFTWARE\BOLID\ORION_PRO\ORICORE.

Если внешнее программное обеспечение работает на том же компьютере, что и модуль то можно указывать как сетевой адрес компьютера, так и адрес заглушки «127.0.0.1», при удаленном взаимодействии необходимо указывать сетевой адрес компьютера.

Модуль можно зарегистрировать как Windows службу, для этого необходимо запустить программу CoreOrion.exe с ключом «-INSTALL», для деинсталляции необходимо запускать программу с ключом «-UNINSTALL».

Способ взаимодействия стороннего ПО с Модулем опроса выглядит следующим образом:



Как видно из схемы, клиент «Модуля управления» для вызова удаленных процедур должен создать клиентский XML – RPC сокет. Если клиент хочет получать данные о результатах выполнения асинхронных процедур или события считанные драйвером с RS интерфейса из приборов, то он должен создать XML-RPC сервер внутри себя и передать модулю управления параметры своего XML-RPC сервера – «подписать» свой сервер на получение данных от модуля управления.

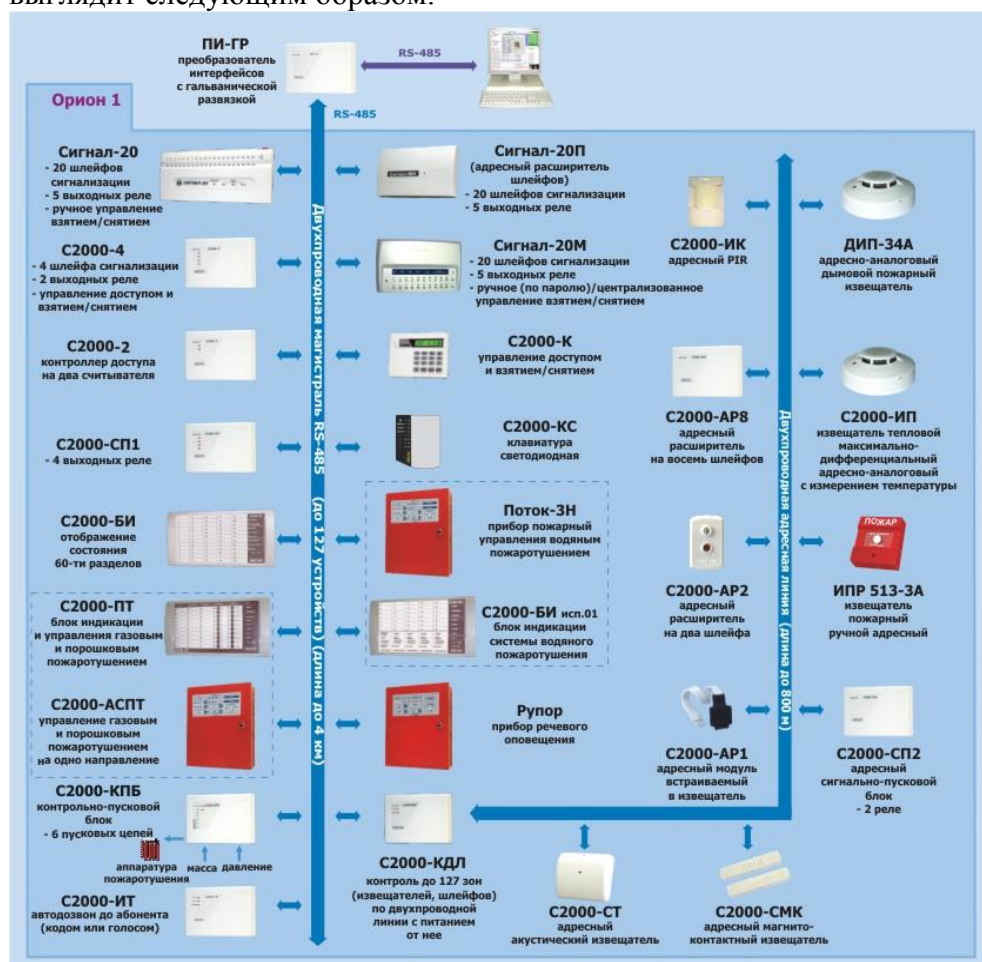
Протоколы обмена устройств с «Модуль управления», способы подключения

ЗАО НВП «Болид» было разработано два протокола обмена компьютера с устройствами:

1. Протокол обмена приборов с сетевым контроллером.

В качестве сетевого контролера может быть использован пульт С2000 или компьютер с установленным опрашивающим ПО.

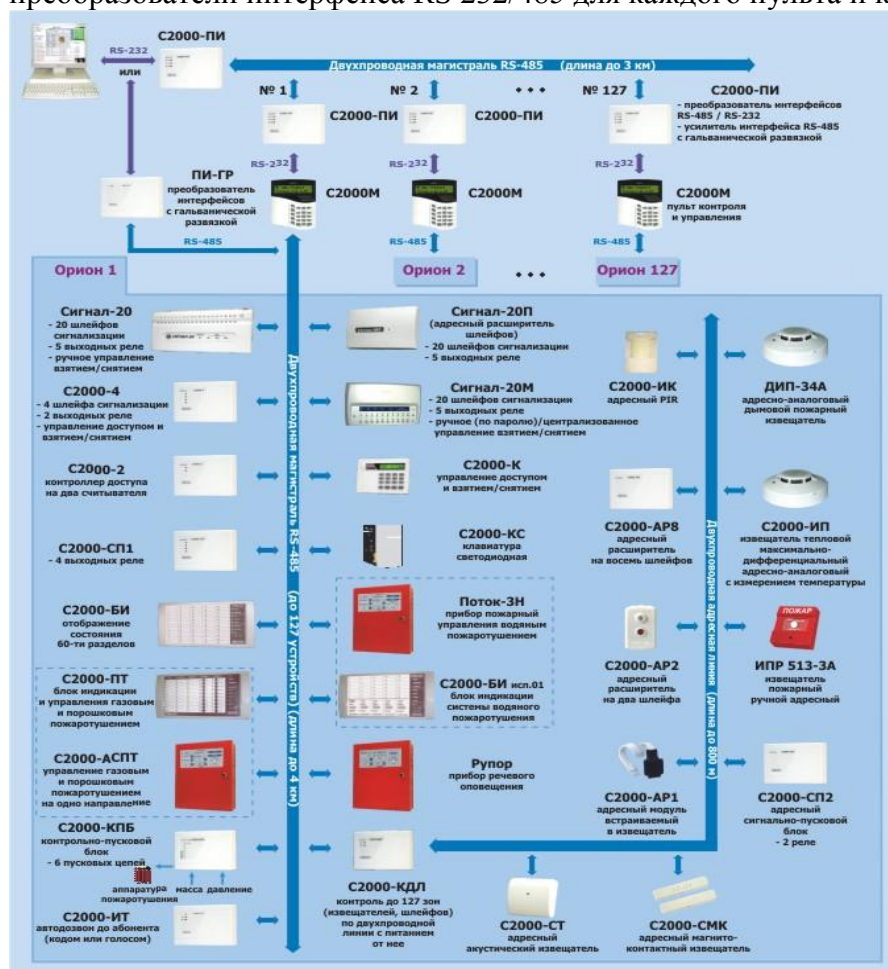
Так как информационный протокол передачи использует протокол передачи данных стандарта RS-485, для работы устройств необходимо использовать соответствующее оборудование позволяющее передавать данные в формате RS-485, такое как: преобразователи интерфейса RS232/485 ПИ-ГР или С2000-ПИ компании ЗАО НВП «Болид», TCC-80 или TCC-100 компании Moxa Technologies, ICP-DAS 7520 компании ICP CON, плату RS-485 серии CP компании Moxa Technologies. Схема подключения приборов к компьютеру выглядит следующим образом.



Основным недостатком данной схемы является невозможность автономной работы системы, при выходе из строя компьютера система полностью лишается централизованного

управления. По данной схеме к одному порту компьютера можно подключить до 127 устройств.

2. Протокол обмена с пультом С2000. В данном случае компьютер опрашивает только пульт С2000 (пульта), а С2000 работает с устройствами, подключенными к нему по RS-485. Команды, транслируемые от компьютеру к устройствам, посылаются пульту, а системный контроллер ретранслирует эти команды в свой внутренний RS-485 нужному устройству. При подключении к компьютеру одного пульта, возможно подключение напрямую на RS-232. При подключении нескольких пультов к компьютеру необходимо использовать преобразователи интерфейса RS 232/485 для каждого пульта и компьютера.



Преимуществом данной схемы, является автономность работы веток интерфейса подключенных к пультам С2000 от работоспособности компьютера. Недостатком является более низкая скорость исполнения команд. По данной схеме к одному порту можно подключить до 127 пультов.

Методы модуля опроса

Работа с модулем управления основывается на вызове удаленных процедур.

В таблице приведены все методы опубликованные в модуле управления.

Наименование	Описание	Тип	Ответ
SetSubscribe	Подпись клиента к модулю управления	Синхронный	Есть
OnRsEvent	Событие на RS интерфейсе от устройства	Событие	Нет
SetConfigurationHwSrv	Передать конфигурацию в модуль управления	Синхронный	Нет
GetListError	Получить список кодов ошибок и описание на них	Синхронный	Нет
GetConfigurationHwServer	Получить конфигурацию модуля управления	Синхронный	Нет
GetDevice	Узнать состояние и информацию о устройстве.	Асинхронный	Есть
GetStateElements	Запрос состояний элемента: зоны, считывателя, реле.	Асинхронный	Есть
RsCommand	Трансляция команд на устройства	Асинхронный	Есть
CloseScribe	Отписаться от клиента	Синхронный	Нет
OnChangeStateElements	Событие о новом состоянии элемента объекта	Событие	Нет
ReadKeyCodeFromReader	Считать код ключа со считывателя	Асинхронный	Есть
GetKeyListFromDevice	Получить список кодов ключей прибора	Синхронный	Есть
OnInitiative	Инициатива управления, для централизованного управления	Событие	Нет
TranslateRightsObjects	Передача на считыватель полномочий управления, списка объектов	Синхронный	Нет
TranslateStatusObjects	Передача на считыватель состояния раздела и других объектов	Синхронный	Нет

ControlObjects	Управление объектами	Синхронный	Нет
GetInfo	Запрос информации о Модуле управления	Синхронный	Есть
OnNewDevice	Событие об обнаружении на RS интерфейсе нового устройства	Событие	Нет
OnResultCommand	Событие, в котором передается ответ на команду (RSCommand)	Событие	Нет
OnCloseScribe	Отписать клиента от модуля управления с генерацией события о результате выполнения	Асинхронный	Есть
GetListDevice	Получить список устройств с которыми работает модуль и информацию по ним	Синхронный	Есть
GetListRsCommand	Получить шаблоны RS команд, метода RSCommand	Синхронный	Есть
TranslateRSEvent	Трансляция события на устройство	Синхронный	Есть
GetDeviceList	Получение списка устройств	Синхронный	Есть
GetDeviceListAsync	Получение списка устройств	Асинхронный	Есть
LoadAccessData	Загрузка конфигурации доступа в модуль управления	Синхронный	Нет
SynchronizeOneKey	Синхронизация одного ключа	Асинхронный	Есть
SynchronizeKeys	Синхронизация всего списка ключей	Асинхронный	Есть
GetKeyFullInformation	Получение полной информации о ключе	Асинхронный	Есть
GetPasswordListWithStatus	Получение информации о состоянии списка ключей	Асинхронный	Есть
ReadConfiguration	Чтение конфигурации из прибора	Асинхронный	Есть
ReadDeviceKeyList	Чтение кодов ключей из прибора	Асинхронный	Есть
GetDeviceConfigurationStatus	Информация о считанных конфигурациях	Синхронный	Есть
ControlAccess	Управление доступом	Асинхронный	Есть

ArmDisarmShleif	Взятие-снятие зоны (ШС) под охрану	Асинхронный	Есть
ControlRelay	Команда внешнего управления реле	Асинхронный	Есть
WriteConfToPKU	Запись конфигурации в пульт С2000	Асинхронный	Есть

Существует три типа вызовов (методов):

- синхронный, команда выполняется немедленно и в структуре RpcResult возвращает данные ответа. При вызове метода программа, выполняющая метод останавливается и ждет выполнения.

- асинхронный, модуль получает данные и «отпускает» соединение с клиентом. Команда ставится в очередь, и время исполнения зависит от количества команд в очереди. Ответ передается посредством вызова удаленного метода у клиента, все необходимые для этого данные передаются в параметрах «входящей» команды.

-событие, генерируется модулем у удаленного клиента в соответствии с параметрами подписи клиента.

Ниже будут описаны все основные методы и параметры методов модуля управления ИСО «Орион».

Описание параметров основных процедур

Все методы модуля управления, за исключением SetSubscribe исполняются модулем только при вызове «подписанных» клиентов, т.е. клиентов которым был выдан специальный код - жетон безопасности. При вызове всех методов, за исключением указанных, обязательно необходимо передавать в параметрах удаленной процедуры выданный жетон, при вызове метода сервер проверяет на соответствие выданного жетона и произведенного вызова клиента и в случае несоответствия выдает ответ, что клиент не «подписан» на выполнение метода:

```
<?xml version="1.0" encoding="Windows-1251" ?>
```

```
<methodResponse>
```

```
<fault>
```

```
<value>
```

```
<struct>
```

```
<member>
```

```
<name>faultCode</name>
```

```
<value><int>113</int></value>
```

```
</member>
```

```
<member>
```

```
<name>faultString</name>
```

```

    <value><string>Client has to be subscribed</string></value>

  </member>

</struct>

</value>

```

Поэтому начало работы с модулем управления должно начинаться с вызова метода **SetSubscribe** осуществляющего подпись клиента к серверу и с получения жетона безопасности.

Описание параметров ответа сервера на асинхронные и синхронные методы

Все ответы модуля управления на синхронные и асинхронные методы имеют стандартную структуру данных и отличаются лишь вложением данных в структуру **ResultData**.

Ответ на запрос может содержать два вида ответа:

1) ответ без ошибки с одним параметром типа Struct, в которой передаются следующие данные:

- **STRING RESULT** – строковая константа результата исполнения:
«METHOD IS EXECUTE» - метод исполнен;
- **STRUCT RESULTDATA** – структура возвращаемых данных может иметь произвольный состав параметров, зависящий от типа запроса.

2) ответ с ошибкой, сформированной стандартным образом с помощью тегов <fault>, в котором в качестве параметров передается структура со следующими значениями:

- **INT64 faultCode** – код ошибки;
- **STRING faultString** - описание ошибки.

Ответ первого типа может выглядеть следующим образом:

```

<?xml version="1.0" encoding="Windows-1251" ?>

<methodResponse>

  <params>

    <param>

      <value>
        <struct>
          <member>
            <name>RESULT</name>
            <value><string>METHOD IS EXECUTE</string></value>
          </member>
          <member>
            <name>RESULTDATA</name>
          </member>
        </struct>
      </value>
    </param>
  </params>
</methodResponse>

```

```

    </member>
  </struct>
</value>

</param>
</params>
</methodResponse>

```

Ответ «с ошибкой» может выглядеть так:

```

<?xml version="1.0" encoding="Windows-1251" ?>

<methodResponse>

  <fault>

    <value>

      <struct>

        <member>

          <name>faultCode</name>

          <value><int>126</int></value>

        </member>

        <member>

          <name>faultString</name>

          <value><string>HWServer is not configured</string></value>

        </member>

      </struct>

    </value>

  </fault>

</methodResponse>

```

Все коды ошибок и их описание можно узнать вызвав метод `GetListError()`, подробнее об этом методе ниже.

При прекращении работы с модулем управления клиенту необходимо отписать себя от сервиса, для этого существуют два метода `CloseScribe` и `OnCloseScribe`. При вызове этих методов в модуле удаляются данные и объекты, связанные с клиентом. В результате чего удаленные вызовы методов со «старым» жетоном безопасности будут не возможны. Метод `OnCloseScribe` отличается от `CloseScribe`, тем что модуль опроса после отписи формирует событие и делает удаленный вызов на стороне своего клиента.

Параметры метода `SetSubscribe`:

Содержит один параметр типа STRUCT(структура), в котором передаются следующие данные:

- **STRING LOGIN** – логин для подключения удаленного клиента, по умолчанию = «ADMINISTRATOR», логин для модуля опроса можно изменить указав необходимое значение для параметра LOGIN раздела реестра HKEY_LOCAL_MACHINE\SOFTWARE\BOLID\ORION_PRO\HWSERVER компьютера на котором работает модуль управления.
- **STRING PASSWORD** – пароль для Login, по умолчанию = ORION, пароль для модуля опроса можно изменить указав необходимое значение для параметра PASSWORD в разделе реестра HKEY_LOCAL_MACHINE\SOFTWARE\BOLID\ORION_PRO\HWSERVER.
- **INT64 SCRIBE** – битовая подпись клиента, с указанием действий, на которые необходимо получить разрешение.

Битовая маска подписи:

- 0 бит – на рассылку событий;
- 1 бит – на изменение состояний объектов;
- 4 бит – на управляющие команды;
- 5 бит – на конфигурирование модуля опроса;
- 6 бит – чтение и запись EEPROM прибора;
- 7 бит – запросы модулю управления;
- 8 бит – ответы на асинхронные запросы;
- 9 бит – на управляющие инициативы управления.

- **STRING IPSEVER**¹ – Ip адрес сервера, используется для получения ответов на асинхронные команды и при подписи на: рассылку событий, событие об изменении состояния объектов, обнаружения устройств и инициатив управления. Если метод SetSubscribe вызывается с данным параметром, то полученный в ответ жетон безопасности нельзя использовать для вызова методов модуля управления.
- **INT64 PORTSERVER** – порт сервера, используется для получения ответов на асинхронные команды и при подписи на: рассылку событий, события об изменении состояния объектов, обнаружения устройств и инициатив управления.
- **STRING SCRIBEPORTS** – ключевое константное слово подписи клиентом всех портов = «SCRIBEALLPORTS» - подписать все порты.
- **ARRAY PORTS** – список портов которые подписывает клиент:

Параметры **ARRAY PORTS** элементы массива представляют собой структуры данных:

- **STRUCT NUMBERPORT** - структура данных подписываемых портов;
 - **INT64 ADDRPORT** – номер порта;
 - **STRING SCRIBEDEVICES** – ключевое константное слово подписи всех клиентов = «SCRIBEALLDEVICES» – подписать все устройства.
- **ARRAY DEVICES** – список адресов устройств которые подписывает клиент для управления. Элементами списка ARRAY DEVICES являются структуры данных:

¹ **ИМЯ ПАРАМЕТРА** - нижнее подчеркивание говорит о том что параметр в методе может отсутствовать

- **INT64 ADDRDEVICE** – адрес устройства по RS 485;
- **INT64 ADDRPULT** – адрес пульта по RS 232;
- **INT64 ADDRPORT** – номер порта к которому подключен интерфейс.

В случае корректного запроса, модуль выдает ответ который содержит один параметр типа STRUCT), в котором передаются следующие данные:

- **STRING RESULT** – строковая константа - результат исполнения:
«METHOD IS EXECUTE» - метод исполнен;
«METHOD IS ERROR» - при выполнении метода возникла ошибка.
- **STRUCT RESULTDATA** – структура возвращаемых данных:
 - **STRING GUID** – жетон безопасности.

Пример: Клиенту необходимо подписаться на полное управление к модулю опроса, по пяти COM портам и по всем устройствам, которые подключены к ним. Кроме этого необходимо подписать сервер для получения событий.

Пример: Подпись клиента к модулю управления с запросом жетона безопасности:

```
<?xml version="1.0" encoding="Windows-1251" ?>
```

```
<methodCall>

  <methodName>SETSUBSCRIBE</methodName>

  <params>

    <param>

      <value>
        <struct>
          <member>
            <name>LOGIN</name>
            <value><string>ADMINISTRATOR</string></value>
          </member>
          <member>
            <name>PASSWORD</name>
            <value><string>ORION</string></value>
          </member>
          <member>
            <name>SCRIBE</name>
            <value><int>65535</int></value><!--подпись на все возможные действия с сервисом-->
          </member>
          <member>
            <name>PORTS</name>
            <value>
              <array>
                <data>
                  <value>
                    <struct>
                      <member>
                        <name>NUMBERPORT</name>
                        <value>
                          <struct>
                            <member>
                              <name>ADDRPORT</name>
                              <value><int>1</int></value>
                            </member>
                          </struct>
                        </value>
                      </struct>
                    </value>
                  </data>
                </array>
              </value>
            </member>
          </param>
        </value>
      </param>
    </params>
  </methodCall>
```

```

    <member>
      <name>SCRIBEDEVICES</name>
      <value><string>SCRIBEALLDEVICES</string></value>//подписываю все устройства которые будут работать по данному порту
    </member>
  </struct>
</value>

<value>
  <struct>
    <member>
      <name>NUMBERPORT</name>
    </member>
  </struct>
</value>

  <struct>
    <member>
      <name>ADDRPORT</name>
      <value><int>2</int></value>
    </member>
  </struct>
</value>

  </member>
  <member>
    <name>SCRIBEDEVICES</name>
    <value><string>SCRIBEALLDEVICES</string></value>
  </member>
</struct>
</value>

<value>
  <struct>
    <member>
      <name>NUMBERPORT</name>
    </member>
  </struct>
</value>

  <struct>
    <member>
      <name>ADDRPORT</name>
      <value><int>3</int></value>
    </member>
  </struct>
</value>

  </member>
  <member>
    <name>SCRIBEDEVICES</name>
    <value><string>SCRIBEALLDEVICES</string></value>
  </member>
</struct>
</value>

<value>
  <struct>
    <member>
      <name>NUMBERPORT</name>
    </member>
  </struct>
</value>

  <struct>
    <member>
      <name>ADDRPORT</name>
      <value><int>4</int></value>
    </member>
  </struct>
</value>

  </member>
  <member>
    <name>SCRIBEDEVICES</name>
    <value><string>SCRIBEALLDEVICES</string></value>
  </member>
</struct>

```



```

</value>

<value>
  <struct>
    <member>
      <name>NUMBERPORT</name>
</value>
<struct>
  <member>
    <name>ADDRPORT</name>
    <value><int>5</int></value>
  </member>
</struct>
</value>

  </member>
  <member>
    <name>SCRIBEDEVICES</name>
    <value><string>SCRIBEALLDEVICES</string></value>
  </member>
</struct>
</value>

<value>
  <struct>
    <member>
      <name>NUMBERPORT</name>
</value>
<struct>
  <member>
    <name>ADDRPORT</name>
    <value><int>6</int></value>
  </member>
</struct>
</value>

  </member>
  <member>
    <name>SCRIBEDEVICES</name>
    <value><string>SCRIBEALLDEVICES</string></value>
  </member>
</struct>
</value>

</data>
</array>
</value>

  </member>
</struct>
</value>

</param>

</params>

</methodCall>

Ответ модуля управления - выдача жетона безопасности:
<?xml version="1.0" encoding="Windows-1251" ?>

<methodResponse>

  <params>

    <param>

```

```

<value>
  <struct>
    <member>
      <name>RESULT</name>
      <value><string>METHOD IS EXECUTE</string></value>
    </member>
    <member>
      <name>RESULTDATA</name>
</value>
<struct>
  <member>
    <name>GUID</name>
    <value><string>{355DA871-1565-48DA-8257-F379277DEFBE}</string></value> //жетон безопасности, теперь все «входящие» запросы
    необходимо вести указывая жетон безопасности.
  </member>
</struct>
</value>

  </member>
</struct>
</value>

```

```

</param>

</params>

</methodResponse>

```

Подпись к модулю управления RPC сервера для того, чтобы получать события и асинхронные ответы на запросы.
 <?xml version="1.0" encoding="Windows-1251" ?>

```

<methodCall>

  <methodName>SETSUBSCRIBE</methodName>

  <params>

    <param>

<value>
  <struct>
    <member>
      <name>IPSERVER</name>
      <value><string>192.168.10.109</string></value> //адрес RPC сервера клиента для получения событий
    </member>
    <member>
      <name>PORTSERVER</name>
      <value><int>8090</int></value> //порт адрес RPC сервера клиента для получения событий
    </member>
    <member>
      <name>LOGIN</name>
      <value><string>ADMINISTRATOR</string></value>
    </member>
    <member>
      <name>PASSWORD</name>
      <value><string>ORION</string></value>
    </member>
    <member>
      <name>SCRIBE</name>
      <value><int>65535</int></value>
    </member>
    <member>
      <name>PORTS</name>
</value>
  <array>
    <data>
</value>

```

```

<struct>
  <member>
    <name>NUMBERPORT</name>
</member>
</struct>
</value>

</member>
<member>
  <name>SCRIBEDEVICES</name>
  <value><string>SCRIBEALLDEVICES</string></value>
</member>
</struct>
</value>

<value>
  <struct>
    <member>
      <name>NUMBERPORT</name>
</member>
</struct>
</value>

</member>
<member>
  <name>SCRIBEDEVICES</name>
  <value><string>SCRIBEALLDEVICES</string></value>
</member>
</struct>
</value>

<value>
  <struct>
    <member>
      <name>NUMBERPORT</name>
</member>
</struct>
</value>

</member>
<member>
  <name>SCRIBEDEVICES</name>
  <value><string>SCRIBEALLDEVICES</string></value>
</member>
</struct>
</value>

<value>
  <struct>
    <member>
      <name>NUMBERPORT</name>
</member>
</struct>
</value>

</member>
<member>
  <name>SCRIBEDEVICES</name>
  <value><string>SCRIBEALLDEVICES</string></value>
</member>
</struct>
</value>

<value>
  <struct>
    <member>
      <name>NUMBERPORT</name>
</member>
</struct>
</value>

```

```

    <member>
      <name>ADDRPORT</name>
      <value><int>4</int></value>
    </member>
  </struct>
</value>

  </member>
  <member>
    <name>SCRIBEDEVICES</name>
    <value><string>SCRIBEALLDEVICES</string></value>
  </member>
</struct>
</value>

<value>
  <struct>
    <member>
      <name>NUMBERPORT</name>
    </member>
  </struct>
  <struct>
    <member>
      <name>ADDRPORT</name>
      <value><int>5</int></value>
    </member>
  </struct>
</value>

  </member>
  <member>
    <name>SCRIBEDEVICES</name>
    <value><string>SCRIBEALLDEVICES</string></value>
  </member>
</struct>
</value>

<value>
  <struct>
    <member>
      <name>NUMBERPORT</name>
    </member>
  </struct>
  <struct>
    <member>
      <name>ADDRPORT</name>
      <value><int>6</int></value>
    </member>
  </struct>
</value>

  </member>
  <member>
    <name>SCRIBEDEVICES</name>
    <value><string>SCRIBEALLDEVICES</string></value>
  </member>
</struct>
</value>

</data>
</array>
</value>

  </member>
</struct>
</value>

</param>

```

```
</params>
</methodCall>
```

Клиенту, модуль управления выдает жетон безопасности, с данным жетоном модуль будет **высылать события на клиент**.

```
<?xml version="1.0" encoding="Windows-1251" ?>
```

```
<methodResponse>

  <params>

    <param>

      <value>
        <struct>
          <member>
            <name>RESULT</name>
            <value><string>METHOD IS EXECUTE</string></value>
          </member>
          <member>
            <name>RESULTDATA</name>
            <value>
              <struct>
                <member>
                  <name>GUID</name>
                  <value><string>{2929AF45-CE93-4E14-A80F-9DD46A99DC5A}</string></value> //жетон для RPC сервера клиента
                </member>
              </struct>
            </value>

            </member>
          </struct>
        </value>

      </param>

    </params>

  </methodResponse>
```

Параметры метода CloseScribe:

Содержит один параметр типа STRUCT (структура), в котором передаются следующие данные:

- **STRING GUID** – жетон безопасности;
- **STRING LOGIN** – логин для подключения удаленного клиента, по умолчанию = ADMINISTRATOR;
- **STRING PASSWORD** – пароль для Login, по умолчанию = ORION;
- **STRING IPSERVER** – Тср Ip адрес сервера, созданного в клиенте, который необходимо отписать от получения данных;
- **INT64 PORTSERVER** – порт сервера, созданного в клиенте, который необходимо отписать от получения данных.

Параметры метода OnCloseScribe:

Содержит один параметр типа STRUCT(структура), в котором передаются следующие данные:

- **STRING GUID** – жетон безопасности;
- **STRING LOGIN** – логин для подключения удаленного клиента, по умолчанию = ADMINISTRATOR;
- **STRING PASSWORD** – пароль для Login, по умолчанию = ORION;
- **STRING IPSEVER** – Tcp Ip адрес сервера, созданного в клиенте, который необходимо отписать от получения данных;
- **INT64 PORTSERVER** – порт сервера, созданного в клиенте, который необходимо отписать от получения данных;
- **STRING RESULTRETURNMETHOD** – имя RPC метода который необходимо вызвать когда будет сформирован ответ на команду и перед тем, как клиент будет отписан.

В ответ на запрос модуль выдает ответ об исполнении без указания возвращаемых данных RESULTDATA или формирует ответ об ошибке.

Примеры с использованием методов CloseScribe и OnCloseScribe:

1) С использованием метода CLOSESCRIBE.

```
<?xml version="1.0" encoding="Windows-1251" ?>
```

```
<methodCall>
```

```
  <methodName>CLOSESCRIBE</methodName>
```

```
  <params>
```

```
    <param>
```

```
      <value>
```

```
        <struct>
```

```
          <member>
```

```
            <name>LOGIN</name>
```

```
            <value><string>ADMINISTRATOR</string></value>
```

```
          </member>
```

```
          <member>
```

```
            <name>PASSWORD</name>
```

```
            <value><string>ORION</string></value>
```

```
          </member>
```

```
          <member>
```

```
            <name>GUID</name>
```

```
            <value><string>{9C6C5CFD-466F-4FED-9930-3C5DB8B7E842}</string></value>
```

```
          </member>
```

```
        </struct>
```

```
      </value>
```

```
    </param>
```

```
  </params>
```

```
</methodCall>
```

Ответ:

Response Client{or 10.08.2006 16:55:22}:<?xml version="1.0" encoding="Windows-1251" ?>

```
<methodResponse>

  <params>

    <param>

      <value>
        <struct>
          <member>
            <name>RESULT</name>
            <value><string>METHOD IS EXECUTE</string></value>
          </member>
        </struct>
      </value>

    </param>

  </params>

</methodResponse>
```

2) Отписывание клиента с генерацией события «ReturnOnClose».

<?xml version="1.0" encoding="Windows-1251" ?>

```
<methodCall>

  <methodName>ONCLOSESCRIBE</methodName>

  <params>

    <param>

      <value>
        <struct>
          <member>
            <name>IPSERVER</name>
            <value><string>192.168.10.109</string></value>
          </member>
          <member>
            <name>PORTSERVER</name>
            <value><int>8090</int></value>
          </member>
          <member>
            <name>LOGIN</name>
            <value><string>ADMINISTRATOR</string></value>
          </member>
          <member>
            <name>PASSWORD</name>
            <value><string>ORION</string></value>
          </member>
          <member>
            <name>GUID</name>
            <value><string>{6360CF1A-0BCA-420D-8576-FDBE6441670F}</string></value>
          </member>
          <member>
            <name>RESULTRETURNMETHOD</name>
            <value><string>ReturnOnClose</string></value>
          </member>
        </struct>
      </value>

    </param>
```

```
</params>
</methodCall>
```

Ответ:

```
<?xml version="1.0" encoding="Windows-1251" ?>

<methodResponse>

  <params>

    <param>

      <value>
        <struct>
          <member>
            <name>RESULT</name>
            <value><string>METHOD IS EXECUTE</string></value>
          </member>
        </struct>
      </value>

      </param>

    </params>

  </methodResponse>
```

Событие у клиента:

```
<?xml version="1.0" encoding="Windows-1251" ?>

<methodCall>

  <methodName>ReturnOnClose</methodName>

  <params>

    <param>

      <value>
        <struct>
          <member>
            <name>RESULT</name>
            <value><string>METHOD IS EXECUTE</string></value>
          </member>
        </struct>
      </value>

      </param>

    </params>

  </methodCall>
```

Настройка модуля управления на работу с RS интерфейсом с помощью метода SetConfigurationHwSrv

Для того чтобы модуль управления работал с приборами, подключенными по RS-485, в драйвер необходимо передать конфигурацию работы драйвера с портами, с преобразователем интерфейса, тип используемого протокола и т.д. Для этого в модуле опубликован метод `SetConfigurationHwSrv` с помощью которого можно передать все необходимые для работы данные.

Параметры метода `SetConfigurationHwSrv`:

Содержит один параметр типа `STRUCT`(структура), в котором передаются следующие данные:

- **STRING GUID** – жетон безопасности;
- **ARRAY PORTS** – содержит список портов, которые необходимо открыть в драйвере для работы с устройствами, элементами списка являются структуры данных (тип `STRUCT`), каждый элемент содержит следующие данные:
 - **STRING TYPEPORT** – константное строковое значение определяющее тип порта через который будет осуществляться обмен данными между устройствами:
«COM» – COM порт (RS 232) (если параметр не указан, то обмены ведутся через COM порт);
«USB» - USB порт;
«LAN» - через сетевую карту.
 - **INT64 NUMBERPORT** – номер порта;
 - **INT64 TYPEPROTOCOL** - константа определяющая тип информационного протокола обмена данными с устройствами:
 - 1- протокол обмена с устройствами, когда драйвер ведет обмен данными с устройствами «напрямую» - не через пульт;
 - 2- протокол обмена с пультом C2000, когда драйвер ведет обмены только с пультом C2000, а команды на устройства отправляются с помощью трансляционных команд, которые ретранслируются C2000 ПКУ.
 - **INT64 TYPECONVERTER** – константа определяющая тип преобразователя интерфейса, который был использован при подключении к компьютеру:

0-стандартный ПИ, при работе в таком режиме драйвер будет управлять приемо-передатчиком преобразователя с помощью сигналов DTR/RTS сигналов. Необходимо указывать данный тип при подключении устройств к компьютеру, с помощью изделия ПИ-ГР компании ЗАО НВП «Болид»;

1-преобразователь с автоматическим управлением приемо-передатчика, в данном случае сам преобразователь управляет этим сигналом, внешнее управление не требуется. Необходимо указывать данный тип при подключении приборов к компьютеру с помощью изделия C2000-ПИ компании ЗАО НВП «Болид».

По умолчанию тип преобразователя = 0;

- **INT64 PORTPRIORITY** – константа определяющая приоритет работы с портом, соответствует приоритету работы потока(thread) созданного для каждого опрашивающего порта:
 - 0 – «самый высокий»;
 - 1 – «очень высокий»;
 - 2 – «высокий»;
 - 3 – «нормальный»;
 - 4 – «низкий»;
 - 5 – «самый низкий»;
 - 6 – «почти остановленный».
 По умолчанию задан приоритет = 3.
- **INT64 PORTBAUD** – скорость работы порта измеряется в байт/сек. По умолчанию скорость равна 9600.
- **STRING RPCSTATEPORT** – константа определяющая состояние порта:
 - «OPEN» - открыть порт;
 - «CLOSE» - закрыть порт.
 По умолчанию порт закрыт.
- **STRING LOOPSTATE** – константа определяющая состояние опроса устройств по данному порту;
 - «ACTIVE» - опрос устройств включен;
 - «DISACTIVE» - опрос устройств выключен.
- **ARRAY DEVICES** – список устройств, которые необходимо включить драйверу RS в цикл опроса, элементами списка являются структуры данных (STRUCT):
 - **STRUCT ADDRESS** – структура данных определяющая адрес устройства в системе, содержит следующие поля:
 - **INT64 ADDRDEVICE** – адрес устройства по RS 485;
 - **INT64 ADDRPULT** – адрес пульта по RS 232;
 - **INT64 ADDRPORT** – номер порта, к которому подключен интерфейс.
 - **STRING RPCSTATUSDEVICE** – константа определяющая состояние устройства по отношению к циклу опроса устройств:
 - «ON» - включить в цикл опроса;
 - «OFF» - исключить из цикла опроса.
 - **INT64 TYPE** – тип добавляемого устройства (см. таблицу 1 в приложении А);
 - **INT64 VERSION** - версия прибора;
 - **INT64 PRIORITYDEVICE** – приоритет опроса устройств в цикле опроса.

В ответ на запрос модуль высылает стандартный ответ об исполнении без указания возвращаемых данных в структуре RESULTDATA или формирует ответ с передачей кода ошибки.

Пример: конфигурации модуля опроса на работу с Сигнал 20П с адресом 70 подключенному к пульта 4 и порту 1.

```
<?xml version="1.0" encoding="Windows-1251" ?>
```

```
<methodCall>
```

```
<methodName>SETCONFIGURATIONHWSRV</methodName>
```

```

<params>

<param>

<value>
<struct>
  <member>
    <name>GUID</name>
    <value><string>{55FD1B46-00B1-4421-B6F4-41AB337563AB}</string></value>
  </member>
  <member>
    <name>IPSERVER</name>
    <value><string>127.0.0.1</string></value>
  </member>
  <member>
    <name>PORTSERVER</name>
    <value><int>8090</int></value>
  </member>
  <member>
    <name>COUNTPORTS</name>
    <value><int>1</int></value>
  </member>
  <member>
    <name>PORTS</name>
</member>
<value>
<array>
  <data>
<value>
<struct>
  <member>
    <name>NUMBERPORT</name>
    <value><int>1</int></value>
  </member>
  <member>
    <name>TYPEPORT</name>
    <value><string>COM</string></value>
  </member>
  <member>
    <name>TYPEPROTOCOL</name>
    <value><int>1</int></value>
  </member>
  <member>
    <name>TYPECONVERTER</name>
    <value><int>2</int></value>
  </member>
  <member>
    <name>PORTPRIORITY</name>
    <value><int>3</int></value>
  </member>
  <member>
    <name>PORTBAUD</name>
    <value><int>9600</int></value>
  </member>
  <member>
    <name>PORTSTATE</name>
    <value><string>OPEN</string></value>
  </member>
  <member>
    <name>LOOPSTATE</name>
    <value><string>ACTIVE</string></value>
  </member>
</struct>
</value>

</data>
</array>
</value>

```

```
</member>
</struct>
</value>
```

```
</param>
</params>
</methodCall>
```

Ответ:

```
<?xml version="1.0" encoding="Windows-1251" ?>
```

```
<methodResponse>

  <params>

    <param>

      <value>
        <struct>
          <member>
            <name>RESULT</name>
            <value><string>METHOD IS EXECUTE</string></value>
          </member>
          <member>
            <name>RESULTDATA</name>
      </value>
      <struct>
        <member>
          <name>GUID</name>
          <value><string>{55FD1B46-00B1-4421-B6F4-41AB337563AB}</string></value>
        </member>
      </struct>
    </value>

    </member>
  </struct>
</value>

  </param>
</params>
</methodResponse>
```

Если необходимо запросить конфигурацию модуля опроса необходимо выполнить метод GETCONFIGURATIONHWSERVER.

Параметры метода GetConfigurationhwServer:

Содержит один параметр типа STRUCT (структура), в котором передаются следующие данные:

- **STRING GUID** – жетон безопасности.

В ответ на запрос модуль высылает стандартный ответ об исполнении, возвращая конфигурацию модуля в структуре данных RESULTDATA, поля данной структуры такие же,

как в запросе метода SETCONFIGURATIONHWSERVER.
Пример запроса конфигурации:

Запрос:

```
<?xml version="1.0" encoding="Windows-1251" ?>

<methodCall>

  <methodName>GETCONFIGURATIONHWSERVER</methodName>

  <params>

    <param>

      <value>
        <struct>
          <member>
            <name>GUID</name>
            <value><string>{55FD1B46-00B1-4421-B6F4-41AB337563AB}</string></value>
          </member>
        </struct>
      </value>

      </param>

    </params>

  </methodCall>
```

Ответ:

```
<xml version="1.0" encoding="Windows-1251" ?>

<methodResponse>

  <params>

    <param>

      <value>
        <struct>
          <member>
            <name>RESULT</name>
            <value><string>METHOD IS EXECUTE</string></value>
          </member>
          <member>
            <name>RESULTDATA</name>
            <value>
              <struct>
                <member>
                  <name>NAMERESULT</name>
                  <value><string>Configuration of HWServer</string></value>
                </member>
                <member>
                  <name>METHODNAME</name>
                  <value><string>GETCONFIGURATIONHWSERVER</string></value>
                </member>
                <member>
                  <name>PORTS</name>
                  <value>
                    <array>
                      <data>
                    </data>
                  </value>
                </member>
              </struct>
            </value>
          </member>
        </struct>
      </value>
    </param>
  </params>
</methodResponse>
```

```

    <name>NUMBERPORT</name>
    <value><int>1</int></value>
  </member>
  <member>
    <name>PORTSTATE</name>
    <value><string>OPEN</string></value>
  </member>
  <member>
    <name>TYPEPROTOCOL</name>
    <value><int>1</int></value>
  </member>
  <member>
    <name>TYPEPROTOCOLNAME</name>
    <value><string>Обмен с приборами</string></value>
  </member>
  <member>
    <name>TYPEPORT</name>
    <value><string>COM</string></value>
  </member>
  <member>
    <name>PORTPRIORITY</name>
    <value><int>0</int></value>
  </member>
  <member>
    <name>PORTPRIORITYNAME</name>
    <value><string>самый высокий</string></value>
  </member>
  <member>
    <name>TYPECONVERTER</name>
    <value><int>2</int></value>
  </member>
  <member>
    <name>NAMECONVERTER</name>
    <value><string>C2000</string></value>
  </member>
  <member>
    <name>PORTBAUD</name>
    <value><double>9600</double></value>
  </member>
  <member>
    <name>LOOPSTATE</name>
    <value><string>ACTIVE</string></value>
  </member>
</struct>
</value>

<value>
  <struct>
    <member>
      <name>NUMBERPORT</name>
      <value><int>2</int></value>
    </member>
    <member>
      <name>PORTSTATE</name>
      <value><string>OPEN</string></value>
    </member>
    <member>
      <name>TYPEPROTOCOL</name>
      <value><int>1</int></value>
    </member>
    <member>
      <name>TYPEPROTOCOLNAME</name>
      <value><string>Обмен с приборами</string></value>
    </member>
    <member>
      <name>TYPEPORT</name>
      <value><string>COM</string></value>
    </member>
  </struct>

```

```

<member>
  <name>PORTPRIORITY</name>
  <value><int>0</int></value>
</member>
<member>
  <name>PORTPRIORITYNAME</name>
  <value><string>самый высокий</string></value>
</member>
<member>
  <name>TYPECONVERTER</name>
  <value><int>0</int></value>
</member>
<member>
  <name>NAMECONVERTER</name>
  <value><string>ПИ</string></value>
</member>
<member>
  <name>PORTBAUD</name>
  <value><double>9600</double></value>
</member>
<member>
  <name>LOOPSTATE</name>
  <value><string>DISACTIVE</string></value>
</member>
</struct>
</value>

<value>
<struct>
  <member>
    <name>NUMBERPORT</name>
    <value><int>3</int></value>
  </member>
  <member>
    <name>PORTSTATE</name>
    <value><string>CLOSE</string></value>
  </member>
  <member>
    <name>TYPEPROTOCOL</name>
    <value><int>1</int></value>
  </member>
  <member>
    <name>TYPEPROTOCOLNAME</name>
    <value><string>Обмен с приборами</string></value>
  </member>
  <member>
    <name>TYPEPORT</name>
    <value><string>COM</string></value>
  </member>
  <member>
    <name>PORTPRIORITY</name>
    <value><int>0</int></value>
  </member>
  <member>
    <name>PORTPRIORITYNAME</name>
    <value><string>самый высокий</string></value>
  </member>
  <member>
    <name>TYPECONVERTER</name>
    <value><int>2</int></value>
  </member>
  <member>
    <name>NAMECONVERTER</name>
    <value><string>C2000</string></value>
  </member>
  <member>
    <name>PORTBAUD</name>
    <value><double>9600</double></value>
  </member>

```

```

</member>

<member>
  <name>LOOPSTATE</name>
  <value><string>DISACTIVE</string></value>
</member>
</struct>
</value>

<value>
<struct>
  <member>
    <name>NUMBERPORT</name>
    <value><int>4</int></value>
  </member>
  <member>
    <name>PORTSTATE</name>
    <value><string>CLOSE</string></value>
  </member>
  <member>
    <name>TYPEPROTOCOL</name>
    <value><int>2</int></value>
  </member>
  <member>
    <name>TYPEPROTOCOLNAME</name>
    <value><string>ПКУ обмен с приборами</string></value>
  </member>
  <member>
    <name>TYPEPORT</name>
    <value><string>COM</string></value>
  </member>
  <member>
    <name>PORTPRIORITY</name>
    <value><int>0</int></value>
  </member>
  <member>
    <name>PORTPRIORITYNAME</name>
    <value><string>самый высокий</string></value>
  </member>
  <member>
    <name>TYPECONVERTER</name>
    <value><int>0</int></value>
  </member>
  <member>
    <name>NAMECONVERTER</name>
    <value><string>ПИИ</string></value>
  </member>
  <member>
    <name>PORTBAUD</name>
    <value><double>9600</double></value>
  </member>
  <member>
    <name>LOOPSTATE</name>
    <value><string>DISACTIVE</string></value>
  </member>
</struct>
</value>

<value>
<struct>
  <member>
    <name>NUMBERPORT</name>
    <value><int>5</int></value>
  </member>
  <member>
    <name>PORTSTATE</name>
    <value><string>OPEN</string></value>
  </member>

```



```

<member>
  <name>TYPEPROTOCOL</name>
  <value><int>2</int></value>
</member>
<member>
  <name>TYPEPROTOCOLNAME</name>
  <value><string>ПКУ обмен с приборами</string></value>
</member>
<member>
  <name>TYPEPORT</name>
  <value><string>COM</string></value>
</member>
<member>
  <name>PORTPRIORITY</name>
  <value><int>0</int></value>
</member>
<member>
  <name>PORTPRIORITYNAME</name>
  <value><string>самый высокий</string></value>
</member>
<member>
  <name>TYPECONVERTER</name>
  <value><int>0</int></value>
</member>
<member>
  <name>NAMECONVERTER</name>
  <value><string>ПИ</string></value>
</member>
<member>
  <name>PORTBAUD</name>
  <value><double>9600</double></value>
</member>
<member>
  <name>LOOPSTATE</name>
  <value><string>DISACTIVE</string></value>
</member>
</struct>
</value>

```

```

<value>
<struct>
  <member>
    <name>NUMBERPORT</name>
    <value><int>6</int></value>
  </member>
  <member>
    <name>PORTSTATE</name>
    <value><string>OPEN</string></value>
  </member>
  <member>
    <name>TYPEPROTOCOL</name>
    <value><int>1</int></value>
  </member>
  <member>
    <name>TYPEPROTOCOLNAME</name>
    <value><string>Обмен с приборами</string></value>
  </member>
  <member>
    <name>TYPEPORT</name>
    <value><string>COM</string></value>
  </member>
  <member>
    <name>PORTPRIORITY</name>
    <value><int>0</int></value>
  </member>
  <member>
    <name>PORTPRIORITYNAME</name>
    <value><string>самый высокий</string></value>
  </member>
</struct>
</value>

```

```

</member>
<member>
  <name>TYPECONVERTER</name>
  <value><int>0</int></value>
</member>
<member>
  <name>NAMECONVERTER</name>
  <value><string>ПИ</string></value>
</member>
<member>
  <name>PORTBAUD</name>
  <value><double>9600</double></value>
</member>
<member>
  <name>LOOPSTATE</name>
  <value><string>DISACTIVE</string></value>
</member>
</struct>
</value>

</data>
</array>
</value>

</member>
</struct>
</value>

</member>
</struct>
</value>

</param>

</params>

</methodResponse>

```

Для того чтобы узнать тип и версию подключенного к интерфейсу устройства необходимо использовать метод **GetDevice**.

Параметры метода GetDevice:

- **STRING GUID** – жетон безопасности;
- **STRING IPSERVER** – Ip адрес сервера, на который необходимо передать результат исполнения команды;
- **INT64 PORTSERVER** – порт сервера, на который необходимо передать результат исполнения команды;
- **ARRAY DEVICES** – массив содержит список устройств, у которых необходимо узнать тип и версию, элементами списка являются структуры данных (STRUCT):
 - **STRUCT ADDRESS** – структура данных определяющая адрес устройства в системе, содержит следующие поля:
 - **INT64 ADDRDEVICE** – адрес устройства по RS 485;
 - **INT64 ADDRPULT** – адрес пульта по RS 232;
 - **INT64 ADDRPORT** – номер порта, к которому подключен интерфейс.
- **STRING RESULTRETURNMETHOD** – имя метода который необходимо вызвать при

возврате результата;

- **STRUCT RETURNDATAFROMCLIENT** – структура данных которую необходимо вернуть клиенту при генерации ответа на команду;

В ответ на запрос модуль высылает стандартный ответ об исполнении, возвращая конфигурацию модуля в структуре данных **RESULTDATA**, параметры структуры указаны ниже:

- **STRING NAMEDEVICE** – строковая константа определяющая наименование типа подключенного устройства (см. таблицу 1 в приложении А);
- **INT64 TYPE** – цифровая константа типа устройства (см. таблицу 1 в приложении А);
 - **INT64 VERSION** – версия устройства;
 - **STRUCT ADDRESS** – структура данных определяющая адрес устройства в системе, содержит следующие поля:
 - **INT64 ADDRDEVICE** – адрес устройства по RS 485;
 - **INT64 ADDRPU** – адрес пульта по RS 232;
 - **INT64 ADDRPORT** – номер порта, к которому подключен интерфейс.
 - **INT64 STATE** – цифровая константа состояния устройства (см. таблицу 2 приложения А);
 - **STRING NAMESTATE** – строковая константа наименования состояния (см. таблицу 2 приложения А);

Пример:

Request Server(от 11.08.2006 17:46:32):<?xml version="1.0" encoding="Windows-1251" ?>

```
<methodCall>
  <methodName>GETDEVICE</methodName>
  <params>
    <param>
      <value>
        <struct>
          <member>
            <name>GUID</name>
            <value><string>{0DD6BCB9-D7F6-4853-9AF3-B4D1FB994212}</string></value>
          </member>
          <member>
            <name>IPSERVER</name>
            <value><string>192.168.10.109</string></value>
          </member>
          <member>
            <name>PORTSERVER</name>
            <value><int>8090</int></value>
          </member>
          <member>
            <name>RESULTRETURNMETHOD</name>
            <value><string>ASYNCRESLT</string></value>
          </member>
          <member>
            <name>DEVICES</name>
          </member>
        </struct>
      </value>
    </param>
  </params>
</methodCall>
```

```

    <struct>
      <member>
        <name>RETURNDATAFROMCLIENT</name>
    </member>
  </struct>
</value>

  <struct>
    <member>
      <name>Text</name>
      <value><string>Данные которые нужно возвратить </string></value>
    </member>
  </struct>
</value>

  </member>
</member>
  <name>ADDRESS</name>
<value>
  <struct>
    <member>
      <name>ADDRDEVICE</name>
      <value><int>2</int></value>
    </member>
    <member>
      <name>ADDRPULT</name>
      <value><int>0</int></value>
    </member>
    <member>
      <name>ADDRPORT</name>
      <value><int>1</int></value>
    </member>
  </struct>
</value>

  </member>
</struct>
</value>

</data>
</array>
</value>

  </member>
</struct>
</value>

  </param>
</params>
</methodCall>

```

Ответ:

Response Client{от 11.08.2006 17:46:32}:<?xml version="1.0" encoding="Windows-1251" ?>

```

<methodResponse>

  <params>

    <param>

  </param>

</params>

  <value>
    <struct>
      <member>
        <name>RESULT</name>
        <value><string>METHOD IS EXECUTE</string></value>
      </member>
    </struct>
  </value>

```

```

    <name>RESULTDATA</name>
<value>
  <struct>
    <member>
      <name>NAMERESULT</name>
      <value><string>GetDevice</string></value>
    </member>
    <member>
      <name>METHODNAME</name>
      <value><string>GETDEVICE</string></value>
    </member>
  </struct>
</value>

</member>
</struct>
</value>

</param>

</params>

</methodResponse>

```

Асинхронный ответ:

```

<?xml version="1.0" encoding="Windows-1251" ?>

<methodCall>

  <methodName>ASYNCRESLUT</methodName>

  <params>

    <param>

<value>
  <struct>
    <member>
      <name>GUID</name>
      <value><string>{50F71AEE-FD70-45D7-B26B-CC899D19401C}</string></value>
    </member>
    <member>
      <name>RESULT</name>
      <value><string>METHOD IS EXECUTE</string></value>
    </member>
    <member>
      <name>RESULTDATA</name>
<value>
  <struct>
    <member>
      <name>NAMERESULT</name>
      <value><string>ASYNCRESLUT</string></value>
    </member>
    <member>
      <name>METHODNAME</name>
      <value><string>GETDEVICE</string></value>
    </member>
    <member>
      <name>RETURNDATAFROMCLIENT</name>
<value>
  <struct>
    <member>
      <name>Text</name>
      <value><string>Данные которые нужно возвратить по любому</string></value>
    </member>
  </struct>

```

```

</value>

  </member>
  <member>
    <name>DEVICES</name>
  </member>
</value>
<struct>
  <member>
    <name>NAMEDEVICE</name>
    <value><string>СИГНАЛ-20П</string></value>
  </member>
  <member>
    <name>TYPE</name>
    <value><int>2</int></value>
  </member>
  <member>
    <name>VERSION</name>
    <value><int>167</int></value>
  </member>
  <member>
    <name>NAME</name>
    <value><int>2</int></value>
  </member>
  <member>
    <name>ADDRESS</name>
  </member>
</struct>
</value>

  <member>
    <name>ADDRDEVICE</name>
    <value><int>2</int></value>
  </member>
  <member>
    <name>ADDRPORT</name>
    <value><int>1</int></value>
  </member>
</struct>
</value>

  </member>
  <member>
    <name>STATE</name>
    <value><int>152</int></value>
  </member>
  <member>
    <name>NAMESTATE</name>
    <value><string>НОРМА</string></value>
  </member>
</struct>
</value>

  </member>
</struct>
</value>

  </member>
</struct>
</value>

</param>

</params>

</methodCall>

```

Для того чтобы узнать состояние объектов системы охраны, таких как: шлейф, зона, считыватель, выход необходимо выполнить метод `GetStateElements`.

Параметры метода *GetStateElements*:

- **STRING GUID** – жетон безопасности;
- **ARRAY ZONES** – содержит список зон, у которых необходимо запросить состояние, элементами списка являются структуры данных (тип STRUCT), каждый элемент содержит следующие поля:
 - **STRUCT ADDRESS** – структура данных определяющая адрес зоны в системе, содержит следующие поля:
 - **INT64 ADDRELEMENT** – адрес элемента (номер шлейфа), зависит от типа устройства;
 - **INT64 ADDRDEVICE** – адрес устройства по RS 485;
 - **INT64 ADDRPUULT** – адрес пульта по RS 232;
 - **INT64 ADDRPORT** – номер порта, к которому подключен интерфейс.
- **ARRAY OUTKEYS** – содержит список выходов, у которых необходимо запросить состояние, элементами списка являются структуры данных (тип STRUCT), каждый элемент содержит следующие поля:
 - **STRUCT ADDRESS** – структура данных определяющая адрес выхода в системе, содержит следующие поля:
 - **INT64 ADDRELEMENT** – адрес элемента (номер реле), зависит от типа устройства;
 - **INT64 ADDRDEVICE** – адрес устройства по RS 485;
 - **INT64 ADDRPUULT** – адрес пульта по RS 232;
 - **INT64 ADDRPORT** – номер порта, к которому подключен интерфейс.
- **ARRAY READERS** – содержит список считывателей, у которых необходимо запросить состояние, элементами списка являются структуры данных (тип STRUCT), каждый элемент содержит следующие данные:
 - **STRUCT ADDRESS** – структура данных определяющая адрес считывателя в системе, содержит следующие поля:
 - **INT64 ADDRELEMENT** – адрес элемента (номер считывателя), зависит от типа устройства;
 - **INT64 ADDRDEVICE** – адрес устройства по RS 485;
 - **INT64 ADDRPUULT** – адрес пульта по RS 232;
 - **INT64 ADDRPORT** – номер порта, к которому подключен интерфейс.

Если состояние элемента известно, то ответ будет возвращен «немедленно» в ответе на команду, в противном случае будет произведен запрос состояния и ответ будет возвращен с помощью события ONCHANGESTATEELEMENTS.

Пример запроса состояний элементов по прибору с адресом 2 для порта 1:

```
<?xml version="1.0" encoding="Windows-1251" ?>
```

```
<methodCall>
```

```
<methodName>GETSTATEELEMENTS</methodName>
```

```
<params>
```

```

<param>
<value>
<struct>
  <member>
    <name>GUID</name>
    <value><string>{16A0093F-986A-4E83-BE00-69A248F00AE9}</string></value>
  </member>
  <member>
    <name>IPSERVER</name>
    <value><string>192.168.10.109</string></value>
  </member>
  <member>
    <name>DEVICES</name>
  </member>
</struct>
<struct>
  <member>
    <name>ADDRESS</name>
  </member>
</struct>
<array>
  <data>
<value>
  <struct>
    <member>
      <name>ADDRDEVICE</name>
      <value><int>2</int></value>
    </member>
    <member>
      <name>ADDRPORT</name>
      <value><int>1</int></value>
    </member>
  </struct>
</value>

  </data>
</array>
</value>

  </member>
</struct>
</value>

  </member>
  <member>
    <name>ZONES</name>
  </member>
</struct>
<array>
  <data>
<value>
  <struct>
    <member>
      <name>ADDRESS</name>
    </member>
  </struct>
  <struct>
    <member>
      <name>ADDRELEMENT</name>
      <value><int>1</int></value>
    </member>
    <member>
      <name>ADDRDEVICE</name>
      <value><int>2</int></value>
    </member>
    <member>
      <name>ADDRPORT</name>
      <value><int>1</int></value>
    </member>
  </struct>
</value>

```



```
<value>
<struct>
  <member>
    <name>ADDRELEMENT</name>
    <value><int>2</int></value>
  </member>
  <member>
    <name>ADDRDEVICE</name>
    <value><int>2</int></value>
  </member>
  <member>
    <name>ADDRPORT</name>
    <value><int>1</int></value>
  </member>
</struct>
</value>
```

```
<value>
<struct>
  <member>
    <name>ADDRELEMENT</name>
    <value><int>3</int></value>
  </member>
  <member>
    <name>ADDRDEVICE</name>
    <value><int>2</int></value>
  </member>
  <member>
    <name>ADDRPORT</name>
    <value><int>1</int></value>
  </member>
</struct>
</value>
```

```
<value>
<struct>
  <member>
    <name>ADDRELEMENT</name>
    <value><int>4</int></value>
  </member>
  <member>
    <name>ADDRDEVICE</name>
    <value><int>2</int></value>
  </member>
  <member>
    <name>ADDRPORT</name>
    <value><int>1</int></value>
  </member>
</struct>
</value>
```

```
<value>
<struct>
  <member>
    <name>ADDRELEMENT</name>
    <value><int>5</int></value>
  </member>
  <member>
    <name>ADDRDEVICE</name>
    <value><int>2</int></value>
  </member>
  <member>
    <name>ADDRPORT</name>
    <value><int>1</int></value>
  </member>
</struct>
</value>
```

```
<value>
<struct>
  <member>
    <name>ADDRELEMENT</name>
    <value><int>6</int></value>
  </member>
  <member>
    <name>ADDRDEVICE</name>
    <value><int>2</int></value>
  </member>
  <member>
    <name>ADDRPORT</name>
    <value><int>1</int></value>
  </member>
</struct>
</value>
```

```
<value>
<struct>
  <member>
    <name>ADDRELEMENT</name>
    <value><int>7</int></value>
  </member>
  <member>
    <name>ADDRDEVICE</name>
    <value><int>2</int></value>
  </member>
  <member>
    <name>ADDRPORT</name>
    <value><int>1</int></value>
  </member>
</struct>
</value>
```

```
<value>
<struct>
  <member>
    <name>ADDRELEMENT</name>
    <value><int>8</int></value>
  </member>
  <member>
    <name>ADDRDEVICE</name>
    <value><int>2</int></value>
  </member>
  <member>
    <name>ADDRPORT</name>
    <value><int>1</int></value>
  </member>
</struct>
</value>
```

```
<value>
<struct>
  <member>
    <name>ADDRELEMENT</name>
    <value><int>9</int></value>
  </member>
  <member>
    <name>ADDRDEVICE</name>
    <value><int>2</int></value>
  </member>
  <member>
    <name>ADDRPORT</name>
    <value><int>1</int></value>
  </member>
</struct>
</value>
```

```
<value>
<struct>
  <member>
    <name>ADDRELEMENT</name>
    <value><int>10</int></value>
  </member>
  <member>
    <name>ADDRDEVICE</name>
    <value><int>2</int></value>
  </member>
  <member>
    <name>ADDRPORT</name>
    <value><int>1</int></value>
  </member>
</struct>
</value>
```

```
<value>
<struct>
  <member>
    <name>ADDRELEMENT</name>
    <value><int>11</int></value>
  </member>
  <member>
    <name>ADDRDEVICE</name>
    <value><int>2</int></value>
  </member>
  <member>
    <name>ADDRPORT</name>
    <value><int>1</int></value>
  </member>
</struct>
</value>
```

```
<value>
<struct>
  <member>
    <name>ADDRELEMENT</name>
    <value><int>12</int></value>
  </member>
  <member>
    <name>ADDRDEVICE</name>
    <value><int>2</int></value>
  </member>
  <member>
    <name>ADDRPORT</name>
    <value><int>1</int></value>
  </member>
</struct>
</value>
```

```
<value>
<struct>
  <member>
    <name>ADDRELEMENT</name>
    <value><int>13</int></value>
  </member>
  <member>
    <name>ADDRDEVICE</name>
    <value><int>2</int></value>
  </member>
  <member>
    <name>ADDRPORT</name>
    <value><int>1</int></value>
  </member>
</struct>
</value>
```

```
<value>
<struct>
  <member>
    <name>ADDRELEMENT</name>
    <value><int>14</int></value>
  </member>
  <member>
    <name>ADDRDEVICE</name>
    <value><int>2</int></value>
  </member>
  <member>
    <name>ADDRPORT</name>
    <value><int>1</int></value>
  </member>
</struct>
</value>
```

```
<value>
<struct>
  <member>
    <name>ADDRELEMENT</name>
    <value><int>15</int></value>
  </member>
  <member>
    <name>ADDRDEVICE</name>
    <value><int>2</int></value>
  </member>
  <member>
    <name>ADDRPORT</name>
    <value><int>1</int></value>
  </member>
</struct>
</value>
```

```
<value>
<struct>
  <member>
    <name>ADDRELEMENT</name>
    <value><int>16</int></value>
  </member>
  <member>
    <name>ADDRDEVICE</name>
    <value><int>2</int></value>
  </member>
  <member>
    <name>ADDRPORT</name>
    <value><int>1</int></value>
  </member>
</struct>
</value>
```

```
<value>
<struct>
  <member>
    <name>ADDRELEMENT</name>
    <value><int>17</int></value>
  </member>
  <member>
    <name>ADDRDEVICE</name>
    <value><int>2</int></value>
  </member>
  <member>
    <name>ADDRPORT</name>
    <value><int>1</int></value>
  </member>
</struct>
</value>
```

```
<value>
<struct>
  <member>
    <name>ADDRELEMENT</name>
    <value><int>18</int></value>
  </member>
  <member>
    <name>ADDRDEVICE</name>
    <value><int>2</int></value>
  </member>
  <member>
    <name>ADDRPORT</name>
    <value><int>1</int></value>
  </member>
</struct>
</value>
```

```
<value>
<struct>
  <member>
    <name>ADDRELEMENT</name>
    <value><int>19</int></value>
  </member>
  <member>
    <name>ADDRDEVICE</name>
    <value><int>2</int></value>
  </member>
  <member>
    <name>ADDRPORT</name>
    <value><int>1</int></value>
  </member>
</struct>
</value>
```

```
<value>
<struct>
  <member>
    <name>ADDRELEMENT</name>
    <value><int>20</int></value>
  </member>
  <member>
    <name>ADDRDEVICE</name>
    <value><int>2</int></value>
  </member>
  <member>
    <name>ADDRPORT</name>
    <value><int>1</int></value>
  </member>
</struct>
</value>
```

```
</data>
</array>
</value>
```

```
</member>
</struct>
</value>
```

```
</member>
</struct>
</value>
```

```
</param>
```

```
</params>
```

</methodCall>

Ответ «Модуля управления» с состоянием элементов:

Response Client{or 14.08.2006 9:44:09}:<?xml version="1.0" encoding="Windows-1251" ?>

<methodResponse>

<params>

<param>

<value>

<struct>

<member>

<name>RESULT</name>

<value><string>METHOD IS EXECUTE</string></value>

</member>

<member>

<name>RESULTDATA</name>

<value>

<struct>

<member>

<name>METHODNAME</name>

<value><string>GETSTATEELEMENTS</string></value>

</member>

<member>

<name>STATEDEVICES</name>

<value>

<array>

<data>

<value>

<struct>

<member>

<name>ADDRESS</name>

<value>

<struct>

<member>

<name>ADDRDEVICE</name>

<value><int>2</int></value>

</member>

<member>

<name>ADDRPORT</name>

<value><int>1</int></value>

</member>

</struct>

</value>

</member>

<member>

<name>STATE</name>

<value><int>149</int></value>

</member>

<member>

<name>NAMESTATE</name>

<value><string>ТРЕБОГА ВЗЛЮМА</string></value>

</member>

</struct>

</value>

</data>

</array>

</value>

</member>

<member>

<name>STATEZONES</name>

```

<value>
  <array>
    <data>
<value>
  <struct>
    <member>
      <name>ADDRESS</name>
<value>
  <struct>
    <member>
      <name>ADDRELEMENT</name>
      <value><int>1</int></value>
    </member>
    <member>
      <name>ADDRDEVICE</name>
      <value><int>2</int></value>
    </member>
    <member>
      <name>ADDRPORT</name>
      <value><int>1</int></value>
    </member>
  </struct>
</value>

  </member>
  <member>
    <name>STATE</name>
    <value><int>109</int></value>
  </member>
  <member>
    <name>NAMESTATE</name>
    <value><string>ЧЯТ</string></value>
  </member>
</struct>
</value>

<value>
  <struct>
    <member>
      <name>ADDRESS</name>
<value>
  <struct>
    <member>
      <name>ADDRELEMENT</name>
      <value><int>2</int></value>
    </member>
    <member>
      <name>ADDRDEVICE</name>
      <value><int>2</int></value>
    </member>
    <member>
      <name>ADDRPORT</name>
      <value><int>1</int></value>
    </member>
  </struct>
</value>

  </member>
  <member>
    <name>STATE</name>
    <value><int>24</int></value>
  </member>
  <member>
    <name>NAMESTATE</name>
    <value><string>ВЗЯТИЕ</string></value>
  </member>
</struct>
</value>

```

```

<value>
<struct>
  <member>
    <name>ADDRESS</name>
</member>
</struct>
</value>

<value>
<struct>
  <member>
    <name>ADDRELEMENT</name>
    <value><int>3</int></value>
  </member>
  <member>
    <name>ADDRDEVICE</name>
    <value><int>2</int></value>
  </member>
  <member>
    <name>ADDRPORT</name>
    <value><int>1</int></value>
  </member>
</struct>
</value>

</member>
<member>
  <name>STATE</name>
  <value><int>109</int></value>
</member>
<member>
  <name>NAMESTATE</name>
  <value><string>ЧЯТ</string></value>
</member>
</struct>
</value>

<value>
<struct>
  <member>
    <name>ADDRESS</name>
</member>
</struct>
</value>

<value>
<struct>
  <member>
    <name>ADDRELEMENT</name>
    <value><int>4</int></value>
  </member>
  <member>
    <name>ADDRDEVICE</name>
    <value><int>2</int></value>
  </member>
  <member>
    <name>ADDRPORT</name>
    <value><int>1</int></value>
  </member>
</struct>
</value>

</member>
<member>
  <name>STATE</name>
  <value><int>24</int></value>
</member>
<member>
  <name>NAMESTATE</name>
  <value><string>ВЗЯТИЕ</string></value>
</member>
</struct>
</value>

<value>

```



```

<struct>
  <member>
    <name>ADDRESS</name>
</member>
</struct>
</value>

<struct>
  <member>
    <name>ADDRELEMENT</name>
    <value><int>5</int></value>
  </member>
  <member>
    <name>ADDRDEVICE</name>
    <value><int>2</int></value>
  </member>
  <member>
    <name>ADDRPORT</name>
    <value><int>1</int></value>
  </member>
</struct>
</value>

</member>
<member>
  <name>STATE</name>
  <value><int>109</int></value>
</member>
<member>
  <name>NAMESTATE</name>
  <value><string>ЧЯТ</string></value>
</member>
</struct>
</value>

<value>
  <struct>
    <member>
      <name>ADDRESS</name>
    </member>
  </struct>
</value>

<struct>
  <member>
    <name>ADDRELEMENT</name>
    <value><int>6</int></value>
  </member>
  <member>
    <name>ADDRDEVICE</name>
    <value><int>2</int></value>
  </member>
  <member>
    <name>ADDRPORT</name>
    <value><int>1</int></value>
  </member>
</struct>
</value>

</member>
<member>
  <name>STATE</name>
  <value><int>109</int></value>
</member>
<member>
  <name>NAMESTATE</name>
  <value><string>ЧЯТ</string></value>
</member>
</struct>
</value>

<value>
  <struct>
    <member>

```

```

    <name>ADDRESS</name>
<value>
<struct>
  <member>
    <name>ADDRELEMENT</name>
    <value><int>7</int></value>
  </member>
  <member>
    <name>ADDRDEVICE</name>
    <value><int>2</int></value>
  </member>
  <member>
    <name>ADDRPORT</name>
    <value><int>1</int></value>
  </member>
</struct>
</value>

</member>
<member>
  <name>STATE</name>
  <value><int>24</int></value>
</member>
<member>
  <name>NAMESTATE</name>
  <value><string>ВЗЯТИЕ</string></value>
</member>
</struct>
</value>

<value>
<struct>
  <member>
    <name>ADDRESS</name>
<value>
<struct>
  <member>
    <name>ADDRELEMENT</name>
    <value><int>8</int></value>
  </member>
  <member>
    <name>ADDRDEVICE</name>
    <value><int>2</int></value>
  </member>
  <member>
    <name>ADDRPORT</name>
    <value><int>1</int></value>
  </member>
</struct>
</value>

</member>
<member>
  <name>STATE</name>
  <value><int>24</int></value>
</member>
<member>
  <name>NAMESTATE</name>
  <value><string>ВЗЯТИЕ</string></value>
</member>
</struct>
</value>

<value>
<struct>
  <member>
    <name>ADDRESS</name>
<value>

```

```

<struct>
  <member>
    <name>ADDRELEMENT</name>
    <value><int>9</int></value>
  </member>
  <member>
    <name>ADDRDEVICE</name>
    <value><int>2</int></value>
  </member>
  <member>
    <name>ADDRPORT</name>
    <value><int>1</int></value>
  </member>
</struct>
</value>

  </member>
  <member>
    <name>STATE</name>
    <value><int>109</int></value>
  </member>
  <member>
    <name>NAMESTATE</name>
    <value><string>ЧЯТ</string></value>
  </member>
</struct>
</value>

<value>
  <struct>
    <member>
      <name>ADDRESS</name>
    </member>
  </struct>
</value>
  <struct>
    <member>
      <name>ADDRELEMENT</name>
      <value><int>10</int></value>
    </member>
    <member>
      <name>ADDRDEVICE</name>
      <value><int>2</int></value>
    </member>
    <member>
      <name>ADDRPORT</name>
      <value><int>1</int></value>
    </member>
  </struct>
</value>

  </member>
  <member>
    <name>STATE</name>
    <value><int>109</int></value>
  </member>
  <member>
    <name>NAMESTATE</name>
    <value><string>ЧЯТ</string></value>
  </member>
</struct>
</value>

<value>
  <struct>
    <member>
      <name>ADDRESS</name>
    </member>
  </struct>
  <struct>
    <member>

```

```

    <name>ADDRELEMENT</name>
    <value><int>11</int></value>
  </member>
  <member>
    <name>ADDRDEVICE</name>
    <value><int>2</int></value>
  </member>
  <member>
    <name>ADDRPORT</name>
    <value><int>1</int></value>
  </member>
</struct>
</value>

  </member>
  <member>
    <name>STATE</name>
    <value><int>109</int></value>
  </member>
  <member>
    <name>NAMESTATE</name>
    <value><string>ЧЯТ</string></value>
  </member>
</struct>
</value>

<value>
  <struct>
    <member>
      <name>ADDRESS</name>
    </member>
  </struct>
  <struct>
    <member>
      <name>ADDRELEMENT</name>
      <value><int>12</int></value>
    </member>
    <member>
      <name>ADDRDEVICE</name>
      <value><int>2</int></value>
    </member>
    <member>
      <name>ADDRPORT</name>
      <value><int>1</int></value>
    </member>
  </struct>
</value>

  </member>
  <member>
    <name>STATE</name>
    <value><int>109</int></value>
  </member>
  <member>
    <name>NAMESTATE</name>
    <value><string>ЧЯТ</string></value>
  </member>
</struct>
</value>

<value>
  <struct>
    <member>
      <name>ADDRESS</name>
    </member>
  </struct>
  <struct>
    <member>
      <name>ADDRELEMENT</name>
      <value><int>13</int></value>
    </member>
  </struct>

```

```

</member>
<member>
  <name>ADDRDEVICE</name>
  <value><int>2</int></value>
</member>
<member>
  <name>ADDRPORT</name>
  <value><int>1</int></value>
</member>
</struct>
</value>

</member>
<member>
  <name>STATE</name>
  <value><int>109</int></value>
</member>
<member>
  <name>NAMESTATE</name>
  <value><string>ЧЯТ</string></value>
</member>
</struct>
</value>

<value>
  <struct>
    <member>
      <name>ADDRESS</name>
</member>
<value>
  <struct>
    <member>
      <name>ADDRELEMENT</name>
      <value><int>14</int></value>
    </member>
    <member>
      <name>ADDRDEVICE</name>
      <value><int>2</int></value>
    </member>
    <member>
      <name>ADDRPORT</name>
      <value><int>1</int></value>
    </member>
  </struct>
</value>

</member>
<member>
  <name>STATE</name>
  <value><int>109</int></value>
</member>
<member>
  <name>NAMESTATE</name>
  <value><string>ЧЯТ</string></value>
</member>
</struct>
</value>

<value>
  <struct>
    <member>
      <name>ADDRESS</name>
</member>
<value>
  <struct>
    <member>
      <name>ADDRELEMENT</name>
      <value><int>15</int></value>
    </member>
    <member>

```

```

    <name>ADDRDEVICE</name>
    <value><int>2</int></value>
  </member>
  <member>
    <name>ADDRPORT</name>
    <value><int>1</int></value>
  </member>
</struct>
</value>

  </member>
  <member>
    <name>STATE</name>
    <value><int>109</int></value>
  </member>
  <member>
    <name>NAMESTATE</name>
    <value><string>ЧЯТ</string></value>
  </member>
</struct>
</value>

<value>
  <struct>
    <member>
      <name>ADDRESS</name>
    </member>
  </struct>
  <value>
    <struct>
      <member>
        <name>ADDRELEMENT</name>
        <value><int>16</int></value>
      </member>
      <member>
        <name>ADDRDEVICE</name>
        <value><int>2</int></value>
      </member>
      <member>
        <name>ADDRPORT</name>
        <value><int>1</int></value>
      </member>
    </struct>
  </value>

  </member>
  <member>
    <name>STATE</name>
    <value><int>109</int></value>
  </member>
  <member>
    <name>NAMESTATE</name>
    <value><string>ЧЯТ</string></value>
  </member>
</struct>
</value>

<value>
  <struct>
    <member>
      <name>ADDRESS</name>
    </member>
  </struct>
  <value>
    <struct>
      <member>
        <name>ADDRELEMENT</name>
        <value><int>17</int></value>
      </member>
      <member>
        <name>ADDRDEVICE</name>
        <value><int>2</int></value>
      </member>
    </struct>
  </value>

```

```

</member>
<member>
  <name>ADDRPORT</name>
  <value><int>1</int></value>
</member>
</struct>
</value>

</member>
<member>
  <name>STATE</name>
  <value><int>109</int></value>
</member>
<member>
  <name>NAMESTATE</name>
  <value><string>ЧЯТ</string></value>
</member>
</struct>
</value>

<value>
<struct>
  <member>
    <name>ADDRESS</name>
<value>
<struct>
  <member>
    <name>ADDRELEMENT</name>
    <value><int>18</int></value>
  </member>
  <member>
    <name>ADDRDEVICE</name>
    <value><int>2</int></value>
  </member>
  <member>
    <name>ADDRPORT</name>
    <value><int>1</int></value>
  </member>
</struct>
</value>

</member>
<member>
  <name>STATE</name>
  <value><int>109</int></value>
</member>
<member>
  <name>NAMESTATE</name>
  <value><string>ЧЯТ</string></value>
</member>
</struct>
</value>

<value>
<struct>
  <member>
    <name>ADDRESS</name>
<value>
<struct>
  <member>
    <name>ADDRELEMENT</name>
    <value><int>19</int></value>
  </member>
  <member>
    <name>ADDRDEVICE</name>
    <value><int>2</int></value>
  </member>
</struct>
</member>

```

```

    <name>ADDRPORT</name>
    <value><int>1</int></value>
  </member>
</struct>
</value>

  </member>
  <member>
    <name>STATE</name>
    <value><int>109</int></value>
  </member>
  <member>
    <name>NAMESTATE</name>
    <value><string>ЧЯТ</string></value>
  </member>
</struct>
</value>

<value>
  <struct>
    <member>
      <name>ADDRESS</name>
    </member>
  </struct>
  <value>
    <struct>
      <member>
        <name>ADDRELEMENT</name>
        <value><int>20</int></value>
      </member>
      <member>
        <name>ADDRDEVICE</name>
        <value><int>2</int></value>
      </member>
      <member>
        <name>ADDRPORT</name>
        <value><int>1</int></value>
      </member>
    </struct>
  </value>

  </member>
  <member>
    <name>STATE</name>
    <value><int>17</int></value>
  </member>
  <member>
    <name>NAMESTATE</name>
    <value><string>НЕВЗЯТИЕ</string></value>
  </member>
</struct>
</value>

</data>
</array>
</value>

  </member>
</struct>
</value>

  </member>
</struct>
</value>

</param>

</params>

```


</methodResponse>

Для того чтобы получить список и состояния устройств, с которыми работает модуль управления необходимо выполнить метод **GetListDevice**.

Параметры метода GetListDevice:

- **STRING GUID** – жетон безопасности.

В ответе на запрос модуль высылает стандартный ответ об исполнении, возвращая список приборов и их состояния в структуре данных **RESULTDATA**, параметры структуры указаны ниже:

- **STRING NAMEDEVICE** – строковая константа определяющая наименование типа подключенного устройства (см. таблицу 1 в приложении А);
- **INT64 TYPE** – цифровая константа типа устройства (см. таблицу 1 в приложении А);
 - **INT64 VERSION** – версия устройства;
 - **STRUCT ADDRESS** – структура данных определяющая адрес устройства в системе, содержит следующие поля:
 - **INT64 ADDRDEVICE** – адрес устройства по RS 485;
 - **INT64 ADDRPU** – адрес пульта по RS 232;
 - **INT64 ADDRPORT** – номер порта, к которому подключен интерфейс.
 - **INT64 STATE** – цифровая константа состояния устройства (см. таблицу 2 в приложении А);
 - **STRING NAMESTATE** – строковая константа наименования состояния.

В примере приведен запрос и ответ об устройствах подключенных к модулю опроса.

Request Server(от 14.08.2006 10:53:53):<?xml version="1.0" encoding="Windows-1251" ?>

<methodCall>

<methodName>GETLISTDEVICE</methodName>

<params>

<param>

<value>

<struct>

<member>

<name>IPSERVER</name>

<value><string>192.168.10.109</string></value>

</member>

<member>

<name>PORTSERVER</name>

<value><int>8090</int></value>

</member>

<member>

<name>STATE</name>

<value><int>0</int></value>

</member>

<member>

<name>GUID</name>

<value><string>{16A0093F-986A-4E83-BE00-69A248F00AE9}</string></value>

```
</member>
</struct>
</value>
```

```
</param>

</params>

</methodCall>
```

Response Client{or 14.08.2006 10:54:04}:<?xml version="1.0" encoding="Windows-1251" ?>

```
<methodResponse>

  <params>

    <param>

      <value>
        <struct>
          <member>
            <name>RESULT</name>
            <value><string>METHOD IS EXECUTE</string></value>
          </member>
          <member>
            <name>RESULTDATA</name>
            <value>
              <struct>
                <member>
                  <name>NAMERESULT</name>
                  <value><string>List of devices</string></value>
                </member>
                <member>
                  <name>METHODNAME</name>
                  <value><string>GETLISTDEVICE</string></value>
                </member>
                <member>
                  <name>COUNTDEVICE</name>
                  <value><int>12</int></value>
                </member>
                <member>
                  <name>DEVICES</name>
                  <value>
                    <array>
                      <data>
                        <value>
                          <struct>
                            <member>
                              <name>NAMEDEVICE</name>
                              <value><string>СИГНАЛ-20П</string></value>
                            </member>
                            <member>
                              <name>TYPE</name>
                              <value><int>2</int></value>
                            </member>
                            <member>
                              <name>VERSION</name>
                              <value><int>124</int></value>
                            </member>
                            <member>
                              <name>NAME</name>
                              <value><int>2</int></value>
                            </member>
                            <member>
                              <name>ADDRESS</name>
                              <value>
```

```

<struct>
  <member>
    <name>ADDRDEVICE</name>
    <value><int>70</int></value>
  </member>
  <member>
    <name>ADDRPORT</name>
    <value><int>1</int></value>
  </member>
</struct>
</value>

  </member>
  <member>
    <name>STATE</name>
    <value><int>250</int></value>
  </member>
  <member>
    <name>NAMESTATE</name>
    <value><string>HET KOHTAKTA</string></value>
  </member>
</struct>
</value>

<value>
  <struct>
    <member>
      <name>NAMEDEVICE</name>
      <value><string>ЧИГ ХАЛ-20П</string></value>
    </member>
    <member>
      <name>TYPE</name>
      <value><int>2</int></value>
    </member>
    <member>
      <name>VERSION</name>
      <value><int>167</int></value>
    </member>
    <member>
      <name>NAME</name>
      <value><int>2</int></value>
    </member>
    <member>
      <name>ADDRESS</name>
    </member>
  </struct>
</value>

  <struct>
    <member>
      <name>ADDRDEVICE</name>
      <value><int>2</int></value>
    </member>
    <member>
      <name>ADDRPORT</name>
      <value><int>1</int></value>
    </member>
  </struct>
</value>

  </member>
  <member>
    <name>STATE</name>
    <value><int>149</int></value>
  </member>
  <member>
    <name>NAMESTATE</name>
    <value><string>ТРЕБОГА ВЗЛОМА</string></value>
  </member>
</struct>
</value>

```

```

<value>
<struct>
  <member>
    <name>NAMEDEVICE</name>
    <value><string>C2000</string></value>
  </member>
  <member>
    <name>TYPE</name>
    <value><int>0</int></value>
  </member>
  <member>
    <name>VERSION</name>
    <value><int>201</int></value>
  </member>
  <member>
    <name>NAME</name>
    <value><int>0</int></value>
  </member>
  <member>
    <name>ADDRESS</name>
</struct>
</value>

  </member>
  <member>
    <name>ADDRDEVICE</name>
    <value><int>4</int></value>
  </member>
  <member>
    <name>ADDRPORT</name>
    <value><int>1</int></value>
  </member>
</struct>
</value>

  </member>
  <member>
    <name>STATE</name>
    <value><int>152</int></value>
  </member>
  <member>
    <name>NAMESTATE</name>
    <value><string>HOPMA</string></value>
  </member>
</struct>
</value>

<value>
<struct>
  <member>
    <name>NAMEDEVICE</name>
    <value><string>C2000-2</string></value>
  </member>
  <member>
    <name>TYPE</name>
    <value><int>16</int></value>
  </member>
  <member>
    <name>VERSION</name>
    <value><int>102</int></value>
  </member>
  <member>
    <name>NAME</name>
    <value><int>16</int></value>
  </member>
  <member>
    <name>ADDRESS</name>
</struct>
</value>

```

```

    <member>
      <name>ADDRDEVICE</name>
      <value><int>10</int></value>
    </member>
    <member>
      <name>ADDRPORT</name>
      <value><int>1</int></value>
    </member>
  </struct>
</value>

  </member>
  <member>
    <name>STATE</name>
    <value><int>149</int></value>
  </member>
  <member>
    <name>NAMESTATE</name>
    <value><string>ТРЕБОГА ВЗЛЮМА</string></value>
  </member>
</struct>
</value>

<value>
  <struct>
    <member>
      <name>NAMEDEVICE</name>
      <value><string>C2000-2</string></value>
    </member>
    <member>
      <name>TYPE</name>
      <value><int>16</int></value>
    </member>
    <member>
      <name>VERSION</name>
      <value><int>104</int></value>
    </member>
    <member>
      <name>NAME</name>
      <value><int>16</int></value>
    </member>
    <member>
      <name>ADDRESS</name>
    </member>
  </struct>
  <struct>
    <member>
      <name>ADDRDEVICE</name>
      <value><int>24</int></value>
    </member>
    <member>
      <name>ADDRPORT</name>
      <value><int>1</int></value>
    </member>
  </struct>
</value>

  </member>
  <member>
    <name>STATE</name>
    <value><int>149</int></value>
  </member>
  <member>
    <name>NAMESTATE</name>
    <value><string>ТРЕБОГА ВЗЛЮМА</string></value>
  </member>
</struct>
</value>

```

```

<value>
<struct>
  <member>
    <name>NAMEDEVICE</name>
    <value><string>C2000-4</string></value>
  </member>
  <member>
    <name>TYPE</name>
    <value><int>4</int></value>
  </member>
  <member>
    <name>VERSION</name>
    <value><int>106</int></value>
  </member>
  <member>
    <name>NAME</name>
    <value><int>4</int></value>
  </member>
  <member>
    <name>ADDRESS</name>
</struct>
</value>

  </member>
  <member>
    <name>STATE</name>
    <value><int>149</int></value>
  </member>
  <member>
    <name>NAMESTATE</name>
    <value><string>ТРЕБОГА ВЗЛОМА</string></value>
  </member>
</struct>
</value>

<value>
<struct>
  <member>
    <name>NAMEDEVICE</name>
    <value><string>C2000-СП1</string></value>
  </member>
  <member>
    <name>TYPE</name>
    <value><int>3</int></value>
  </member>
  <member>
    <name>VERSION</name>
    <value><int>120</int></value>
  </member>
  <member>
    <name>NAME</name>
    <value><int>3</int></value>
  </member>
  <member>
    <name>ADDRESS</name>
</struct>
</value>

```

```

    <name>ADDRDEVICE</name>
    <value><int>32</int></value>
  </member>
  <member>
    <name>ADDRPORT</name>
    <value><int>1</int></value>
  </member>
</struct>
</value>

  </member>
  <member>
    <name>STATE</name>
    <value><int>149</int></value>
  </member>
  <member>
    <name>NAMESTATE</name>
    <value><string>ТРЕБОГ А ВЗЛОМА</string></value>
  </member>
</struct>
</value>

<value>
  <struct>
    <member>
      <name>NAMEDEVICE</name>
      <value><string>C2000-БИ</string></value>
    </member>
    <member>
      <name>TYPE</name>
      <value><int>10</int></value>
    </member>
    <member>
      <name>VERSION</name>
      <value><int>105</int></value>
    </member>
    <member>
      <name>NAME</name>
      <value><int>10</int></value>
    </member>
    <member>
      <name>ADDRESS</name>
    </member>
  </struct>
</value>

  <struct>
    <member>
      <name>ADDRDEVICE</name>
      <value><int>34</int></value>
    </member>
    <member>
      <name>ADDRPORT</name>
      <value><int>1</int></value>
    </member>
  </struct>
</value>

  </member>
  <member>
    <name>STATE</name>
    <value><int>152</int></value>
  </member>
  <member>
    <name>NAMESTATE</name>
    <value><string>HOPMA</string></value>
  </member>
</struct>
</value>

<value>

```

```

<struct>
  <member>
    <name>NAMEDEVICE</name>
    <value><string>C2000-2</string></value>
  </member>
  <member>
    <name>TYPE</name>
    <value><int>16</int></value>
  </member>
  <member>
    <name>VERSION</name>
    <value><int>105</int></value>
  </member>
  <member>
    <name>NAME</name>
    <value><int>16</int></value>
  </member>
  <member>
    <name>ADDRESS</name>
</value>
<struct>
  <member>
    <name>ADDRDEVICE</name>
    <value><int>35</int></value>
  </member>
  <member>
    <name>ADDRPORT</name>
    <value><int>1</int></value>
  </member>
</struct>
</value>

  </member>
  <member>
    <name>STATE</name>
    <value><int>149</int></value>
  </member>
  <member>
    <name>NAMESTATE</name>
    <value><string>TPEBOT A B3JIOMA</string></value>
  </member>
</struct>
</value>

<value>
<struct>
  <member>
    <name>NAMEDEVICE</name>
    <value><string>C2000-4</string></value>
  </member>
  <member>
    <name>TYPE</name>
    <value><int>4</int></value>
  </member>
  <member>
    <name>VERSION</name>
    <value><int>202</int></value>
  </member>
  <member>
    <name>NAME</name>
    <value><int>4</int></value>
  </member>
  <member>
    <name>ADDRESS</name>
</value>
<struct>
  <member>
    <name>ADDRDEVICE</name>

```



```

    <value><int>50</int></value>
  </member>
  <member>
    <name>ADDRPORT</name>
    <value><int>1</int></value>
  </member>
</struct>
</value>

  </member>
  <member>
    <name>STATE</name>
    <value><int>149</int></value>
  </member>
  <member>
    <name>NAMESTATE</name>
    <value><string>TPEBOFA B3JIOMA</string></value>
  </member>
</struct>
</value>

<value>
  <struct>
    <member>
      <name>NAMEDEVICE</name>
      <value><string>C2000-KC</string></value>
    </member>
    <member>
      <name>TYPE</name>
      <value><int>13</int></value>
    </member>
    <member>
      <name>VERSION</name>
      <value><int>103</int></value>
    </member>
    <member>
      <name>NAME</name>
      <value><int>13</int></value>
    </member>
    <member>
      <name>ADDRESS</name>
    </member>
  </struct>
  <struct>
    <member>
      <name>ADDRDEVICE</name>
      <value><int>63</int></value>
    </member>
    <member>
      <name>ADDRPORT</name>
      <value><int>1</int></value>
    </member>
  </struct>
</value>

  </member>
  <member>
    <name>STATE</name>
    <value><int>250</int></value>
  </member>
  <member>
    <name>NAMESTATE</name>
    <value><string>HET KOHTAKTA</string></value>
  </member>
</struct>
</value>

<value>
  <struct>

```

```

    <member>
      <name>NAMEDEVICE</name>
      <value><string>C2000-K</string></value>
    </member>
    <member>
      <name>TYPE</name>
      <value><int>7</int></value>
    </member>
    <member>
      <name>VERSION</name>
      <value><int>105</int></value>
    </member>
    <member>
      <name>NAME</name>
      <value><int>7</int></value>
    </member>
    <member>
      <name>ADDRESS</name>
    <value>
      <struct>
        <member>
          <name>ADDRDEVICE</name>
          <value><int>66</int></value>
        </member>
        <member>
          <name>ADDRPORT</name>
          <value><int>1</int></value>
        </member>
      </struct>
    </value>

    </member>
    <member>
      <name>STATE</name>
      <value><int>149</int></value>
    </member>
    <member>
      <name>NAMESTATE</name>
      <value><string>ТРЕБОГА ВЗЛОМА</string></value>
    </member>
  </struct>
</value>

</data>
</array>
</value>

  </member>
</struct>
</value>

  </member>
</struct>
</value>

</param>

</params>

</methodResponse>
Request Server(от 14.08.2006 10:54:07):<?xml version="1.0" encoding="Windows-1251" ?>

<methodCall>

  <methodName>GETLISTDEVICE</methodName>

  <params>

```

```

<param>

<value>
<struct>
  <member>
    <name>IPSERVER</name>
    <value><string>192.168.10.109</string></value>
  </member>
  <member>
    <name>PORTSERVER</name>
    <value><int>8090</int></value>
  </member>
  <member>
    <name>STATE</name>
    <value><int>0</int></value>
  </member>
  <member>
    <name>GUID</name>
    <value><string>{16A0093F-986A-4E83-BE00-69A248F00AE9}</string></value>
  </member>
</struct>
</value>

</param>

</params>

</methodCall>

```

Response Client{or 14.08.2006 10:55:01};<?xml version="1.0" encoding="Windows-1251" ?>

```

<methodResponse>

  <params>

    <param>

      <value>
        <struct>
          <member>
            <name>RESULT</name>
            <value><string>METHOD IS EXECUTE</string></value>
          </member>
          <member>
            <name>RESULTDATA</name>
            <value>
              <struct>
                <member>
                  <name>NAMERESULT</name>
                  <value><string>List of devices</string></value>
                </member>
                <member>
                  <name>METHODNAME</name>
                  <value><string>GETLISTDEVICE</string></value>
                </member>
                <member>
                  <name>COUNTDEVICE</name>
                  <value><int>12</int></value>
                </member>
                <member>
                  <name>DEVICES</name>
                  <value>
                    <array>
                      <data>

```

```

<struct>
  <member>
    <name>NAMEDEVICE</name>
    <value><string>СИГНАЛ-20П</string></value>
  </member>
  <member>
    <name>TYPE</name>
    <value><int>2</int></value>
  </member>
  <member>
    <name>VERSION</name>
    <value><int>124</int></value>
  </member>
  <member>
    <name>NAME</name>
    <value><int>2</int></value>
  </member>
  <member>
    <name>ADDRESS</name>
  <value>
    <struct>
      <member>
        <name>ADDRDEVICE</name>
        <value><int>70</int></value>
      </member>
      <member>
        <name>ADDRPORT</name>
        <value><int>1</int></value>
      </member>
    </struct>
  </value>

  </member>
  <member>
    <name>STATE</name>
    <value><int>250</int></value>
  </member>
  <member>
    <name>NAMESTATE</name>
    <value><string>HET KOHTAKTA</string></value>
  </member>
</struct>
</value>

<value>
  <struct>
    <member>
      <name>NAMEDEVICE</name>
      <value><string>СИГНАЛ-20П</string></value>
    </member>
    <member>
      <name>TYPE</name>
      <value><int>2</int></value>
    </member>
    <member>
      <name>VERSION</name>
      <value><int>167</int></value>
    </member>
    <member>
      <name>NAME</name>
      <value><int>2</int></value>
    </member>
    <member>
      <name>ADDRESS</name>
  <value>
    <struct>
      <member>
        <name>ADDRDEVICE</name>

```

```

    <value><int>2</int></value>
  </member>
  <member>
    <name>ADDRPORT</name>
    <value><int>1</int></value>
  </member>
</struct>
</value>

  </member>
  <member>
    <name>STATE</name>
    <value><int>149</int></value>
  </member>
  <member>
    <name>NAMESTATE</name>
    <value><string>TPEBOFA B3JIOMA</string></value>
  </member>
</struct>
</value>

<value>
  <struct>
    <member>
      <name>NAMEDEVICE</name>
      <value><string>C2000</string></value>
    </member>
    <member>
      <name>TYPE</name>
      <value><int>0</int></value>
    </member>
    <member>
      <name>VERSION</name>
      <value><int>201</int></value>
    </member>
    <member>
      <name>NAME</name>
      <value><int>0</int></value>
    </member>
    <member>
      <name>ADDRESS</name>
    </member>
  </struct>
  <struct>
    <member>
      <name>ADDRDEVICE</name>
      <value><int>4</int></value>
    </member>
    <member>
      <name>ADDRPORT</name>
      <value><int>1</int></value>
    </member>
  </struct>
</value>

  </member>
  <member>
    <name>STATE</name>
    <value><int>152</int></value>
  </member>
  <member>
    <name>NAMESTATE</name>
    <value><string>HOPMA</string></value>
  </member>
</struct>
</value>

<value>
  <struct>

```

```

<member>
  <name>NAMEDEVICE</name>
  <value><string>C2000-2</string></value>
</member>
<member>
  <name>TYPE</name>
  <value><int>16</int></value>
</member>
<member>
  <name>VERSION</name>
  <value><int>102</int></value>
</member>
<member>
  <name>NAME</name>
  <value><int>16</int></value>
</member>
<member>
  <name>ADDRESS</name>
<value>
<struct>
  <member>
    <name>ADDRDEVICE</name>
    <value><int>10</int></value>
  </member>
  <member>
    <name>ADDRPORT</name>
    <value><int>1</int></value>
  </member>
</struct>
</value>

  </member>
  <member>
    <name>STATE</name>
    <value><int>149</int></value>
  </member>
  <member>
    <name>NAMESTATE</name>
    <value><string>ТРЕБОГА ВЗЛОМА</string></value>
  </member>
</struct>
</value>

<value>
<struct>
  <member>
    <name>NAMEDEVICE</name>
    <value><string>C2000-2</string></value>
  </member>
  <member>
    <name>TYPE</name>
    <value><int>16</int></value>
  </member>
  <member>
    <name>VERSION</name>
    <value><int>104</int></value>
  </member>
  <member>
    <name>NAME</name>
    <value><int>16</int></value>
  </member>
  <member>
    <name>ADDRESS</name>
<value>
<struct>
  <member>
    <name>ADDRDEVICE</name>
    <value><int>24</int></value>

```

```

</member>
<member>
  <name>ADDRPORT</name>
  <value><int>1</int></value>
</member>
</struct>
</value>

</member>
<member>
  <name>STATE</name>
  <value><int>149</int></value>
</member>
<member>
  <name>NAMESTATE</name>
  <value><string>ТРЕБОГ А ВЗЛОМА</string></value>
</member>
</struct>
</value>

<value>
<struct>
  <member>
    <name>NAMEDEVICE</name>
    <value><string>C2000-4</string></value>
  </member>
  <member>
    <name>TYPE</name>
    <value><int>4</int></value>
  </member>
  <member>
    <name>VERSION</name>
    <value><int>106</int></value>
  </member>
  <member>
    <name>NAME</name>
    <value><int>4</int></value>
  </member>
  <member>
    <name>ADDRESS</name>
  </member>
</struct>
</value>

<struct>
  <member>
    <name>ADDRDEVICE</name>
    <value><int>30</int></value>
  </member>
  <member>
    <name>ADDRPORT</name>
    <value><int>1</int></value>
  </member>
</struct>
</value>

</member>
<member>
  <name>STATE</name>
  <value><int>149</int></value>
</member>
<member>
  <name>NAMESTATE</name>
  <value><string>ТРЕБОГ А ВЗЛОМА</string></value>
</member>
</struct>
</value>

<value>
<struct>
  <member>

```

```

    <name>NAMEDEVICE</name>
    <value><string>C2000-CII1</string></value>
  </member>
  <member>
    <name>TYPE</name>
    <value><int>3</int></value>
  </member>
  <member>
    <name>VERSION</name>
    <value><int>120</int></value>
  </member>
  <member>
    <name>NAME</name>
    <value><int>3</int></value>
  </member>
  <member>
    <name>ADDRESS</name>
  <value>
    <struct>
      <member>
        <name>ADDRDEVICE</name>
        <value><int>32</int></value>
      </member>
      <member>
        <name>ADDRPORT</name>
        <value><int>1</int></value>
      </member>
    </struct>
  </value>

  </member>
  <member>
    <name>STATE</name>
    <value><int>149</int></value>
  </member>
  <member>
    <name>NAMESTATE</name>
    <value><string>TPEBOГ A B3JIOMA</string></value>
  </member>
</struct>
</value>

<value>
  <struct>
    <member>
      <name>NAMEDEVICE</name>
      <value><string>C2000-БИ</string></value>
    </member>
    <member>
      <name>TYPE</name>
      <value><int>10</int></value>
    </member>
    <member>
      <name>VERSION</name>
      <value><int>105</int></value>
    </member>
    <member>
      <name>NAME</name>
      <value><int>10</int></value>
    </member>
    <member>
      <name>ADDRESS</name>
  <value>
    <struct>
      <member>
        <name>ADDRDEVICE</name>
        <value><int>34</int></value>
      </member>

```



```

    <member>
      <name>ADDRPORT</name>
      <value><int>1</int></value>
    </member>
  </struct>
</value>

  </member>
  <member>
    <name>STATE</name>
    <value><int>152</int></value>
  </member>
  <member>
    <name>NAMESTATE</name>
    <value><string>HOPMA</string></value>
  </member>
</struct>
</value>

<value>
  <struct>
    <member>
      <name>NAMEDEVICE</name>
      <value><string>C2000-2</string></value>
    </member>
    <member>
      <name>TYPE</name>
      <value><int>16</int></value>
    </member>
    <member>
      <name>VERSION</name>
      <value><int>105</int></value>
    </member>
    <member>
      <name>NAME</name>
      <value><int>16</int></value>
    </member>
    <member>
      <name>ADDRESS</name>
    </member>
  </struct>
</value>

  <struct>
    <member>
      <name>ADDRDEVICE</name>
      <value><int>35</int></value>
    </member>
    <member>
      <name>ADDRPORT</name>
      <value><int>1</int></value>
    </member>
  </struct>
</value>

  </member>
  <member>
    <name>STATE</name>
    <value><int>149</int></value>
  </member>
  <member>
    <name>NAMESTATE</name>
    <value><string>ТРЕБОГА ВЗЛЮМА</string></value>
  </member>
</struct>
</value>

<value>
  <struct>
    <member>
      <name>NAMEDEVICE</name>

```

```

    <value><string>C2000-4</string></value>
  </member>
  <member>
    <name>TYPE</name>
    <value><int>4</int></value>
  </member>
  <member>
    <name>VERSION</name>
    <value><int>202</int></value>
  </member>
  <member>
    <name>NAME</name>
    <value><int>4</int></value>
  </member>
  <member>
    <name>ADDRESS</name>
  <value>
    <struct>
      <member>
        <name>ADDRDEVICE</name>
        <value><int>50</int></value>
      </member>
      <member>
        <name>ADDRPORT</name>
        <value><int>1</int></value>
      </member>
    </struct>
  </value>

  </member>
  <member>
    <name>STATE</name>
    <value><int>149</int></value>
  </member>
  <member>
    <name>NAMESTATE</name>
    <value><string>ТРЕБОГ А ВЗЛЮМА</string></value>
  </member>
</struct>
</value>

<value>
  <struct>
    <member>
      <name>NAMEDEVICE</name>
      <value><string>C2000-KC</string></value>
    </member>
    <member>
      <name>TYPE</name>
      <value><int>13</int></value>
    </member>
    <member>
      <name>VERSION</name>
      <value><int>103</int></value>
    </member>
    <member>
      <name>NAME</name>
      <value><int>13</int></value>
    </member>
    <member>
      <name>ADDRESS</name>
  <value>
    <struct>
      <member>
        <name>ADDRDEVICE</name>
        <value><int>63</int></value>
      </member>
    </struct>
  </value>

```

```

    <name>ADDRPORT</name>
    <value><int>1</int></value>
  </member>
</struct>
</value>

  </member>
  <member>
    <name>STATE</name>
    <value><int>152</int></value>
  </member>
  <member>
    <name>NAMESTATE</name>
    <value><string>HOPMA</string></value>
  </member>
</struct>
</value>

<value>
  <struct>
    <member>
      <name>NAMEDEVICE</name>
      <value><string>C2000-K</string></value>
    </member>
    <member>
      <name>TYPE</name>
      <value><int>7</int></value>
    </member>
    <member>
      <name>VERSION</name>
      <value><int>105</int></value>
    </member>
    <member>
      <name>NAME</name>
      <value><int>7</int></value>
    </member>
    <member>
      <name>ADDRESS</name>
    </member>
  </struct>
  <struct>
    <member>
      <name>ADDRDEVICE</name>
      <value><int>66</int></value>
    </member>
    <member>
      <name>ADDRPORT</name>
      <value><int>1</int></value>
    </member>
  </struct>
</value>

  </member>
  <member>
    <name>STATE</name>
    <value><int>149</int></value>
  </member>
  <member>
    <name>NAMESTATE</name>
    <value><string>ТРЕБОГ А ВЗЛЮМА</string></value>
  </member>
</struct>
</value>

</data>
</array>
</value>

  </member>

```

```

    </struct>
  </value>

  </member>
</struct>
</value>

</param>

</params>

</methodResponse>
Request Server{or 14.08.2006 10:55:08}:<?xml version="1.0" encoding="Windows-1251" ?>

<methodCall>

  <methodName>GETLISTDEVICE</methodName>

  <params>

    <param>

      <value>
        <struct>
          <member>
            <name>IPSERVER</name>
            <value><string>192.168.10.109</string></value>
          </member>
          <member>
            <name>PORTSERVER</name>
            <value><int>8090</int></value>
          </member>
          <member>
            <name>STATE</name>
            <value><int>0</int></value>
          </member>
          <member>
            <name>GUID</name>
            <value><string>{16A0093F-986A-4E83-BE00-69A248F00AE9}</string></value>
          </member>
        </struct>
      </value>

    </param>

  </params>

</methodCall>

```

```

Response Client{or 14.08.2006 11:00:16}:<?xml version="1.0" encoding="Windows-1251" ?>

<methodResponse>

  <params>

    <param>

      <value>
        <struct>
          <member>
            <name>RESULT</name>
            <value><string>METHOD IS EXECUTE</string></value>
          </member>
          <member>
            <name>RESULTDATA</name>

```

```

<value>
<struct>
  <member>
    <name>NAMERESULT</name>
    <value><string>List of devices</string></value>
  </member>
  <member>
    <name>METHODNAME</name>
    <value><string>GETLISTDEVICE</string></value>
  </member>
  <member>
    <name>COUNTDEVICE</name>
    <value><int>12</int></value>
  </member>
  <member>
    <name>DEVICES</name>
  </member>
</struct>
</value>
<array>
  <data>
    <value>
      <struct>
        <member>
          <name>NAMEDEVICE</name>
          <value><string>СИГНАЛ-20П</string></value>
        </member>
        <member>
          <name>TYPE</name>
          <value><int>2</int></value>
        </member>
        <member>
          <name>VERSION</name>
          <value><int>124</int></value>
        </member>
        <member>
          <name>NAME</name>
          <value><int>2</int></value>
        </member>
        <member>
          <name>ADDRESS</name>
        </member>
      </struct>
    </value>

    </member>
    <member>
      <name>STATE</name>
      <value><int>250</int></value>
    </member>
    <member>
      <name>NAMESTATE</name>
      <value><string>HET КОНТАКТА</string></value>
    </member>
  </struct>
</value>

<value>
<struct>
  <member>
    <name>NAMEDEVICE</name>

```

```

    <value><string>СИГНАЛ-20П</string></value>
  </member>
  <member>
    <name>TYPE</name>
    <value><int>2</int></value>
  </member>
  <member>
    <name>VERSION</name>
    <value><int>167</int></value>
  </member>
  <member>
    <name>NAME</name>
    <value><int>2</int></value>
  </member>
  <member>
    <name>ADDRESS</name>
  <value>
    <struct>
      <member>
        <name>ADDRDEVICE</name>
        <value><int>2</int></value>
      </member>
      <member>
        <name>ADDRPORT</name>
        <value><int>1</int></value>
      </member>
    </struct>
  </value>

  </member>
  <member>
    <name>STATE</name>
    <value><int>149</int></value>
  </member>
  <member>
    <name>NAMESTATE</name>
    <value><string>ТРЕБОГ А ВЗЛЮМА</string></value>
  </member>
</struct>
</value>

<value>
  <struct>
    <member>
      <name>NAMEDEVICE</name>
      <value><string>C2000</string></value>
    </member>
    <member>
      <name>TYPE</name>
      <value><int>0</int></value>
    </member>
    <member>
      <name>VERSION</name>
      <value><int>201</int></value>
    </member>
    <member>
      <name>NAME</name>
      <value><int>0</int></value>
    </member>
    <member>
      <name>ADDRESS</name>
    <value>
      <struct>
        <member>
          <name>ADDRDEVICE</name>
          <value><int>4</int></value>
        </member>
        <member>

```

```

    <name>ADDRPORT</name>
    <value><int>1</int></value>
  </member>
</struct>
</value>

  </member>
  <member>
    <name>STATE</name>
    <value><int>152</int></value>
  </member>
  <member>
    <name>NAMESTATE</name>
    <value><string>HOPMA</string></value>
  </member>
</struct>
</value>

<value>
  <struct>
    <member>
      <name>NAMEDEVICE</name>
      <value><string>C2000-2</string></value>
    </member>
    <member>
      <name>TYPE</name>
      <value><int>16</int></value>
    </member>
    <member>
      <name>VERSION</name>
      <value><int>102</int></value>
    </member>
    <member>
      <name>NAME</name>
      <value><int>16</int></value>
    </member>
    <member>
      <name>ADDRESS</name>
    </member>
  </struct>
</value>

  <struct>
    <member>
      <name>ADDRDEVICE</name>
      <value><int>10</int></value>
    </member>
    <member>
      <name>ADDRPORT</name>
      <value><int>1</int></value>
    </member>
  </struct>
</value>

  </member>
  <member>
    <name>STATE</name>
    <value><int>149</int></value>
  </member>
  <member>
    <name>NAMESTATE</name>
    <value><string>TPEBOГ A B3JIOMA</string></value>
  </member>
</struct>
</value>

<value>
  <struct>
    <member>
      <name>NAMEDEVICE</name>
      <value><string>C2000-2</string></value>
    </member>
  </struct>
</value>

```

```

</member>
<member>
  <name>TYPE</name>
  <value><int>16</int></value>
</member>
<member>
  <name>VERSION</name>
  <value><int>104</int></value>
</member>
<member>
  <name>NAME</name>
  <value><int>16</int></value>
</member>
<member>
  <name>ADDRESS</name>
<value>
  <struct>
    <member>
      <name>ADDRDEVICE</name>
      <value><int>24</int></value>
    </member>
    <member>
      <name>ADDRPORT</name>
      <value><int>1</int></value>
    </member>
  </struct>
</value>

  </member>
<member>
  <name>STATE</name>
  <value><int>149</int></value>
</member>
<member>
  <name>NAMESTATE</name>
  <value><string>ТРЕБОГА БЗЈИОМА</string></value>
</member>
</struct>
</value>

<value>
  <struct>
    <member>
      <name>NAMEDEVICE</name>
      <value><string>C2000-4</string></value>
    </member>
    <member>
      <name>TYPE</name>
      <value><int>4</int></value>
    </member>
    <member>
      <name>VERSION</name>
      <value><int>106</int></value>
    </member>
    <member>
      <name>NAME</name>
      <value><int>4</int></value>
    </member>
    <member>
      <name>ADDRESS</name>
<value>
  <struct>
    <member>
      <name>ADDRDEVICE</name>
      <value><int>30</int></value>
    </member>
    <member>
      <name>ADDRPORT</name>

```



```

    <value><int>1</int></value>
  </member>
</struct>
</value>

</member>
<member>
  <name>STATE</name>
  <value><int>149</int></value>
</member>
<member>
  <name>NAMESTATE</name>
  <value><string>ТРЕБОГА ВЪЗЛОМА</string></value>
</member>
</struct>
</value>

<value>
  <struct>
    <member>
      <name>NAMEDEVICE</name>
      <value><string>C2000-СН1</string></value>
    </member>
    <member>
      <name>TYPE</name>
      <value><int>3</int></value>
    </member>
    <member>
      <name>VERSION</name>
      <value><int>120</int></value>
    </member>
    <member>
      <name>NAME</name>
      <value><int>3</int></value>
    </member>
    <member>
      <name>ADDRESS</name>
    </member>
  </struct>
</value>

  <struct>
    <member>
      <name>ADDRDEVICE</name>
      <value><int>32</int></value>
    </member>
    <member>
      <name>ADDRPORT</name>
      <value><int>1</int></value>
    </member>
  </struct>
</value>

</member>
<member>
  <name>STATE</name>
  <value><int>149</int></value>
</member>
<member>
  <name>NAMESTATE</name>
  <value><string>ТРЕБОГА ВЪЗЛОМА</string></value>
</member>
</struct>
</value>

<value>
  <struct>
    <member>
      <name>NAMEDEVICE</name>
      <value><string>C2000-БН</string></value>
    </member>
  </struct>

```

```

<member>
  <name>TYPE</name>
  <value><int>10</int></value>
</member>
<member>
  <name>VERSION</name>
  <value><int>105</int></value>
</member>
<member>
  <name>NAME</name>
  <value><int>10</int></value>
</member>
<member>
  <name>ADDRESS</name>
</member>
<value>
<struct>
  <member>
    <name>ADDRDEVICE</name>
    <value><int>34</int></value>
  </member>
  <member>
    <name>ADDRPORT</name>
    <value><int>1</int></value>
  </member>
</struct>
</value>

  </member>
  <member>
    <name>STATE</name>
    <value><int>152</int></value>
  </member>
  <member>
    <name>NAMESTATE</name>
    <value><string>HOPMA</string></value>
  </member>
</struct>
</value>

<value>
<struct>
  <member>
    <name>NAMEDEVICE</name>
    <value><string>C2000-2</string></value>
  </member>
  <member>
    <name>TYPE</name>
    <value><int>16</int></value>
  </member>
  <member>
    <name>VERSION</name>
    <value><int>105</int></value>
  </member>
  <member>
    <name>NAME</name>
    <value><int>16</int></value>
  </member>
  <member>
    <name>ADDRESS</name>
  </member>
</struct>
</value>
<struct>
  <member>
    <name>ADDRDEVICE</name>
    <value><int>35</int></value>
  </member>
  <member>
    <name>ADDRPORT</name>
    <value><int>1</int></value>
  </member>

```

```

    </member>
  </struct>
</value>

  </member>
  <member>
    <name>STATE</name>
    <value><int>149</int></value>
  </member>
  <member>
    <name>NAMESTATE</name>
    <value><string>ТРЕБОГ А ВЗЛЮМА</string></value>
  </member>
</struct>
</value>

<value>
  <struct>
    <member>
      <name>NAMEDEVICE</name>
      <value><string>C2000-4</string></value>
    </member>
    <member>
      <name>TYPE</name>
      <value><int>4</int></value>
    </member>
    <member>
      <name>VERSION</name>
      <value><int>202</int></value>
    </member>
    <member>
      <name>NAME</name>
      <value><int>4</int></value>
    </member>
    <member>
      <name>ADDRESS</name>
    </member>
  </struct>
</value>

  <struct>
    <member>
      <name>ADDRDEVICE</name>
      <value><int>50</int></value>
    </member>
    <member>
      <name>ADDRPORT</name>
      <value><int>1</int></value>
    </member>
  </struct>
</value>

  </member>
  <member>
    <name>STATE</name>
    <value><int>149</int></value>
  </member>
  <member>
    <name>NAMESTATE</name>
    <value><string>ТРЕБОГ А ВЗЛЮМА</string></value>
  </member>
</struct>
</value>

<value>
  <struct>
    <member>
      <name>NAMEDEVICE</name>
      <value><string>C2000-KC</string></value>
    </member>
    <member>

```

```

    <name>TYPE</name>
    <value><int>13</int></value>
  </member>
  <member>
    <name>VERSION</name>
    <value><int>103</int></value>
  </member>
  <member>
    <name>NAME</name>
    <value><int>13</int></value>
  </member>
  <member>
    <name>ADDRESS</name>
  <value>
    <struct>
      <member>
        <name>ADDRDEVICE</name>
        <value><int>63</int></value>
      </member>
      <member>
        <name>ADDRPORT</name>
        <value><int>1</int></value>
      </member>
    </struct>
  </value>

  </member>
  <member>
    <name>STATE</name>
    <value><int>250</int></value>
  </member>
  <member>
    <name>NAMESTATE</name>
    <value><string>HET KOHTAKTA</string></value>
  </member>
</struct>
</value>

<value>
  <struct>
    <member>
      <name>NAMEDEVICE</name>
      <value><string>C2000-K</string></value>
    </member>
    <member>
      <name>TYPE</name>
      <value><int>7</int></value>
    </member>
    <member>
      <name>VERSION</name>
      <value><int>105</int></value>
    </member>
    <member>
      <name>NAME</name>
      <value><int>7</int></value>
    </member>
    <member>
      <name>ADDRESS</name>
  <value>
    <struct>
      <member>
        <name>ADDRDEVICE</name>
        <value><int>66</int></value>
      </member>
      <member>
        <name>ADDRPORT</name>
        <value><int>1</int></value>
      </member>
    </struct>
  </value>

```

```

</struct>
</value>

</member>
<member>
  <name>STATE</name>
  <value><int>149</int></value>
</member>
<member>
  <name>NAMESTATE</name>
  <value><string>ТРЕВОГА ВЗЛЮМА</string></value>
</member>
</struct>
</value>

</data>
</array>
</value>

</member>
</struct>
</value>

</member>
</struct>
</value>

</param>
</params>

</methodResponse>

```

Для управления взятием/снятием различных объектов охранной системы, таких как: зона, направление пожаротушения, раздел, группа разделов в модуле опубликован метод ControlObjects.

Параметры метода ControlObjects:

- **STRING GUID** – жетон безопасности.
- **ARRAY OBJECTS** – список объектов которыми необходимо управлять, элемент списка представляет из себя структуру данных:
 - **STRING TYPEOBJECT** – тип объекта которыми необходимо управлять:
 - «RZD» - раздел;
 - «GROUPEPRZD» - группа разделов;
 - «ZONE» - зона охраны.
 - **STRING CONTROLWORD** – ключевое слово управления объектом:
 - «ARM» - взять под охрану объект;
 - «DISARM» - снять с охраны объект;
 - «OPEN» - предоставить доступ для точки доступа.
 - **INT64 INDEXOBJECT** – индекс(номер) объекта, необходим при управлении разделом, направлением пожаротушения, группой разделов;
 - **STRUCT ADDRESS** – структура данных определяющая адрес зоны которыми необходимо управлять, содержит следующие поля:

- **INT64 ADDRELEMENT** – адрес элемента (номер шлейфа), зависит от типа устройства;
- **INT64 ADDRDEVICE** – адрес устройства по RS 485;
- **INT64 ADDRPULT** – адрес пульта по RS 232;
- **INT64 ADDRPORT** – номер порта, к которому подключен интерфейс.

На запрос метода модуль выдает стандартный ответ об исполнении или об ошибке.

Пример взятия всех зон охрана у прибора «Сигнал 20П» с адресом 2.

Request Server(от 14.08.2006 13:35:06):<?xml version="1.0" encoding="Windows-1251" ?>

```
<methodCall>

  <methodName>CONTROLOBJECTS</methodName>

  <params>

    <param>

      <value>
        <struct>
          <member>
            <name>GUID</name>
            <value><string>{F728B45F-4CEF-4691-82DC-514F4ED9D13D}</string></value>
          </member>
          <member>
            <name>OBJECTS</name>
          </member>
        </struct>
        <array>
          <data>
            <value>
              <struct>
                <member>
                  <name>TYPEOBJECT</name>
                  <value><string>ZONE</string></value>
                </member>
                <member>
                  <name>CONTROLWORD</name>
                  <value><string>ARM</string></value>
                </member>
                <member>
                  <name>ADDRESS</name>
                </member>
              </struct>
              <struct>
                <member>
                  <name>ADDRELEMENT</name>
                  <value><int>1</int></value>
                </member>
                <member>
                  <name>ADDRDEVICE</name>
                  <value><int>2</int></value>
                </member>
                <member>
                  <name>ADDRPORT</name>
                  <value><int>1</int></value>
                </member>
              </struct>
            </value>
          </data>
        </array>
      </value>

      </param>
    </params>
  </methodCall>
```

```

<struct>
  <member>
    <name>TYPEOBJECT</name>
    <value><string>ZONE</string></value>
  </member>
  <member>
    <name>CONTROLWORD</name>
    <value><string>ARM</string></value>
  </member>
  <member>
    <name>ADDRESS</name>
<value>
  <struct>
    <member>
      <name>ADDRELEMENT</name>
      <value><int>2</int></value>
    </member>
    <member>
      <name>ADDRDEVICE</name>
      <value><int>2</int></value>
    </member>
    <member>
      <name>ADDRPORT</name>
      <value><int>1</int></value>
    </member>
  </struct>
</value>

  </member>
</struct>
</value>

<value>
  <struct>
    <member>
      <name>TYPEOBJECT</name>
      <value><string>ZONE</string></value>
    </member>
    <member>
      <name>CONTROLWORD</name>
      <value><string>ARM</string></value>
    </member>
    <member>
      <name>ADDRESS</name>
<value>
  <struct>
    <member>
      <name>ADDRELEMENT</name>
      <value><int>3</int></value>
    </member>
    <member>
      <name>ADDRDEVICE</name>
      <value><int>2</int></value>
    </member>
    <member>
      <name>ADDRPORT</name>
      <value><int>1</int></value>
    </member>
  </struct>
</value>

  </member>
</struct>
</value>

<value>
  <struct>
    <member>

```

```

    <name>TYPEOBJECT</name>
    <value><string>ZONE</string></value>
  </member>
  <member>
    <name>CONTROLWORD</name>
    <value><string>ARM</string></value>
  </member>
  <member>
    <name>ADDRESS</name>
  <value>
    <struct>
      <member>
        <name>ADDRELEMENT</name>
        <value><int>4</int></value>
      </member>
      <member>
        <name>ADDRDEVICE</name>
        <value><int>2</int></value>
      </member>
      <member>
        <name>ADDRPORT</name>
        <value><int>1</int></value>
      </member>
    </struct>
  </value>

  </member>
</struct>
</value>

<value>
  <struct>
    <member>
      <name>TYPEOBJECT</name>
      <value><string>ZONE</string></value>
    </member>
    <member>
      <name>CONTROLWORD</name>
      <value><string>ARM</string></value>
    </member>
    <member>
      <name>ADDRESS</name>
  <value>
    <struct>
      <member>
        <name>ADDRELEMENT</name>
        <value><int>5</int></value>
      </member>
      <member>
        <name>ADDRDEVICE</name>
        <value><int>2</int></value>
      </member>
      <member>
        <name>ADDRPORT</name>
        <value><int>1</int></value>
      </member>
    </struct>
  </value>

  </member>
</struct>
</value>

<value>
  <struct>
    <member>
      <name>TYPEOBJECT</name>
      <value><string>ZONE</string></value>

```



```

</member>
<member>
  <name>CONTROLWORD</name>
  <value><string>ARM</string></value>
</member>
<member>
  <name>ADDRESS</name>
<value>
  <struct>
    <member>
      <name>ADDRELEMENT</name>
      <value><int>6</int></value>
    </member>
    <member>
      <name>ADDRDEVICE</name>
      <value><int>2</int></value>
    </member>
    <member>
      <name>ADDRPORT</name>
      <value><int>1</int></value>
    </member>
  </struct>
</value>

  </member>
</struct>
</value>

<value>
  <struct>
    <member>
      <name>TYPEOBJECT</name>
      <value><string>ZONE</string></value>
    </member>
    <member>
      <name>CONTROLWORD</name>
      <value><string>ARM</string></value>
    </member>
    <member>
      <name>ADDRESS</name>
<value>
  <struct>
    <member>
      <name>ADDRELEMENT</name>
      <value><int>7</int></value>
    </member>
    <member>
      <name>ADDRDEVICE</name>
      <value><int>2</int></value>
    </member>
    <member>
      <name>ADDRPORT</name>
      <value><int>1</int></value>
    </member>
  </struct>
</value>

  </member>
</struct>
</value>

<value>
  <struct>
    <member>
      <name>TYPEOBJECT</name>
      <value><string>ZONE</string></value>
    </member>
    <member>

```

```

    <name>CONTROLWORD</name>
    <value><string>ARM</string></value>
  </member>
  <member>
    <name>ADDRESS</name>
  <value>
    <struct>
      <member>
        <name>ADDRELEMENT</name>
        <value><int>8</int></value>
      </member>
      <member>
        <name>ADDRDEVICE</name>
        <value><int>2</int></value>
      </member>
      <member>
        <name>ADDRPORT</name>
        <value><int>1</int></value>
      </member>
    </struct>
  </value>

  </member>
</struct>
</value>

<value>
  <struct>
    <member>
      <name>TYPEOBJECT</name>
      <value><string>ZONE</string></value>
    </member>
    <member>
      <name>CONTROLWORD</name>
      <value><string>ARM</string></value>
    </member>
    <member>
      <name>ADDRESS</name>
    <value>
      <struct>
        <member>
          <name>ADDRELEMENT</name>
          <value><int>9</int></value>
        </member>
        <member>
          <name>ADDRDEVICE</name>
          <value><int>2</int></value>
        </member>
        <member>
          <name>ADDRPORT</name>
          <value><int>1</int></value>
        </member>
      </struct>
    </value>

    </member>
  </struct>
</value>

<value>
  <struct>
    <member>
      <name>TYPEOBJECT</name>
      <value><string>ZONE</string></value>
    </member>
    <member>
      <name>CONTROLWORD</name>
      <value><string>ARM</string></value>

```

```

    </member>
    <member>
      <name>ADDRESS</name>
    <value>
      <struct>
        <member>
          <name>ADDRELEMENT</name>
          <value><int>10</int></value>
        </member>
        <member>
          <name>ADDRDEVICE</name>
          <value><int>2</int></value>
        </member>
        <member>
          <name>ADDRPORT</name>
          <value><int>1</int></value>
        </member>
      </struct>
    </value>

    </member>
  </struct>
</value>

<value>
  <struct>
    <member>
      <name>TYPEOBJECT</name>
      <value><string>ZONE</string></value>
    </member>
    <member>
      <name>CONTROLWORD</name>
      <value><string>ARM</string></value>
    </member>
    <member>
      <name>ADDRESS</name>
    <value>
      <struct>
        <member>
          <name>ADDRELEMENT</name>
          <value><int>11</int></value>
        </member>
        <member>
          <name>ADDRDEVICE</name>
          <value><int>2</int></value>
        </member>
        <member>
          <name>ADDRPORT</name>
          <value><int>1</int></value>
        </member>
      </struct>
    </value>

    </member>
  </struct>
</value>

<value>
  <struct>
    <member>
      <name>TYPEOBJECT</name>
      <value><string>ZONE</string></value>
    </member>
    <member>
      <name>CONTROLWORD</name>
      <value><string>ARM</string></value>
    </member>
    <member>

```

```

    <name>ADDRESS</name>
<value>
<struct>
  <member>
    <name>ADDRELEMENT</name>
    <value><int>12</int></value>
  </member>
  <member>
    <name>ADDRDEVICE</name>
    <value><int>2</int></value>
  </member>
  <member>
    <name>ADDRPORT</name>
    <value><int>1</int></value>
  </member>
</struct>
</value>

  </member>
</struct>
</value>

<value>
<struct>
  <member>
    <name>TYPEOBJECT</name>
    <value><string>ZONE</string></value>
  </member>
  <member>
    <name>CONTROLWORD</name>
    <value><string>ARM</string></value>
  </member>
  <member>
    <name>ADDRESS</name>
<value>
<struct>
  <member>
    <name>ADDRELEMENT</name>
    <value><int>13</int></value>
  </member>
  <member>
    <name>ADDRDEVICE</name>
    <value><int>2</int></value>
  </member>
  <member>
    <name>ADDRPORT</name>
    <value><int>1</int></value>
  </member>
</struct>
</value>

  </member>
</struct>
</value>

<value>
<struct>
  <member>
    <name>TYPEOBJECT</name>
    <value><string>ZONE</string></value>
  </member>
  <member>
    <name>CONTROLWORD</name>
    <value><string>ARM</string></value>
  </member>
  <member>
    <name>ADDRESS</name>
<value>

```

```

<struct>
  <member>
    <name>ADDRELEMENT</name>
    <value><int>14</int></value>
  </member>
  <member>
    <name>ADDRDEVICE</name>
    <value><int>2</int></value>
  </member>
  <member>
    <name>ADDRPORT</name>
    <value><int>1</int></value>
  </member>
</struct>
</value>

  </member>
</struct>
</value>

<value>
<struct>
  <member>
    <name>TYPEOBJECT</name>
    <value><string>ZONE</string></value>
  </member>
  <member>
    <name>CONTROLWORD</name>
    <value><string>ARM</string></value>
  </member>
  <member>
    <name>ADDRESS</name>
<value>
<struct>
  <member>
    <name>ADDRELEMENT</name>
    <value><int>15</int></value>
  </member>
  <member>
    <name>ADDRDEVICE</name>
    <value><int>2</int></value>
  </member>
  <member>
    <name>ADDRPORT</name>
    <value><int>1</int></value>
  </member>
</struct>
</value>

  </member>
</struct>
</value>

<value>
<struct>
  <member>
    <name>TYPEOBJECT</name>
    <value><string>ZONE</string></value>
  </member>
  <member>
    <name>CONTROLWORD</name>
    <value><string>ARM</string></value>
  </member>
  <member>
    <name>ADDRESS</name>
<value>
<struct>
  <member>

```

```

    <name>ADDRELEMENT</name>
    <value><int>16</int></value>
  </member>
  <member>
    <name>ADDRDEVICE</name>
    <value><int>2</int></value>
  </member>
  <member>
    <name>ADDRPORT</name>
    <value><int>1</int></value>
  </member>
</struct>
</value>

</member>
</struct>
</value>

<value>
<struct>
  <member>
    <name>TYPEOBJECT</name>
    <value><string>ZONE</string></value>
  </member>
  <member>
    <name>CONTROLWORD</name>
    <value><string>ARM</string></value>
  </member>
  <member>
    <name>ADDRESS</name>
<value>
<struct>
  <member>
    <name>ADDRELEMENT</name>
    <value><int>17</int></value>
  </member>
  <member>
    <name>ADDRDEVICE</name>
    <value><int>2</int></value>
  </member>
  <member>
    <name>ADDRPORT</name>
    <value><int>1</int></value>
  </member>
</struct>
</value>

  </member>
</struct>
</value>

<value>
<struct>
  <member>
    <name>TYPEOBJECT</name>
    <value><string>ZONE</string></value>
  </member>
  <member>
    <name>CONTROLWORD</name>
    <value><string>ARM</string></value>
  </member>
  <member>
    <name>ADDRESS</name>
<value>
<struct>
  <member>
    <name>ADDRELEMENT</name>
    <value><int>18</int></value>

```

```

</member>
<member>
  <name>ADDRDEVICE</name>
  <value><int>2</int></value>
</member>
<member>
  <name>ADDRPORT</name>
  <value><int>1</int></value>
</member>
</struct>
</value>

</member>
</struct>
</value>

<value>
<struct>
  <member>
    <name>TYPEOBJECT</name>
    <value><string>ZONE</string></value>
  </member>
  <member>
    <name>CONTROLWORD</name>
    <value><string>ARM</string></value>
  </member>
  <member>
    <name>ADDRESS</name>
<value>
<struct>
  <member>
    <name>ADDRELEMENT</name>
    <value><int>19</int></value>
  </member>
  <member>
    <name>ADDRDEVICE</name>
    <value><int>2</int></value>
  </member>
  <member>
    <name>ADDRPORT</name>
    <value><int>1</int></value>
  </member>
</struct>
</value>

</member>
</struct>
</value>

<value>
<struct>
  <member>
    <name>TYPEOBJECT</name>
    <value><string>ZONE</string></value>
  </member>
  <member>
    <name>CONTROLWORD</name>
    <value><string>ARM</string></value>
  </member>
  <member>
    <name>ADDRESS</name>
<value>
<struct>
  <member>
    <name>ADDRELEMENT</name>
    <value><int>20</int></value>
  </member>
  <member>

```

```

    <name>ADDRDEVICE</name>
    <value><int>2</int></value>
  </member>
  <member>
    <name>ADDRPORT</name>
    <value><int>1</int></value>
  </member>
</struct>
</value>

</member>
</struct>
</value>

</data>
</array>
</value>

</member>
</struct>
</value>

</param>

</params>
</methodCall>

Response Client{or 14.08.2006 13:35:06}:<?xml version="1.0" encoding="Windows-1251" ?>

<methodResponse>

  <params>

    <param>

      <value>
        <struct>
          <member>
            <name>RESULT</name>
            <value><string>METHOD IS EXECUTE</string></value>
          </member>
        </struct>
      </value>

    </param>

  </params>
</methodResponse>

```

Одной из основных функций модуля управления, является функция контроля изменения состояния объекта. Контроль изменения состояния прибора или его агрегата (охранной зоны, считывателя, датчика контролирующего взлом корпуса прибора и т.д.) осуществляется с помощью механизма циклического опроса всех подключенных приборов. При этом драйвер RS делает специальный запрос прибору о возникновении события, на этот запрос прибор отвечает посылкой ответа об «отсутствии событий» или передает данные самого позднего события. Модуль опроса, получив данные, формирует телеграмму о возникшем событии и ретранслирует на все «подписанные» XML-RPC сервера, посредством удаленного вызова процедуры ONRSEVENT.

В таблице приведены все исходящие вызовы, обязательные для регистрации в клиентской программе при работе с модулем управления.

№	Наименование	Описание события
1	OnChangeStateElements	Об изменение состоянии агрегатов (зон, выходов, реле, приборов и т.д.). Модуль контролирует состояние объектов и оповещает клиентов об их изменении.
2	OnRsEvent	Считанные, с RS интерфейса данные, а также события сформированные модулем опроса.
3	OnInitiative	Инициатива централизованного управления, событие инициирующее механизм управления объектами охраны с приборов С2000-К, С2000-2, С2000-4, С2000-БИ, С2000-КС.
4	OnNewDevice	Обнаружения нового устройства или изменения состояния устройства.
5	OnResultCommand	Возвращает результат послыки асинхронной команды.

Параметры метода OnRsEvent:

- **STRING GUID** – жетон безопасности;
- **STRUCT DATAEVENTS** – структура данных у события;
 - **DOUBLE DATETIME** – время и дата возникновения события;
 - **STRUCT ADDRESS** – структура данных определяющая адрес объекта в системе, содержит следующие поля:
 - **INT64 ADDRELEMENT** – адрес элемента (номер шлейфа), зависит от типа устройства;
 - **INT64 ADDRDEVICE** – адрес устройства по RS 485;
 - **INT64 ADDRPUULT** – адрес пульта по RS 232;
 - **INT64 ADDRPORT** – номер порта, к которому подключен интерфейс.
 - **INT64 TYPE** - тип объекта:
 - 2 – раздел;
 - 3 – шлейф;
 - 4 – устройство;
 - 9 – реле (выход);
 - 11 – зона доступа;
 - 12 – точка доступа;
 - 16 - порт;
 - 17 – группа разделов.
 - **INT64 NAME TYPE** – наименование типа объекта;
 - **INT64 EVENT** – код события;

- **STRING NAMEEVENT** – наименование события;
- **STRING RPCNAMETYPEEVENT** – тип события:
 - «NEWEVENT» - новое событие;
 - «OLDEVENT» - старое событие вычитанное из устройства позднее двух минут от момента возникновения;
 - «INITIATIVE» - инициатива управления;
 - «UNKNOWNEVENT» - неизвестный тип события;
- **INT64 IDPERSON** – идентификатор человека к которому относится событие;
- **STRUCT EXTENDEDATA** – структура расширенных данных:
 - **STRING STRINGPASSWORD** – код пароля в строковом варианте;
 - **STRING RPCTYPEPASSWORD** – тип пароля (число):
 - 0 – код ключа в формате PIN код из двух байт;
 - 1 – код ключа в формате DALLAS TOUCH MEMMORY из восьми байт.
 - **STRING NAMETYPEPASSWORD** – тип пароля (строка):
 - «PIN» - – код ключа в формате PIN;
 - «TOUCH» - код ключа в формате DALLAS TOUCH MEMMORY;
 - **ARRAY RPCPASSWORD** – побайтный список в коде ключа, каждый элемент- байт кода ключа;
 - **INT64 ZONEACCESS** – индекс(номер) зоны доступа для события.

Пример события:

Response Server(от 14.08.2006 17:58:10)<?xml version="1.0" encoding="Windows-1251" ?>

```
<methodCall>

<methodName>ONRSEVENT</methodName>

<params>

<param>

<value>
<struct>
  <member>
    <name>GUID</name>
    <value><string>{114AA6E1-DAA1-429E-AF35-97FEAD352872}</string></value>
  </member>
  <member>
    <name>DATAEVENTS</name>
  </member>
</struct>
<value>
<struct>
  <member>
    <name>DATETIME</name>
    <value><double>38943,7487314005</double></value>
  </member>
  <member>
    <name>ADDRESS</name>
  </member>
</struct>
<value>
<struct>
  <member>
    <name>ADDRELEMENT</name>
    <value><int>2</int></value>
  </member>
  <member>
    <name>ADDRDEVICE</name>
    <value><int>24</int></value>
  </member>
</struct>
</value>
</param>
</params>
</methodCall>
```

```

</member>
<member>
  <name>ADDRPORT</name>
  <value><int>1</int></value>
</member>
</struct>
</value>

</member>
<member>
  <name>TYPE</name>
  <value><int>8</int></value>
</member>
<member>
  <name>NAMETYPE</name>
  <value><string>СЧИТЫВАТЕЛЬ</string></value>
</member>
<member>
  <name>EVENT</name>
  <value><int>28</int></value>
</member>
<member>
  <name>NAMEEVENT</name>
  <value><string>Доступ предоставлен</string></value>
</member>
<member>
  <name>TYPEEVENT</name>
  <value><int>0</int></value>
</member>
<member>
  <name>NAMETYPEEVENT</name>
  <value><string>NEWEVENT</string></value>
</member>
<member>
  <name>EXTENDEDDEDATA</name>
<value>
  <struct>
    <member>
      <name>STRINGPASSWORD</name>
      <value><string>9E00000287DD8501</string></value>
    </member>
    <member>
      <name>TYPEPASSWORD</name>
      <value><int>1</int></value>
    </member>
    <member>
      <name>NAMETYPEPASSWORD</name>
      <value><string>TOUCH</string></value>
    </member>
    <member>
      <name>PASSWORD</name>
<value>
  <array>
    <data>
      <value><int>1</int></value>
      <value><int>133</int></value>
      <value><int>221</int></value>
      <value><int>135</int></value>
      <value><int>2</int></value>
      <value><int>0</int></value>
      <value><int>0</int></value>
      <value><int>158</int></value>
    </data>
  </array>
</value>

</member>
<member>

```

```

        <name>ZONEACCESS</name>
        <value><int>0</int></value>
    </member>
</struct>
</value>

    </member>
</struct>
</value>

    </member>
</struct>
</value>

</param>

</params>

</methodCall>

```

При обнаружении нового устройства на RS интерфейсе, или изменения параметров устройства, или его состояния модуль формирует событие ONNEWDEVICE.

Параметры метода OnNewDevice:

- **STRING GUID** – жетон безопасности;
- **STRUCT ADDRESS** – структура данных определяющая адрес прибора в системе, содержит следующие поля:
 - **INT64 ADDRDEVICE** – адрес устройства по RS 485;
 - **INT64 ADDRPULT** – адрес пульта по RS 232;
 - **INT64 ADDRPORT** – номер порта, к которому подключен интерфейс.
- **STRING NAMEDEVICE** – строковая константа определяющая наименование типа подключенного устройства (см. таблицу 1 в приложении А);
- **INT64 TYPE** – цифровая константа типа устройства (см. таблицу 1 в приложении А);
 - **INT64 VERSION** – версия устройства;
- **ARRAY ZONES** – список зон устройства, элемент представляет из себя структуру данных:
 - **STRUCT ADDRESS** – структура данных определяющая адрес зоны которыми необходимо управлять, содержит следующие поля:
 - **INT64 ADDRELEMENT** – адрес элемента (номер шлейфа), зависит от типа устройства;
 - **INT64 ADDRDEVICE** – адрес устройства по RS 485;
 - **INT64 ADDRPULT** – адрес пульта по RS 232;
 - **INT64 ADDRPORT** – номер порта, к которому подключен интерфейс.
 - **INT64 TYPE** – константа типа зоны:
 - 3-шлейф сигнализации;
 - 8 - считыватель;
 - 9 – выход(реле).
 - **STRING NAMETYPE** – наименование типа зоны;

«ШЛЕЙФ» - шлейф сигнализации;
«СЧИТЫВАТЕЛЬ» - считыватель;
«ВЫХОД» – выход (реле).

Пример события ONNEWDEVICE:

```
<?xml version="1.0" encoding="Windows-1251" ?>
```

```
<methodCall>

  <methodName>ONNEWDEVICE</methodName>

  <params>

    <param>

      <value>
        <struct>
          <member>
            <name>GUID</name>
            <value><string>{77FEC8C3-4689-414D-A6E9-5FE1BCADDEA9}</string></value>
          </member>
          <member>
            <name>ADDRESS</name>
          </member>
        </struct>
      </value>

      <struct>
        <member>
          <name>ADDRDEVICE</name>
          <value><int>10</int></value>
        </member>
        <member>
          <name>ADDRPORT</name>
          <value><int>1</int></value>
        </member>
      </struct>
    </value>

    </member>
    <member>
      <name>NAMETYPE</name>
      <value><string>ПРИБОР</string></value>
    </member>
    <member>
      <name>TYPE</name>
      <value><int>16</int></value>
    </member>
    <member>
      <name>NAMEDEVICE</name>
      <value><string>C2000-2</string></value>
    </member>
    <member>
      <name>VERSION</name>
      <value><int>102</int></value>
    </member>
    <member>
      <name>ZONES</name>
    </member>
  </value>
  <array>
    <data>
      <value>
        <struct>
          <member>
            <name>ADDRESS</name>
          </member>
        </struct>
      </value>
      <struct>
        <member>
          <name>ADDRELEMENT</name>
          <value><int>1</int></value>
        </member>
      </struct>
    </data>
  </array>
</params>
</methodCall>
```

```

</member>
<member>
  <name>ADDRDEVICE</name>
  <value><int>10</int></value>
</member>
<member>
  <name>ADDRPORT</name>
  <value><int>1</int></value>
</member>
</struct>
</value>

</member>
<member>
  <name>TYPE</name>
  <value><int>3</int></value>
</member>
<member>
  <name>NAMETYPE</name>
  <value><string>ШЛЕЙФ</string></value>
</member>
</struct>
</value>

<value>
  <struct>
    <member>
      <name>ADDRESS</name>
</member>
<value>
  <struct>
    <member>
      <name>ADDRELEMENT</name>
      <value><int>2</int></value>
    </member>
    <member>
      <name>ADDRDEVICE</name>
      <value><int>10</int></value>
    </member>
    <member>
      <name>ADDRPORT</name>
      <value><int>1</int></value>
    </member>
  </struct>
</value>

</member>
<member>
  <name>TYPE</name>
  <value><int>3</int></value>
</member>
<member>
  <name>NAMETYPE</name>
  <value><string>ШЛЕЙФ</string></value>
</member>
</struct>
</value>

<value>
  <struct>
    <member>
      <name>ADDRESS</name>
</member>
<value>
  <struct>
    <member>
      <name>ADDRELEMENT</name>
      <value><int>1</int></value>
    </member>
  </struct>
</value>

```

```

    <name>ADDRDEVICE</name>
    <value><int>10</int></value>
  </member>
  <member>
    <name>ADDRPORT</name>
    <value><int>1</int></value>
  </member>
</struct>
</value>

  </member>
  <member>
    <name>TYPE</name>
    <value><int>8</int></value>
  </member>
  <member>
    <name>NAMETYPE</name>
    <value><string>СЧИТЫВАТЕЛЬ</string></value>
  </member>
</struct>
</value>

<value>
  <struct>
    <member>
      <name>ADDRESS</name>
    </member>
  </struct>
  <value>
    <struct>
      <member>
        <name>ADDRELEMENT</name>
        <value><int>2</int></value>
      </member>
      <member>
        <name>ADDRDEVICE</name>
        <value><int>10</int></value>
      </member>
      <member>
        <name>ADDRPORT</name>
        <value><int>1</int></value>
      </member>
    </struct>
  </value>

  </member>
  <member>
    <name>TYPE</name>
    <value><int>8</int></value>
  </member>
  <member>
    <name>NAMETYPE</name>
    <value><string>СЧИТЫВАТЕЛЬ</string></value>
  </member>
</struct>
</value>

<value>
  <struct>
    <member>
      <name>ADDRESS</name>
    </member>
  </struct>
  <value>
    <struct>
      <member>
        <name>ADDRELEMENT</name>
        <value><int>1</int></value>
      </member>
      <member>
        <name>ADDRDEVICE</name>
        <value><int>10</int></value>
      </member>
    </struct>
  </value>

```

```

    </member>
    <member>
      <name>ADDRPORT</name>
      <value><int>1</int></value>
    </member>
  </struct>
</value>

  </member>
  <member>
    <name>TYPE</name>
    <value><int>9</int></value>
  </member>
  <member>
    <name>NAMETYPE</name>
    <value><string>БЫХОД</string></value>
  </member>
</struct>
</value>

<value>
  <struct>
    <member>
      <name>ADDRESS</name>
    </member>
  </struct>
  <struct>
    <member>
      <name>ADDRELEMENT</name>
      <value><int>2</int></value>
    </member>
    <member>
      <name>ADDRDEVICE</name>
      <value><int>10</int></value>
    </member>
    <member>
      <name>ADDRPORT</name>
      <value><int>1</int></value>
    </member>
  </struct>
</value>

  </member>
  <member>
    <name>TYPE</name>
    <value><int>9</int></value>
  </member>
  <member>
    <name>NAMETYPE</name>
    <value><string>БЫХОД</string></value>
  </member>
</struct>
</value>

</data>
</array>
</value>

  </member>
</struct>
</value>

</param>
</params>
</methodCall>

```


При изменении состояний агрегатов (шлейф, зона, выход-реле, считыватель и т.д.) прибора модуль осуществляет удаленный вызов метода ONCHANGESTATEELEMENTS.

Параметры метода OnChangeStateElements:

- **STRING GUID** – жетон безопасности;
- **STRUCT ADDRESS** – структура данных определяющая адрес зоны которыми необходимо управлять, содержит следующие поля:
 - **INT64 ADDRELEMENT** – адрес элемента (номер шлейфа), зависит от типа устройства;
 - **INT64 ADDRDEVICE** – адрес устройства по RS 485;
 - **INT64 ADDRPUULT** – адрес пульта по RS 232;
 - **INT64 ADDRPORT** – номер порта, к которому подключен интерфейс.
- **INT64 TYPE** – константа типа зоны:
 - 3-шлейф сигнализации;
 - 4 – прибор;
 - 8 - считыватель;
 - 9 – выход (реле).
- **STRING NAMETYPE** – наименование типа зоны:
 - «ПРИБОР» - прибор;
 - «ШЛЕЙФ» - шлейф сигнализации;
 - «СЧИТЫВАТЕЛЬ» - считыватель;
 - «ВЫХОД» – выход (реле).
- **INT64 STATE** – состояние агрегата, все состояния приведены в таблице (см. таблицу 2 в приложении А);
- **STRING NAMESTATE** – наименование состояния агрегата (см. таблицу 2 в приложении А);
- **INT64 IDPERSON** – идентификатор человека к которому относится событие;
- **INT64 STATEEVENT** – если данный параметр = 1, то состояние изменилось после считывания события, в противном случае после запроса состояния.

Пример события ONCHANGESTATEELEMENTS:

```
<?xml version="1.0" encoding="Windows-1251" ?>

<methodCall>

  <methodName>ONCHANGESTATEELEMENTS</methodName>

  <params>

    <param>

      <value>
        <struct>
          <member>
            <name>GUID</name>
            <value><string>{77FEC8C3-4689-414D-A6E9-5FE1BCADDEA9}</string></value>
          </member>
```

```

    <member>
      <name>ADDRESS</name>
    </member>
  </struct>
</value>

  <member>
    <name>ADDRDEVICE</name>
    <value><int>10</int></value>
  </member>
  <member>
    <name>ADDRPORT</name>
    <value><int>1</int></value>
  </member>
</struct>
</value>

  </member>
  <member>
    <name>TYPE</name>
    <value><int>4</int></value>
  </member>
  <member>
    <name>NAMETYPE</name>
    <value><string>ПРИБОР</string></value>
  </member>
  <member>
    <name>STATE</name>
    <value><int>152</int></value>
  </member>
  <member>
    <name>NAMESTATE</name>
    <value><string>НОРМА</string></value>
  </member>
</struct>
</value>

</param>
</params>
</methodCall>

```

Модуль управления позволяет отрабатывать централизованные тактики управления доступом и взятие-снятие объектов под охрану, а также транслировать состояние объектов на индикационные приборы типа С2000-К, С2000-БИ и С2000-КС.

При этом механизм управления следующий:

- 1) при поднесении ключа к считывателю прибора, либо наборе ПИН-КОДА на клавиатуре, прибор формирует специальное сообщение - «инициатива сообщения», которое считывается RS драйвером и ретранслируется модулем управления клиентскому программному обеспечению с помощью вызова метода ONINITIATIVE.
- 2) Программное обеспечение получившее инициативу управления, должно определить полномочия управления и вызвав метод TranslateRightsObjects передать необходимые данные для отработки управляющего воздействия: предоставления или не предоставления доступа или передачи полномочий управления.

Параметры метода OnInitiative:

- **INT64 GUID** – жетон безопасности;
- **STRUCT READERS** – структура данных для считывателя с которого «пришла» инициатива:
 - **STRUCT ADDRESS** – структура данных определяющая адрес считывателя:
 - **INT64 ADDRELEMENT** – адрес элемента (номер шлейфа), зависит от типа устройства;
 - **INT64 ADDRDEVICE** – адрес устройства по RS 485;
 - **INT64 ADDRPUULT** – адрес пульта по RS 232;
 - **INT64 ADDRPORT** – номер порта, к которому подключен интерфейс.
- **STRUCT RSCOMMINITIAIVE** – структура данных инициативы управления:
 - **STRING NAME** – имя запроса;
 - **INT64 NUMBER** – номер запроса;
 - **STRUCT TYPEINITIATIVE** – структура параметров для запроса инициативы:
 - **STRING NAME** – описание параметра;
 - **INT64 NUMBER** – порядковый номер параметра;
 - **ARRAY QUERY** – список данных в параметре, элемент списка представляет из себя структуру данных:
 - **STRING NAME** – наименование данных;
 - **STRING DATA** – тип данных:
 - «DATA» - данные;
 - «CONSTANT» - заранее описанные константные значения;
 - **STRUCT DATA** - структура с данными для данного параметра запроса:
 - **STRING NAME** – описание данных;
 - **INT64 TYPEVALUE** – тип данных;
 - **INT64 LENGTHVALUE** –длина списка значений;
 - **ARRAY VALUES** – список значений для данного параметра, каждый элемент списка представляет из себя структуру в которой хранится один байт значения;
 - **INT64 VALUE** – значение элемента списка.

В таблице приведены все возможные параметры запроса инициативы управления, структуру данных в запросе можно получить по команде GetListRSCommand.

Номер	Наименование запроса	Данные	Дл.	Варианты
4	Идентификация хозоргана (вариант 1)	Дополнительные данные	1	0-не указано 1-желательное действие взятие-снятие 2-получить состояние всех разделов
		Тип ключа	1	1-PIN код 10,12-TOUCH MEMMORY
		Код ключа	2 8	Если Тип ключа=1 Если Тип ключа=10,12
14	Идентификация хозоргана (вариант 2)	Номер считывателя	1	Нет

		Дополнительные данные	1	0-не указано 1-желательное действие взятие-снятие 2-получить состояние всех разделов
7	Запрос состояния раздела (вариант 1)	Номер раздела	2	Нет
		Номер хозоргана	2	Нет
		Дополнительные данные	1	0-не указано 1-желательное действие взятие-снятие 2-получить состояние всех разделов
15	Запрос состояния раздела (вариант 2)	Номер считывателя	1	Нет
		Номер раздела	2	Нет
		Дополнительные данные	1	0-не указано 1-желательное действие взятие-снятие 2-получить состояние всех разделов
		[Код ключа] ²	8	Нет

На запросы типа 4, 5, 7, 15 необходимо передавать список объектов для управления с помощью вызова метода TranslateRightsObjects.

Получив список объектов модуль отрабатывает в соответствии с переданными ему полномочиями все тактики управления.

Ниже приведен пример управления разделом 33 с помощью инициативы управления с запросом 14.

Инициатива управления, пришедшая от прибора.

```
<?xml version="1.0" encoding="Windows-1251" ?>
<methodCall>
  <methodName>ONINITIATIVE</methodName>
  <params>
    <param>
      <value>
        <struct>
          <member>
            <name>GUID</name>
            <value><string>{3AC23F75-8A8E-43ED-A645-738F72A3A656}</string></value>
          </member>
          <member>
            <name>READERS</name>
```

² -[имя параметра] – такое обозначение говорит о том что параметр не обязательный.

```

<value>
<struct>
  <member>
    <name>ADDRESS</name>
</struct>
</value>
<struct>
  <member>
    <name>ADDRELEMENT</name>
    <value><int>1</int></value>
  </member>
  <member>
    <name>ADDRDEVICE</name>
    <value><int>24</int></value>
  </member>
  <member>
    <name>ADDRPORT</name>
    <value><int>1</int></value>
  </member>
</struct>
</value>

</member>
<member>
  <name>RSCOMMINITATIVE</name>
</member>
<value>
<struct>
  <member>
    <name>NAME</name>
    <value><string>Инициатива управления</string></value>
  </member>
  <member>
    <name>NUMBER</name>
    <value><int>14</int></value>
  </member>
  <member>
    <name>TYPEINITIATIVE</name>
  </member>
</struct>
</value>
<struct>
  <member>
    <name>NAME</name>
    <value><string>Идентификация хозоргана новых приборов</string></value>
  </member>
  <member>
    <name>NUMBER</name>
    <value><int>14</int></value>
  </member>
  <member>
    <name>QUERY</name>
  </member>
</struct>
</value>
<array>
  <data>
</data>
</array>
<value>
<struct>
  <member>
    <name>NAME</name>
    <value><string>№ считывателя</string></value>
  </member>
  <member>
    <name>NUMBER</name>
    <value><int>1</int></value>
  </member>
  <member>
    <name>TYPEDATA</name>
    <value><string>DATA</string></value>
  </member>
  <member>
    <name>DATA</name>
  </member>
</struct>
</value>

```

```

<struct>
  <member>
    <name>NAME</name>
    <value><string>Номер считывателя</string></value>
  </member>
  <member>
    <name>NUMBER</name>
    <value><int>1</int></value>
  </member>
  <member>
    <name>TYPEVALUE</name>
    <value><string>DATA</string></value>
  </member>
  <member>
    <name>LENGTHVALUE</name>
    <value><int>1</int></value>
  </member>
  <member>
    <name>VALUES</name>
  <value>
    <array>
      <data>
        <value>
          <struct>
            <member>
              <name>VALUE</name>
              <value><int>1</int></value>
            </member>
          </struct>
        </value>
      </data>
    </array>
  </value>

  </member>
</struct>
</value>

  </member>
</struct>
</value>

  </member>
</struct>
</value>

<value>
  <struct>
    <member>
      <name>NAME</name>
      <value><string>Дополнительные данные</string></value>
    </member>
    <member>
      <name>NUMBER</name>
      <value><int>2</int></value>
    </member>
    <member>
      <name>TYPEDATA</name>
      <value><string>CONSTANT</string></value>
    </member>
    <member>
      <name>CONSTANT</name>
  <value>
    <struct>
      <member>
        <name>NAME</name>
        <value><string>Не указано</string></value>
      </member>
      <member>
        <name>NUMBER</name>
        <value><int>2</int></value>
      </member>
    </struct>
  </value>

```

```

</member>
<member>
  <name>TYPEVALUE</name>
  <value><string>CONSTANT</string></value>
</member>
<member>
  <name>LENGTHVALUE</name>
  <value><int>1</int></value>
</member>
<member>
  <name>VALUES</name>
<value>
  <array>
    <data>
<value>
<struct>
  <member>
    <name>VALUE</name>
    <value><int>0</int></value>
  </member>
</struct>
</value>

</data>
</array>
</value>

  </member>
</struct>
</value>

  </member>
</struct>
</value>

<value>
<struct>
  <member>
    <name>NAME</name>
    <value><string>Код ключа</string></value>
  </member>
  <member>
    <name>NUMBER</name>
    <value><int>3</int></value>
  </member>
  <member>
    <name>TYPEDATA</name>
    <value><string>DATA</string></value>
  </member>
  <member>
    <name>DATA</name>
<value>
<struct>
  <member>
    <name>NAME</name>
    <value><string>Код ключа</string></value>
  </member>
  <member>
    <name>NUMBER</name>
    <value><int>1</int></value>
  </member>
  <member>
    <name>TYPEVALUE</name>
    <value><string>DATA</string></value>
  </member>
  <member>
    <name>LENGTHVALUE</name>
    <value><int>8</int></value>

```

```

    </member>
    <member>
      <name>VALUES</name>
    <value>
      <array>
        <data>
          <value>
            <struct>
              <member>
                <name>VALUE</name>
                <value><int>1</int></value>
              </member>
            </struct>
          </value>

          <value>
            <struct>
              <member>
                <name>VALUE</name>
                <value><int>232</int></value>
              </member>
            </struct>
          </value>

          <value>
            <struct>
              <member>
                <name>VALUE</name>
                <value><int>143</int></value>
              </member>
            </struct>
          </value>

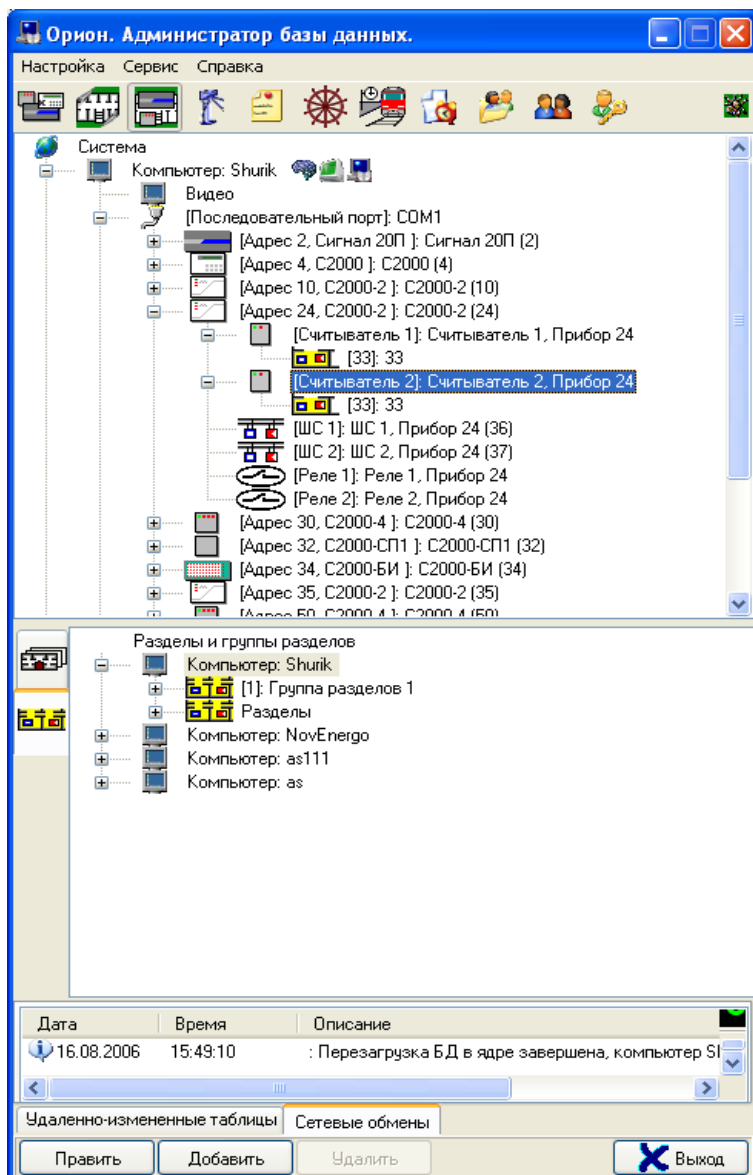
          <value>
            <struct>
              <member>
                <name>VALUE</name>
                <value><int>192</int></value>
              </member>
            </struct>
          </value>

          <value>
            <struct>
              <member>
                <name>VALUE</name>
                <value><int>6</int></value>
              </member>
            </struct>
          </value>

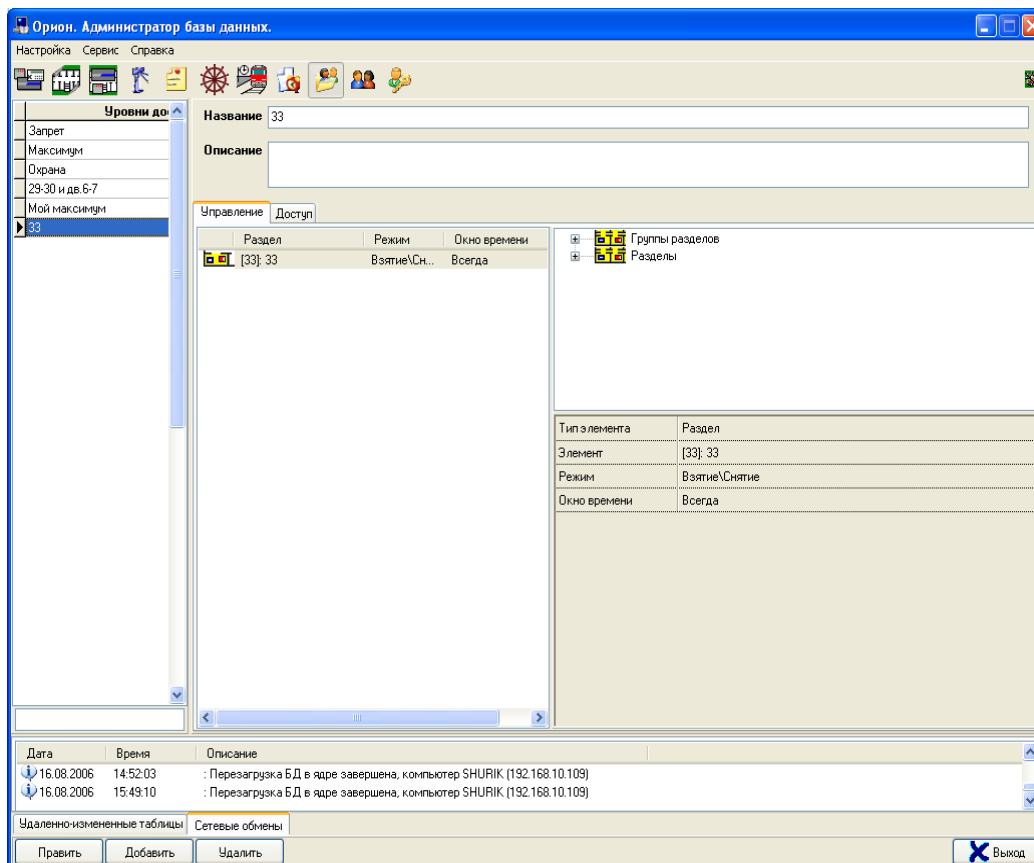
          <value>
            <struct>
              <member>
                <name>VALUE</name>
                <value><int>0</int></value>
              </member>
            </struct>
          </value>

          <value>
            <struct>
              <member>
                <name>VALUE</name>
                <value><int>0</int></value>
              </member>
            </struct>
          </value>

```

2) настройка уровня доступа:



3)полномочия пароля:


```
<member>
  <name>ADDRESS</name>
</value>
<struct>
  <member>
    <name>ADDRELEMENT</name>
    <value><int>1</int></value>
  </member>
  <member>
    <name>ADDRDEVICE</name>
    <value><int>24</int></value>
  </member>
  <member>
    <name>ADDRPORT</name>
    <value><int>1</int></value>
  </member>
</struct>
</value>

</member>
<member>
  <name>OBJECTS</name>
</value>
<array>
  <data>
</value>
<struct>
  <member>
    <name>TYPEOBJECT</name>
    <value><string>RZD</string></value>
  </member>
  <member>
    <name>STATUSOBJECT</name>
    <value><int>24</int></value>
  </member>
  <member>
    <name>INDEXOBJECT</name>
    <value><int>33</int></value>
  </member>
  <member>
    <name>RIGHTS</name>
    <value><int>3</int></value>
  </member>
  <member>
    <name>ZONES</name>
</value>
<array>
  <data>
</value>
<struct>
  <member>
    <name>ADDRESS</name>
</value>
<struct>
  <member>
    <name>ADDRELEMENT</name>
    <value><int>1</int></value>
  </member>
  <member>
    <name>ADDRDEVICE</name>
    <value><int>2</int></value>
  </member>
  <member>
    <name>ADDRPORT</name>
    <value><int>1</int></value>
  </member>
</struct>
</value>
</value>
```

```

    </member>
    <member>
      <name>STATUSOBJECT</name>
      <value><int>24</int></value>
    </member>
  </struct>
</value>

  </data>
</array>
</value>

  </member>
</struct>
</value>

  </data>
</array>
</value>

  </member>
</struct>
</value>

  </member>
  <member>
    <name>CONTRLPERSON</name>
    <value><int>7</int></value>
  </member>
  <member>
    <name>GUID</name>
    <value><string>{6630E19B-4B5A-4195-A20B-A775678B0174}</string></value>
  </member>
</struct>
</value>

  </param>

  </params>
</methodCall>

```

Далее модуль отрабатывает в соответствии с полномочиями удаленный запрос на взятие и берет под охрану раздел №33.

Орбон Про. Монитор оперативной задачи

16 19 45

1. Мой компьютер (SHURIK)

Цех №1

Коридор

102

101

Склад

Ворота

Первый этаж.

PM	Время	Событие	Раздел	Описание	Адрес	Зона доступа	Хозорган
SHURIK	16.08.2006 16:17:39	Снятие	1	Группа разделов 1		-	
SHURIK	16.08.2006 16:17:40	Снятие ШС	33	ШС 1, Прибор 2	1/0/2/1	-	
SHURIK	16.08.2006 16:17:40	Запуск сценария управления	0	НовыйСценарий1		-	
SHURIK	16.08.2006 16:18:04	Восстановление контакта	0	C2000-БИ (34)	1/0/34/0	-	
SHURIK	16.08.2006 16:18:19	Нет связи с прибором	0	C2000-БИ (34)	1/0/34/0	-	
SHURIK	16.08.2006 16:19:34	Идентификатор хозоргана	0	Считыватель 1, Прибор 24	1/0/24/1	-	Ишкина К.Н.
SHURIK	16.08.2006 16:19:37	Запрос на взятие	0	Считыватель 1, Прибор 24	1/0/24/1	-	Ишкина К.Н.
SHURIK	16.08.2006 16:19:37	Запрос на взятие	33	33		-	Ишкина К.Н.
SHURIK	16.08.2006 16:19:43	Взятие зоны под охрану	33	ШС 1, Прибор 2	1/0/2/1	-	
SHURIK	16.08.2006 16:19:43	Взят	33	33		-	Ишкина К.Н.
SHURIK	16.08.2006 16:19:43	Взятие	1	Группа разделов 1		-	
SHURIK	16.08.2006 16:19:43	Взятие	1	Группа разделов 1		-	

Запрос состояний

ВЗЯТ

Ниже приведен пример запроса состояния раздела №33 от C2000- БИ с помощью инициативы управления с запросом 5.

```
<?xml version="1.0" encoding="Windows-1251" ?>
```

```
<methodCall>
```

```
<methodName>ONINITIATIVE</methodName>
```

```
<params>
```

```
<param>
```

```
<value>
```

```
<struct>
```

```
<member>
```

```
<name>GUID</name>
```

```
<value><string>{990AF18D-D4F1-4718-A24E-9222F0F45CE5}</string></value>
```

```
</member>
```

```
<member>
<name>READERS</name>
```

```
<value>
```

```
<struct>
```

```
<member>
```

```
<name>ADDRESS</name>
```

```
<value>
```

```
<struct>
```

```
<member>
```

```

    <name>ADDRELEMENT</name>
    <value><int>1</int></value>
  </member>
  <member>
    <name>ADDRDEVICE</name>
    <value><int>34</int></value>
  </member>
  <member>
    <name>ADDRPORT</name>
    <value><int>1</int></value>
  </member>
</struct>
</value>

  </member>
  <member>
    <name>RSCOMMINITIAIVE</name>
  <value>
    <struct>
      <member>
        <name>NAME</name>
        <value><string>Инициатива управления</string></value>
      </member>
      <member>
        <name>NUMBER</name>
        <value><int>14</int></value>
      </member>
      <member>
        <name>TYPEINITIATIVE</name>
      <value>
        <struct>
          <member>
            <name>NAME</name>
            <value><string>Запрос состояния раздела</string></value>
          </member>
          <member>
            <name>NUMBER</name>
            <value><int>15</int></value>
          </member>
          <member>
            <name>QUERY</name>
          <value>
            <array>
              <data>
            <value>
              <struct>
                <member>
                  <name>NAME</name>
                  <value><string>№ считывателя</string></value>
                </member>
                <member>
                  <name>NUMBER</name>
                  <value><int>1</int></value>
                </member>
                <member>
                  <name>TYPEDATA</name>
                  <value><string>DATA</string></value>
                </member>
                <member>
                  <name>DATA</name>
                <value>
              <struct>
                <member>
                  <name>NAME</name>
                  <value><string>Номер считывателя</string></value>
                </member>
                <member>

```



```

    <name>NUMBER</name>
    <value><int>1</int></value>
  </member>
  <member>
    <name>TYPEVALUE</name>
    <value><string>DATA</string></value>
  </member>
  <member>
    <name>LENGTHVALUE</name>
    <value><int>1</int></value>
  </member>
  <member>
    <name>VALUES</name>
  <value>
    <array>
      <data>
        <value>
          <struct>
            <member>
              <name>VALUE</name>
              <value><int>1</int></value>
            </member>
          </struct>
        </value>

        </data>
      </array>
    </value>

    </member>
  </struct>
</value>

  </member>
</struct>
</value>

  <value>
    <struct>
      <member>
        <name>NAME</name>
        <value><string>Номер раздела</string></value>
      </member>
      <member>
        <name>NUMBER</name>
        <value><int>2</int></value>
      </member>
      <member>
        <name>TYPEDATA</name>
        <value><string>DATA</string></value>
      </member>
      <member>
        <name>DATA</name>
      <value>
        <struct>
          <member>
            <name>NAME</name>
            <value><string>Номер раздела</string></value>
          </member>
          <member>
            <name>NUMBER</name>
            <value><int>1</int></value>
          </member>
        </member>
      </value>
    </struct>
  </value>

```

```

    <name>TYPEVALUE</name>
    <value><string>DATA</string></value>
  </member>
  <member>
    <name>LENGTHVALUE</name>
    <value><int>2</int></value>
  </member>
  <member>
    <name>VALUES</name>
  <value>
    <array>
      <data>
        <value>
          <struct>
            <member>
              <name>VALUE</name>
              <value><int>33</int></value>
            </member>
          </struct>
        </value>

        <value>
          <struct>
            <member>
              <name>VALUE</name>
              <value><int>0</int></value>
            </member>
          </struct>
        </value>

        </data>
      </array>
    </value>

    </member>
  </struct>
</value>

  </member>
</struct>
</value>

  </member>
</struct>
</value>

  <value>
    <struct>
      <member>
        <name>NAME</name>
        <value><string>Дополнительные данные</string></value>
      </member>
      <member>
        <name>NUMBER</name>
        <value><int>3</int></value>
      </member>
      <member>
        <name>TYPEDATA</name>
        <value><string>CONSTANT</string></value>
      </member>
      <member>
        <name>CONSTANT</name>
      <value>
        <struct>
          <member>
            <name>NAME</name>
            <value><string>Не указано</string></value>
          </member>

```

```

    <member>
      <name>NUMBER</name>
      <value><int>2</int></value>
    </member>
    <member>
      <name>TYPEVALUE</name>
      <value><string>CONSTANT</string></value>
    </member>
    <member>
      <name>LENGTHVALUE</name>
      <value><int>1</int></value>
    </member>
    <member>
      <name>VALUES</name>
    <value>
      <array>
        <data>
          <value>
            <struct>
              <member>
                <name>VALUE</name>
                <value><int>0</int></value>
              </member>
            </struct>
          </value>

          </data>
        </array>
      </value>

    </member>
  </struct>
</value>

  </member>
</struct>
</value>

  </member>
</struct>
</value>

<value>
  <struct>
    <member>
      <name>NAME</name>
      <value><string>Код ключа</string></value>
    </member>
    <member>
      <name>NUMBER</name>
      <value><int>4</int></value>
    </member>
    <member>
      <name>TYPEDATA</name>
      <value><string>DATA</string></value>
    </member>
    <member>
      <name>DATA</name>
    <value>
      <struct>
        <member>
          <name>NAME</name>
          <value><string>Код ключа</string></value>
        </member>
        <member>
          <name>NUMBER</name>
          <value><int>1</int></value>
        </member>

```

```

    <member>
      <name>TYPEVALUE</name>
      <value><string>DATA</string></value>
    </member>
    <member>
      <name>LENGTHVALUE</name>
      <value><int>8</int></value>
    </member>
    <member>
      <name>VALUES</name>
    <value>
      <array>
        <data>
        </data>
      </array>
    </value>

```

На данных запрос клиентская программа определяет состояние объекта и передает состояние раздела 33 с помощью вызова TranslateRightsObjects.

```
<?xml version="1.0" encoding="Windows-1251" ?>
```

```

<methodCall>

  <methodName>TRANSLATERIGHTSOBJECTS</methodName>

  <params>

    <param>

      <value>
        <struct>
          <member>
            <name>READERS</name>
          <value>
            <struct>
              <member>
                <name>ADDRESS</name>
              <value>
                <struct>
                  <member>
                    <name>ADDRELEMENT</name>
                    <value><int>1</int></value>
                  </member>
                  <member>
                    <name>ADDRDEVICE</name>
                    <value><int>34</int></value>
                  </member>
                  <member>
                    <name>ADDRPORT</name>
                    <value><int>1</int></value>
                  </member>
                </struct>
              </value>
            </member>
          </value>
        </struct>
      </value>
    </param>
    <param>
      <value>
        <array>
          <data>
          </data>
        </array>
      </value>
    </param>
  </params>
</methodCall>

```

```

    <name>STATUSOBJECT</name>
    <value><int>24</int></value>
  </member>
  <member>
    <name>INDEXOBJECT</name>
    <value><int>33</int></value>
  </member>
  <member>
    <name>RIGHTS</name>
    <value><int>0</int></value>
  </member>
  <member>
    <name>ZONES</name>
  <value>
    <array>
      <data>
        <value>
          <struct>
            <member>
              <name>ADDRESS</name>
            <value>
              <struct>
                <member>
                  <name>ADDRELEMENT</name>
                  <value><int>1</int></value>
                </member>
                <member>
                  <name>ADDRDEVICE</name>
                  <value><int>2</int></value>
                </member>
                <member>
                  <name>ADDRPORT</name>
                  <value><int>1</int></value>
                </member>
              </struct>
            </value>

            </member>
            <member>
              <name>STATUSOBJECT</name>
              <value><int>24</int></value>
            </member>
          </struct>
        </value>

      </data>
    </array>
  </value>

  </member>
</struct>
</value>

  </data>
</array>
</value>

  </member>
</struct>
</value>

  </data>
</array>
</value>

  </member>
</struct>
</value>

  </member>
  <member>
    <name>GUID</name>
    <value><string>{5B6BDF1E-44CB-4252-B3E7-B6E6E08F7C4E}</string></value>
  </member>
</struct>
</value>

```

</param>
</params>
</methodCall>

При централизованном управлении трансляция состояния объекта управления-объектов управления осуществляется с помощью метода TranslateRightsObjects.

Параметры метода TranslateRightsObjects:

- **INT64 GUID** – жетон безопасности;
- **INT64 CONTRLPERSON** – идентификатор сотрудника которому принадлежит инициатива управления;
- **STRUCT READERS** – структура данных для считывателя с которого осуществляется управление:
 - **STRUCT ADDRESS** – структура данных определяющая адрес считывателя;
 - **INT64 ADDRELEMENT** – адрес элемента (номер шлейфа), зависит от типа устройства;
 - **INT64 ADDRDEVICE** – адрес устройства по RS 485;
 - **INT64 ADDRPUIT** – адрес пульта по RS 232;
 - **INT64 ADDRPORT** – номер порта, к которому подключен интерфейс.
 - **STRING NOTICE** – строковая константа с уведомительным сообщением, которое необходимо передать на считыватель иницирующего запрос:
 - «SENDEXT» - переслать текстовое сообщение;
 - «SENDBEEPS» - передать звуковой гудок;
 - «ACCESSDENIED» - доступ запрещен;
 - «WRONGPASSWORD» - не верный пароль;
 - «UNKNOWNPASSWORD» - не известный пароль;
 - «WRONGOBJECT» - не известный объект управления.
 - **STRING TEXTNOTICE** – текстовое сообщение которое необходимо передать на считыватель C2000-K;
 - **STRING TYPEBEEP** – константа, которая соответствует типу звукового сигнала:
 - 0 – выключение звука;
 - 1 – одиночный гудок;
 - 2 – двойной гудок;
 - 3 - тройной гудок;
 - 4 – длинный гудок;
 - 5 – прерывистый звуковой сигнал.
 - **ARRAY OBJECTS** – список объектов которыми необходимо управлять, элементы списка представляют из себя структуры данных:

- **STRING TYPEOBJECT**- тип объекта:

"RZD" - раздел;

"GROUPEPRZD" - группа разделов;

"DOOR" - точка доступа.

- **INT64 STATUSOBJECT** – значение определяющее состояние объекта (раздела) (см. таблицу 2 в приложении А);
- **INT64 INDEXOBJECT** – номер управляемого объекта (раздела);
- **INT64 RIGHTS** – права на управление объектом:

Для разделов и групп разделов:

0-просмотр;

1-взятие;

2-снятие;

3-взятие и снятие;

Для точек доступа:

0- проход;

1- вход;

2- выход;

3- вход/выход.

- **ARRAY ZONES** – список зон в разделе:

- **STRUCT ADDRESS** – структура данных определяющая адрес считывателя:

- **INT64 ADDRELEMENT** – адрес элемента (номер шлейфа), зависит от типа устройства;
- **INT64 ADDRDEVICE** – адрес устройства по RS 485;
- **INT64 ADDRPUIT** – адрес пульта по RS 232;
- **INT64 ADDRPORT** – номер порта, к которому подключен интерфейс.
- **INT64 STATUSOBJECT** – значение определяющее состояние зоны (см. таблицу 2 в приложении А);

- **ARRAY DOORS** – список полномочий доступа для управления, элементом является структуру данных:

- **STRUCT ADDRESS** – структура данных определяющая адрес считывателя с которого разрешено управлять дверью:

- **INT64 ADDRELEMENT** – адрес элемента (номер шлейфа), зависит от типа устройства;
- **INT64 ADDRDEVICE** – адрес устройства по RS 485;
- **INT64 ADDRPUIT** – адрес пульта по RS 232;
- **INT64 ADDRPORT** – номер порта, к которому подключен интерфейс.

- **INT64 ACCESSZONE** – зона в которую пускает указанный считыватель в заданном направлении;

- **INT64 ACCESSMODE** – режим прохода:

0 – проход;

1- вход;

2 – выход.

- **STRUCT CONTROLOUTKEY** – структура с данными управляющих реле:
 - **STRUCT ADDRESS** – структура данных определяющая адрес управляющего реле открывающего дверь в заданном направлении:
 - **INT64 ADDRELEMENT** – адрес элемента (номер шлейфа), зависит от типа устройства;
 - **INT64 ADDRDEVICE** – адрес устройства по RS 485;
 - **INT64 ADDRPUULT** – адрес пульта по RS 232;
 - **INT64 ADDRPORT** – номер порта, к которому подключен интерфейс.
 - **INT64 CONTROLOUTKEYCOMMAND** – управляющая команда для выхода, необходимая для открытия двери:
 - 1 – включить на время (для электромеханических замков);
 - 2 – выключить на время (для электромагнитных замков).
 - **INT64 CONTROLOUTKEYDURATION** – время воздействия команды (в сек).

Пример трансляции прав для совмещенного ключа (и на управление точкой доступа и на управление разделом).

```
<?xml version="1.0" encoding="Windows-1251" ?>
```

```
<methodCall>
```

```
  <methodName>TRANSLATERIGHTSOBJECTS</methodName>
```

```
  <params>
```

```
    <param>
```

```
      <value>
```

```
        <struct>
```

```
          <member>
```

```
            <name>READERS</name>
```

```
      <value>
```

```
        <struct>
```

```
          <member>
```

```
            <name>ADDRESS</name>
```

```
      <value>
```

```
        <struct>
```

```
          <member>
```

```
            <name>ADDRELEMENT</name>
```

```
            <value><int>1</int></value>
```

```
          </member>
```

```
          <member>
```

```
            <name>ADDRDEVICE</name>
```

```
            <value><int>24</int></value>
```

```
          </member>
```

```
          <member>
```

```
            <name>ADDRPORT</name>
```

```
            <value><int>1</int></value>
```

```
          </member>
```

```
        </struct>
```

```
      </value>
```

```
    </member>
```

```
    <member>
```

```
      <name>OBJECTS</name>
```

```
  <value>
```

```
    <array>
```

```
    <data>
```

```
  <value>
```

```
    <struct>
```



```

    <member>
      <name>TYPEOBJECT</name>
      <value><string>DOOR</string></value>
    </member>
    <member>
      <name>INDEXOBJECT</name>
      <value><int>11</int></value>
    </member>
    <member>
      <name>RIGHTS</name>
      <value><int>1</int></value>
    </member>
    <member>
      <name>DOORS</name>
    <value>
      <array>
        <data>
    <value>
      <struct>
        <member>
          <name>ADDRESS</name>
    <value>
      <struct>
        <member>
          <name>ADDRELEMENT</name>
          <value><int>1</int></value>
        </member>
        <member>
          <name>ADDRDEVICE</name>
          <value><int>24</int></value>
        </member>
        <member>
          <name>ADDRPORT</name>
          <value><int>1</int></value>
        </member>
      </struct>
    </value>

    </member>
    <member>
      <name>ACCESSMODE</name>
      <value><int>1</int></value>
    </member>
    <member>
      <name>ACCESSZONE</name>
      <value><int>1</int></value>
    </member>
    <member>
      <name>CONTROLOUTKEY</name>
    <value>
      <struct>
        <member>
          <name>ADDRELEMENT</name>
          <value><int>1</int></value>
        </member>
        <member>
          <name>ADDRDEVICE</name>
          <value><int>24</int></value>
        </member>
        <member>
          <name>ADDRPORT</name>
          <value><int>1</int></value>
        </member>
      </struct>
    </value>

    </member>
    <member>

```

```

    <name>CONTROLOUTKEYCOMMAND</name>
    <value><int>3</int></value>
  </member>
  <member>
    <name>CONTROLOUTKEYDURATION</name>
    <value><int>5</int></value>
  </member>
</struct>
</value>

</data>
</array>
</value>

</member>
</struct>
</value>

</data>
</array>
</value>

</member>
</struct>
</value>

</member>
<member>
  <name>CONTRLPERSON</name>
  <value><int>7</int></value>
</member>
<member>
  <name>GUID</name>
  <value><string>{899AE949-C674-448C-AC22-06C2DC41BCFA}</string></value>
</member>
</struct>
</value>

</param>

</params>

</methodCall>

```

Для того чтобы индикационные приборы типа С2000-БИ, С2000- ПТ и С2000-КС отображали состояния объектов (разделов, направления зон пожаротушения) системный контроллер должен при изменении состояния объектов отсылать соответствующее уведомительное сообщение с помощью метода TRANSLATESTATUSOBJECTS.

Параметры метода *TranslateStatusObjects*:

- **INT64 GUID** – жетон безопасности;
- **STRUCT ADDRESS** – структура данных определяющая адрес считывателя с которого осуществляется управление:
 - **INT64 ADDRELEMENT** – номер считывателя;
 - **INT64 ADDRDEVICE** – адрес устройства по RS 485;
 - **INT64 ADDRPUULT** – адрес пульта по RS 232;

- **INT64 ADDRPORT** – номер порта, к которому подключен интерфейс.
- **STRING NOTICE** – строковая константа с уведомительным сообщением которое необходимо передать на считыватель с которого «пришел» запрос об управлении:
 «SENTEXT» - переслать текстовое сообщение;
 «SENDBEEPS» - передать звуковой гудок;
 «ACCESSDENIED» - доступ запрещен;
 «WRONGPASSWORD» - не верный пароль;
 «UNKNOWNPASSWORD» - не известный пароль;
 «WRONGOBJECT» - не известный объект управления.
- **STRING TEXTNOTICE** – текстовое сообщение которое необходимо передать на считыватель C2000-K;
- **STRING TYPEBEEP** – константа, которая соответствует типу звукового сигнала:
 0 – выключение звука;
 1 – одиночный гудок;
 2 – двойной гудок;
 3 – тройной гудок;
 4 – длинный гудок;
 5 – прерывистый звуковой сигнал.
- **ARRAY OBJECTS** – список объектов которыми необходимо управлять, элементы списка представляют из себя структуры данных:
 - **INT64 TYPEOBJECT**- тип объекта:
 2 – раздел;
 11 – точка доступа;
 12 – зона доступа;
 17 – группа разделов.
 - **INT64 STATUSOBJECT** – значение определяющее состояние объекта;
 - **STRING CHANGESTATUSACTION** - признак того, что состояние было изменено по приходу события:
 «NEWSTATUS» - состояние изменилось по приходу события;
 «ACTION» - состояние транслируется и не связано с приходом события.
 - **ARRAY ZONES** – список зон в разделе:
 - **STRUCT ADDRESS** – структура данных определяющая адрес считывателя;
 - **INT64 ADDRELEMENT** – номер шлейфа;
 - **INT64 ADDRDEVICE** – адрес устройства по RS 485;
 - **INT64 ADDRPUULT** – адрес пульта по RS 232;
 - **INT64 ADDRPORT** – номер порта, к которому подключен интерфейс.
 - **INT64 STATUSOBJECT** – значение определяющее состояние зоны.

Пример передачи состояния раздела №33, на прибор с адресом 63(C2000-KC):

```
<?xml version="1.0" encoding="Windows-1251" ?>
```

```
<methodCall>
```

```
<methodName>TRANSLATESTATUSOBJECTS</methodName>
```

```
<params>
```

```
<param>
```

```
<value>
```

```
<struct>
```

```
<member>
```

```
<name>READERS</name>
```

```
</value>
```

```
<struct>
```

```
<member>
```

```
<name>ADDRESS</name>
```

```
</value>
```

```
<struct>
```

```
<member>
```

```
<name>ADDRELEMENT</name>
```

```
<value><int>1</int></value>
```

```
</member>
```

```
<member>
```

```
<name>ADDRDEVICE</name>
```

```
<value><int>63</int></value>
```

```
</member>
```

```
<member>
```

```
<name>ADDRPORT</name>
```

```
<value><int>1</int></value>
```

```
</member>
```

```
</struct>
```

```
</value>
```

```
</member>
```

```
<member>
```

```
<name>OBJECTS</name>
```

```
</value>
```

```
<array>
```

```
<data>
```

```
<value>
```

```
<struct>
```

```
<member>
```

```
<name>TYPEOBJECT</name>
```

```
<value><string>RZD</string></value>
```

```
</member>
```

```
<member>
```

```
<name>STATUSOBJECT</name>
```

```
<value><int>109</int></value>
```

```
</member>
```

```
<member>
```

```
<name>INDEXOBJECT</name>
```

```
<value><int>33</int></value>
```

```
</member>
```

```
<member>
```

```
<name>CHANGESTATUSACTION</name>
```

```
<value><string>NEWSTATUS</string></value>
```

```
</member>
```

```
</struct>
```

```
</value>
```

```
</data>
```

```
</array>
```

```
</value>
```

Метод **TRANSLATERSEVENT** транслирует на прибор типа С2000-К события для отображения текстовой информацией на индикаторе прибора.

Параметры метода *TranslateRsEvent*:

- **INT64 GUID** – жетон безопасности;
- **ARRAY READERS** – список считывателей на которые необходимо передать событие, элемент списка является структурой данных:
 - **STRUCT ADDRESS** – структура данных определяющая адрес считывателя на который необходимо ретранслировать событие:
 - **INT64 ADDRELEMENT** – номер считывателя;
 - **INT64 ADDRDEVICE** – адрес устройства по RS 485;
 - **INT64 ADDRPUULT** – адрес пульта по RS 232;
 - **INT64 ADDRPORT** – номер порта, к которому подключен интерфейс.
- **INT64 EVENT** – код события, в таблице приведены коды основных событий системы (см. таблицу 3 в приложении А);
- **INT64 ZONE** – номер зоны по которой произошло событие;
- **INT64 RZD** – номер раздела(объекта) в котором произошло событие;
- **INT64 NUMBERPERSONAL** – идентификатор сотрудника для которого произошло событие;
- **FLOAT DATETIME** – время и дата вычитывания события из прибора;
- **STRUCT TRANSLATEADDRESS** – структура данных определяющая адрес устройства (элемента) с которого было считано событие:
 - **INT64 ADDRELEMENT** – адрес элемента (номер шлейфа, считывателя), зависит от типа устройства;
 - **INT64 ADDRDEVICE** – адрес устройства по RS 485;
 - **INT64 ADDRPUULT** – адрес пульта по RS 232;
 - **INT64 ADDRPORT** – номер порта, к которому подключен интерфейс.

Пример трансляции события на С2000-К с адресом 66.

```
<?xml version="1.0" encoding="Windows-1251" ?>
```

```
<methodCall>
```

```
<methodName>TRANSLATERSEVENT</methodName>
```

```
<params>
```

```
<param>
```

```
<value>
```

```
<struct>
```

```
<member>
```

```
<name>DEVICES</name>
```

```
<value>
```

```
<array>
```

```
<data>
```

```
<value>
```

```
<struct>
```

```
<member>
```

```
<name>ADDRESS</name>
```

```
<value>
```

```
<struct>
```

```

<member>
  <name>ADDRDEVICE</name>
  <value><int>66</int></value>
</member>
<member>
  <name>ADDRPORT</name>
  <value><int>1</int></value>
</member>
</struct>
</value>

</member>
<member>
  <name>EVENT</name>
  <value><int>242</int></value>
</member>
<member>
  <name>RZD</name>
  <value><int>33</int></value>
</member>
<member>
  <name>DATETIME</name>
  <value><double>38947,5785224074</double></value>
</member>
</struct>
</value>

</data>
</array>
</value>

</member>
<member>
  <name>GUID</name>
  <value><string>{C7DEDFE7-1A71-4C0F-B7AB-F1BC95852CD8}</string></value>
</member>
</struct>
</value>

</param>

</params>

</methodCall>

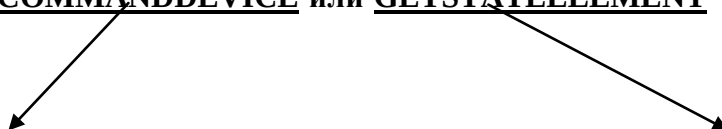
```

Метод RsCommand позволяет управлять устройствами и их агрегатами.

Метод представляет из себя входящий вызов, ответ на который будет получен в результате вызова у клиента метода, имя которого должно быть передано в параметре RESULTRETURNMETHOD.

Параметры метода RsCommand:

- **INT64 GUID** – жетон безопасности;
- **STRING IPSEVER** – Ip адрес сервера, на который необходимо выслать ответ с помощью вызова метода указанного в параметре RESULTRETURNMETHOD;
- **INT64 PORTSERVER** – порт сервера, , на который необходимо выслать ответ с помощью вызова метода указанного в параметре RESULTRETURNMETHOD;
- **STRUCT RSCOMMANDDEVICE** или **GETSTATEELEMENT** – структу с данными



команды(команд) для устройства или запрос состояния агрегата устройства:

Array Devices – список устройств которым необходимо транслировать команды

Array Devices или(и) Readers или(и) OutKeys или(и) Zones- списки агрегатов для которых необходимо запросить состояние

Struct Address – адрес устройства на который высылается команда;
Int64 ContrlPersonal – идентификатор сотрудника
Stuct ReturnDataFromClient – структура данных которые необходимо вернуть клиенту в ответе на команду;
Struct RSCommandParams – данные запроса;
String ResultReturnMethod – имя метода который необходимо вызвать при посылке ответа

Struct Address – адрес агрегата у которого необходимо запросить состояние;
Int64 ContrlPersonal – идентификатор сотрудника;
Struct ReturnDataFromClient – структура данных которые необходимо вернуть клиенту в ответе на команду;
String ResultReturnMethod – имя метода который необходимо вызвать при посылке ответа

- **ARRAY DEVICES** (для обоих запросов) и (или) **READERS** (только для **GETSTATEELEMENT**) и (или) **OUTKEYS** (только для **GETSTATEELEMENT**) и (или) **READERS** (только для **GETSTATEELEMENT**) и (или) **ZONES** (только для **GETSTATEELEMENT**) – список устройств (агрегатов) на которые необходимо выслать команду или узнать состояние, элементом списка является структура данных (STRUCT):
 - **STRUCT ADDRESS** – структура данных определяющая адрес прибора (агрегата):
 - **INT64 ADDRDEVICE** – адрес устройства по RS 485;
 - **INT64 ADDRELEMENT** – адрес элемента (номер шлейфа, считывателя, реле), зависит от типа устройства;
 - **INT64 ADDRPUOT** – адрес пульта по RS 232;
 - **INT64 ADDRPORT** – номер порта, к которому подключен интерфейс;
 - **STRUCT RETURNDATAFROMCLIENT** – структура с данными которые необходимо вернуть клиенту вместе с ответом на команду;
 - **INT64 CONTRLPERSON** – идентификатор сотрудника которому принадлежит запрос;
 - **STRING RESULTRETURNMETHOD** – имя запроса с помощью которого необходимо возвратить ответ;

- **STRUCT RSCOMMANDPARAMS** (только для RSCOMMANDDEVICE) – структура данных для команды:

В таблице приведены все возможные параметры команд управления, структуру данных также можно получить по запросу GetListRSCommand.

Номер	Наименование запроса	Данные	Дл.	Варианты
13	Запрос данных об устройстве	Дополнительные данные	1	0
		Дополнительные данные 2	1	0
19	Взятие-снятие зоны (ШС) под охрану	Номер шлейфа, зоны	1	
		Тип взятия	1	Константа: 0 - снятие ШС; 1 - групповое снятие; 2 - взятие ШС; 3 - групповое взятие; 4 - сброс тревог; 5 - отключение автоматики; 6 - включение автоматики; 7 - сброс пуска АСПТ; 8 - дистанционный пуск;
		Дополнительные данные	1	0-не указано 1-желательное действие взятие-снятие 2-получить состояние всех разделов
		[Код ключа]	8	
21	Команда внешнего управления реле	Номер реле	1	
		Номер программы управления	1	0 - Возврат в исходное состояние; 1 -Включить; 2 – Выключить; 3 - Включить на время; 4 - Выключить на время; 5 - Мигать из состояния выключено; 6 - Мигать из состояния включено; 7 - Мигать из состояния выключено на время; 8 - Мигать из состояния

				включено на время; 9 – Лампа; 10 – ПЦН; 11 - Пуск АСПТ; 12 - Пуск-стоп речевого оповещения.
		[Номер маски мигания]	1	С 1- 64
		[Задержка управления]	2	В одной единице 0.125мсек.
		[Время управления]	2	В одной единице 0.125мсек.
27	Запрос информации телеизмерения(АЦП)	Номер шлейфа, зоны	1	
35	Управление доступом	Управляющая команда доступом	1	0 -Предоставление доступа; 1 - Разрешение(восстановление) доступа; 2 - Разрешение входа; 3 - Разрешение выхода; 4 - Запрет доступа; 5 - Запрет входа; 6 - Запрет выхода; 7 - Открытие доступа.
		Номер считывателя	1	
		[Нарушения]	1	
		[Код ключа]	8	
49	Трансляция событий	№ сообщений	1	
		Код события	1	
		Номер зоны	1	
		Адрес прибора	1	
		Номер раздела	1	
		Номер хозоргана	1	
		Часы	1	
		Минуты	1	
		Секунды	1	
57	Запрос параметров адресной зоны	№ зоны		
		Вид параметра	1	1 - тип адресного устройства; 2 - Запрос состояния ШС; 3 - Запрос режима ШС; 4 - Запрос АЦП;

				5 - Блочный запрос АЦП; 6 - Запрос АЦП выхода; 7 - Запрос состояния КЦ; 8 - Запрос состояния двери; 9 - Запрос состояния доступа; 10 - Запрос АЦП в текстовом формате.
71	Состояние доступа	№ считывателя	1	

- **INT64 NUMBER** – номер команды;
- **ARRAY COLLECTDATA**– список данных в параметре, представляет из себя структуру данных:
 - **STRING NAME** – наименование данных;
 - **STRING DATA** – тип данных:
 - «DATA» - данные;
 - «CONSTANT» - заранее описанные константные значения;
 - **STRUCT DATA** - структура данных для данного параметра запроса:
 - **STRING NAME** – описание данных;
 - **INT64 TYPEVALUE** – тип данных;
 - **INT64 LENGTHVALUE** – длина списка значений;
 - **ARRAY VALUES** – список значений для данного параметра, каждый элемент списка представляет из себя структуру в которой хранится один байт значения:
 - **INT64 VALUE** – значение элемента списка.

В ответ на вызов метода RSCommand возможны два варианта ответа с различными структурами RESULTDATA:.

Ответы на вызов RsCommand:

- на запрос «GetStateElements» (запрос состояния устройства):

- **STRING NAMERESULT** - имя метода который был вызван у клиента для возврата ответа на команду;
- **STRUCT RETURNDATAFROMCLIENT** – структура данных которая была возвращена клиенту из метода RSCommand из поля ReturnDataFromClient;
- **STRING METHODNAME** - имя команды в запросе RSCommand на который был возвращен ответ:

Значение данного поля может принимать одно из двух значений в зависимости от запроса на который был выслан ответ:

- «GetStateElements», запрос состояния агрегата (прибор, считыватель, реле, зона, шлейф);
- «RSCommandDevice», команда устройству;

IF METHODNAME =
«GetStateElements»- ответ
на запрос о состоянии
агрегатов

IF METHODNAME =
«RSCOMMANDDEVICE» -
ответ на команду
устройству

Array Devices или(и)
Readers или(и) OutKeys
или(и) Zones – список
агрегатов и их состояний

Array Devices – список
устройств для которых был
сформирован ответ

Struct Address – адрес
агрегата
Int64 State – состояние
String NameState –
наименование состояния

Struct
RSCOMMANDANSWER –
структура с данными ответа

- **ARRAY DEVICES** (для обоих запросов) и (или) **READERS**(только для **GETSTATEELEMENT**) и (или) **OUTKEYS** (только для **GETSTATEELEMENT**) и (или) **READERS** только для **GETSTATEELEMENT**) и (или) **ZONES** (только для **GETSTATEELEMENT**) – список устройств (агрегатов) для которых был получен ответ, элемент списка-структура данных (STRUCT):

- **STRUCT ADDRESS** – структура данных определяющая адрес прибора (агрегата):
 - **INT64 ADDRELEMENT** – адрес элемента (номер шлейфа, считывателя, реле), зависит от типа устройства;
 - **INT64 ADDRDEVICE** – адрес устройства по RS 485;
 - **INT64 ADDRPUIT** – адрес пульта по RS 232;
 - **INT64 ADDRPORT** – номер порта, к которому подключен интерфейс;
 - **STRUCT RETURNDATAFROMCLIENT** – структура с данными которые необходимо вернуть клиенту вместе с ответом на команду;
- **INT64 CONTRLPERSON** – идентификатор сотрудника которому принадлежал запрос;
- **INT64 STATE** (только для **GetStateElement**) – состояние агрегата (шлейф, зона, реле, прибор);
- **STRING NAMESTATE** (только для **GetStateElement**)– наименование состояния агрегата;
- **STRUCT RSCOMMANDANSWER** (только для **RSCommandDevice**) – структура ответа:

Номер	Наименование запроса	Данные	Дл.	Варианты
-------	----------------------	--------	-----	----------

10	Ответ: команда не поддерживается (возможен ответ на любую команду)	Тип ответа	1	Константа: 0 - "Причина неизвестна"; 1 - нет контакта; 2- не поддерживаемый тип обмена команды; 3 -приняты не все необходимые данные; 4 - недопустимое значение данных; 5 - нет квитанции; 6 - адресное устройство команду не поддерживает; 7 - команда заблокирована; 8 - неверный код доступа; 9 -недостаточно полномочий.
0	Ответ на запрос 13	Тип устройства	1	(см таблицу 1 в приложении А)
		Версия прибора	1 2	
20	Подтверждение взятия-снятия(ответ на 19)	Номер шлейфа, зоны	1	
		Тип взятия	1	0 - снятие ШС; 1 - групповое снятие; 2 - взятие ШС; 3 - групповое взятие; 4 - сброс тревог; 5 - отключение автоматики; 6 - включение автоматики; 7 - сброс пуска АСПТ; 8 - дистанционный пуск;
		Дополнительные данные	1	0-не указано 1-желательное действие взятие-снятие 2-получить состояние всех разделов
		[Код ключа]	8	
22	Подтверждение управления выходами (ответ на 21)	Номер реле	1	
		Номер программы управления	1	0 - Возврат в исходное состояние;

				1 -Включить; 2 – Выключить; 3 - Включить на время; 4 - Выключить на время; 5 - Мигать из состояния выключено; 6 - Мигать из состояния включено; 7 - Мигать из состояния выключено на время; 8 - Мигать из состояния включено на время; 9 – Лампа; 10 – ПЦН; 11 - Пуск АСПТ; 12 - Пуск-стоп речевого оповещения.
28	Данные телеизмерения (ответ 27)	Номер шлейфа, зоны	1	
		Значение АЦП	1	
36	Ответ на запрос 35 (ответ на запрос 35)	Управляющая команда доступом	1	0 -Предоставление доступа; 1 - Разрешение(восстановление) доступа; 2 - Разрешение входа; 3 - Разрешение выхода; 4 - Запрет доступа; 5 - Запрет входа; 6 - Запрет выхода; 7 - Открытие доступа.
		Номер считывателя	1	
50	Подтверждение получения события(ответ на запрос 49)	№ сообщений	1	
58	Ответ на запрос 57	Тип параметра	1	0 - Ошибка параметра; 1 - Тип адресного устройства; 2 - Состояние зоны; 3- Режим зоны; 4 -Значение АЦП; 5 -Значение АЦП группы зон; 6- Значение АЦП выхода; 7 - Состояние КЦ;

				10 – Значен/ие АЦП в текстовом формате.
		Данные	1 или произвольн ая до 250 байт	Константы: Ответы по типу параметра «Ошибка параметра»(0): 0 - Отключен; 1 - не поддерживается; 2 - устройство не готово. Ответы по типу параметра «Тип адресного устройства»(1): 1 -C2000-AP1 v1.01; 2 - C2000-AP1 v1.03; 3 - ДИП-34А v1.02; 4 - C2000-AP8 v1.00; 5 - C2000-AP8 v1.01; 6 - C2000-AP2 v1.04; 7 - C2000-AP8 v1.05; 8 -C2000-AP1 исп01 v1.03 9 -C2000-AP2 v1.06; 10 - C2000-ИК v1.00; 11 - C2000-ИК v1.01; 20 - C2000-СТ; 30 -C2000-СП2; 40-C2000-ИП; 50-ИПР313-3А v1.00 51 -ИПР313-3А v1.01; 60 -C2000-AP1 v1.04; 61 -C2000-AP1 исп. 01 v1.05; 70 - C2000-AP8 v1.02; 80-ДИП 34А v1.03; 81 -ДИП 34А v1.04; Данные: Ответы на запрос «Состояние зоны» (2), «Номер режима»(3), «АЦП»(4), «Номер ШС»(5), «АЦП выхода»(6), «Код состояния КЦ»(7); Данные произвольной длины: Текстовая строка, N символов в кодировке 866

				кодовой страницы".
		[Значение АЦП при блочном запросе]	До 64 байт	
72	Ответ на запрос о состоянии доступа	№ считывателя	1	
		Состояние доступа	1	Байт состояния доступа: Бит 0 - запрет выхода (по кнопке); Бит 1 – запрет входа; Бит 2 – доступ открыт.

- **INT64 NUMBER** – номер команды;
- **ARRAY COLLECTDATA**– список данных в параметре, представляет из себя структуру данных:
 - **STRING NAME** – наименование данных;
 - **STRING DATA** – тип данных:
 - «DATA» - данные;
 - «CONSTANT» - заранее описанные константные значения;
 - **STRUCT DATA** - структура с данными для данного параметра запроса:
 - **STRING NAME** – описание данных;
 - **INT64 TYPEVALUE** – тип данных;
 - **INT64 LENGTHVALUE** – длина списка значений;
 - **ARRAY VALUES** – список значений для данного параметра, каждый элемент списка представляет из себя структуру в которой хранится один байт значения:
 - **INT64 VALUE** – значение элемента списка.

Пример:

1. Запрос состояния считывателя 1.34.1.

```
<?xml version="1.0" encoding="Windows-1251" ?>
```

```
<methodCall>
```

```
<methodName>RSCOMMAND</methodName>
```

```
<params>
```

```
<param>
```

```
<value>
```

```
<struct>
```

```
<member>
```

```
<name>GETSTATEELEMENT</name>
```

```
<value>
```

```
<struct>
```

```
<member>
```

```
<name>READERS</name>
```

```
<value>
```

```
<array>
```

```

    <data>
  <value>
    <struct>
      <member>
        <name>RETURNDATAFROMCLIENT</name>
  <value>
    <struct>
      <member>
        <name>IDSTACKCOMMAND</name>
        <value><double>260</double></value>
      </member>
    </struct>
  </value>

  </member>
  <member>
    <name>ADDRESS</name>
  <value>
    <struct>
      <member>
        <name>ADDRELEMENT</name>
        <value><int>1</int></value>
      </member>
      <member>
        <name>ADDRDEVICE</name>
        <value><int>34</int></value>
      </member>
      <member>
        <name>ADDRPORT</name>
        <value><int>1</int></value>
      </member>
    </struct>
  </value>

  </member>
  <member>
    <name>RESULTRETURNMETHOD</name>
    <value><string>ONRESULTCOMMAND</string></value>
  </member>
  <member>
    <name>IDPERSON</name>
    <value><int>0</int></value>
  </member>
</struct>
</value>

</data>
</array>
</value>

  </member>
</struct>
</value>

  </member>
  <member>
    <name>IPSERVER</name>
    <value><string>192.168.10.109</string></value>
  </member>
  <member>
    <name>PORTSERVER</name>
    <value><int>8080</int></value>
  </member>
  <member>
    <name>GUID</name>
    <value><string>{B771B48A-67B3-431A-95C1-A28D1561DC26}</string></value>
  </member>
</struct>

```


</value>

</param>

</params>

</methodCall>

Ответ на запрос GetStateElement.

<?xml version="1.0" encoding="Windows-1251" ?>

<methodCall>

<methodName>ONRESULTCOMMAND</methodName>

<params>

<param>

<value>

<struct>

<member>

<name>GUID</name>

<value><string>{167E7DED-3F75-4F41-9966-ED8F24067B26}</string></value>

</member>

<member>

<name>RESULT</name>

<value><string>METHOD IS EXECUTE</string></value>

</member>

<member>

<name>RESULTDATA</name>

<value>

<struct>

<member>

<name>NAMERESULT</name>

<value><string>ONRESULTCOMMAND</string></value>

</member>

<member>

<name>METHODNAME</name>

<value><string>GETSTATEELEMENT</string></value>

</member>

<member>

<name>RETURNDATAFROMCLIENT</name>

<value>

<struct>

<member>

<name>IDSTACKCOMMAND</name>

<value><double>260</double></value>

</member>

</struct>

</value>

</member>

<member>

<name>READERS</name>

<value>

<struct>

<member>

<name>ADDRESS</name>

<value>

<struct>

<member>

<name>ADDRELEMENT</name>

<value><int>1</int></value>

</member>

<member>

```

    <name>ADDRDEVICE</name>
    <value><int>34</int></value>
  </member>
  <member>
    <name>ADDRPORT</name>
    <value><int>1</int></value>
  </member>
</struct>
</value>

  </member>
  <member>
    <name>STATE</name>
    <value><int>0</int></value>
  </member>
  <member>
    <name>NAMESTATE</name>
    <value><string>HOPMA</string></value>
  </member>
</struct>
</value>

  </member>
</struct>
</value>

  </member>
</struct>
</value>

</param>

</params>

</methodCall>

```

2. Запрос взятия зоны под охрану.

```

<?xml version="1.0" encoding="Windows-1251" ?>

<methodCall>

  <methodName>RSCOMMAND</methodName>

  <params>

    <param>

      <value>
        <struct>
          <member>
            <name>RSCOMMANDDEVICE</name>
          </member>
        </struct>
      </value>
      <struct>
        <member>
          <name>DEVICES</name>
        </member>
      </struct>
      <value>
        <array>
          <data>
            <value>
              <struct>
                <member>
                  <name>RETURNDATAFROMCLIENT</name>
                </member>
              </struct>
            </value>
          </data>
        </array>
      </value>
    </param>
  </params>
</methodCall>

```

```

    <member>
      <name>IDSTACKCOMMAND</name>
      <value><double>171</double></value>
    </member>
  </struct>
</value>

  </member>
  <member>
    <name>ADDRESS</name>
  <value>
    <struct>
      <member>
        <name>ADDRDEVICE</name>
        <value><int>30</int></value>
      </member>
      <member>
        <name>ADDRPORT</name>
        <value><int>1</int></value>
      </member>
    </struct>
  </value>

  </member>
  <member>
    <name>RESULTRETURNMETHOD</name>
    <value><string>ONRESULTCOMMAND</string></value>
  </member>
  <member>
    <name>RSCOMMANDPARAMS</name>
  <value>
    <struct>
      <member>
        <name>NAME</name>
        <value><string>Взятие-снятие зоны</string></value>
      </member>
      <member>
        <name>NUMBER</name>
        <value><int>19</int></value>
      </member>
      <member>
        <name>TYPESCOLLECT</name>
        <value><string>DATA</string></value>
      </member>
      <member>
        <name>COLLECTDATA</name>
  <value>
    <array>
      <data>
  <value>
    <struct>
      <member>
        <name>NAME</name>
        <value><string>Homep III</string></value>
      </member>
      <member>
        <name>NUMBER</name>
        <value><int>1</int></value>
      </member>
      <member>
        <name>TYPEDATA</name>
        <value><string>DATA</string></value>
      </member>
      <member>
        <name>DATA</name>
  <value>
    <struct>
      <member>

```

```

    <name>NAME</name>
    <value><string>Homep IIIC 0-все IIIC</string></value>
  </member>
  <member>
    <name>NUMBER</name>
    <value><int>1</int></value>
  </member>
  <member>
    <name>TYPEVALUE</name>
    <value><string>DATA</string></value>
  </member>
  <member>
    <name>LENGTHVALUE</name>
    <value><int>1</int></value>
  </member>
  <member>
    <name>VALUES</name>
  <value>
    <array>
      <data>
        <value>
          <struct>
            <member>
              <name>VALUE</name>
              <value><int>1</int></value>
            </member>
          </struct>
        </value>

        </data>
      </array>
    </value>

    </member>
  </struct>
</value>

  </member>
</struct>
</value>

  </member>
</struct>
</value>

<value>
  <struct>
    <member>
      <name>NAME</name>
      <value><string>Вид взятия-снятия</string></value>
    </member>
    <member>
      <name>NUMBER</name>
      <value><int>2</int></value>
    </member>
    <member>
      <name>TYPEDATA</name>
      <value><string>CONSTANT</string></value>
    </member>
    <member>
      <name>CONSTANT</name>
    <value>
      <struct>
        <member>
          <name>NAME</name>
          <value><string>Снятие IIIC</string></value>
        </member>
        <member>
          <name>NUMBER</name>
          <value><int>1</int></value>
        </member>
      </struct>
    </member>
  </struct>

```

```

    <name>TYPEVALUE</name>
    <value><string>CONSTANT</string></value>
  </member>
  <member>
    <name>LENGTHVALUE</name>
    <value><int>1</int></value>
  </member>
  <member>
    <name>VALUES</name>
  <value>
    <array>
      <data>
        <value>
          <struct>
            <member>
              <name>VALUE</name>
              <value><int>0</int></value>
            </member>
          </struct>
        </value>

        </data>
      </array>
    </value>

    </member>
  </struct>
</value>

  </member>
</struct>
</value>

  </member>
</struct>
</value>

  </data>
</array>
</value>

  </member>
</struct>
</value>

  </member>
  <member>
    <name>IDPERSON</name>
    <value><int>1</int></value>
  </member>
</struct>
</value>

  </data>
</array>
</value>

  </member>
</struct>
</value>

  </member>
  <member>
    <name>IPSERVER</name>
    <value><string>192.168.10.109</string></value>
  </member>
  <member>
    <name>PORTSERVER</name>
    <value><int>8080</int></value>
  </member>
  <member>
    <name>GUID</name>

```

```

    <value><string>{32807E58-B7C5-4DCF-BA57-AA82F37DDB55}</string></value>
  </member>
</struct>
</value>

</param>

</params>

</methodCall>

```

Ответ на запрос:

22.08.2006 14:30:56 REQUEST XML-RPC Server

<?xml version="1.0" encoding="Windows-1251" ?>

```

<methodCall>

  <methodName>ONRESULTCOMMAND</methodName>

  <params>

    <param>

      <value>
        <struct>
          <member>
            <name>GUID</name>
            <value><string>{BEE1ED54-28CA-4057-B9C0-CAEF95D4F5DD}</string></value>
          </member>
          <member>
            <name>RESULT</name>
            <value><string>METHOD IS EXECUTE</string></value>
          </member>
          <member>
            <name>RESULTDATA</name>

```

```

    <name>RSCOMMANDANSWER</name>
<value>
<struct>
  <member>
    <name>NAME</name>
    <value><string>Подтверждение втятия-снятия</string></value>
  </member>
  <member>
    <name>NUMBER</name>
    <value><int>20</int></value>
  </member>
  <member>
    <name>TYPESCOLLECT</name>
    <value><string>DATA</string></value>
  </member>
  <member>
    <name>COLLECTDATA</name>
</value>
<array>
  <data>
<value>
<struct>
  <member>
    <name>NAME</name>
    <value><string>Homep IIIC</string></value>
  </member>
  <member>
    <name>NUMBER</name>
    <value><int>1</int></value>
  </member>
  <member>
    <name>TYPEDATA</name>
    <value><string>DATA</string></value>
  </member>
  <member>
    <name>DATA</name>
</value>
<struct>
  <member>
    <name>NAME</name>
    <value><string>Homep IIIC 0-все IIIC</string></value>
  </member>
  <member>
    <name>NUMBER</name>
    <value><int>1</int></value>
  </member>
  <member>
    <name>TYPEVALUE</name>
    <value><string>DATA</string></value>
  </member>
  <member>
    <name>LENGTHVALUE</name>
    <value><int>1</int></value>
  </member>
  <member>
    <name>VALUES</name>
</value>
<array>
  <data>
<value>
<struct>
  <member>
    <name>VALUE</name>
    <value><int>1</int></value>
  </member>
</struct>
</value>

```

```

</data>
</array>
</value>

    </member>
  </struct>
</value>

    </member>
  </struct>
</value>

<value>
  <struct>
    <member>
      <name>NAME</name>
      <value><string>Вид взятия-снятия</string></value>
    </member>
    <member>
      <name>NUMBER</name>
      <value><int>2</int></value>
    </member>
    <member>
      <name>TYPEDATA</name>
      <value><string>CONSTANT</string></value>
    </member>
    <member>
      <name>CONSTANT</name>
    </member>
  </struct>
</value>
  <struct>
    <member>
      <name>NAME</name>
      <value><string>Снятие ИС</string></value>
    </member>
    <member>
      <name>NUMBER</name>
      <value><int>1</int></value>
    </member>
    <member>
      <name>TYPEVALUE</name>
      <value><string>CONSTANT</string></value>
    </member>
    <member>
      <name>LENGTHVALUE</name>
      <value><int>1</int></value>
    </member>
    <member>
      <name>VALUES</name>
    </member>
  </struct>
</value>
  <array>
    <data>
  </data>
</array>
</value>

    </member>
  </struct>
</value>

</data>
</array>
</value>

    </member>
  </struct>
</value>

```



```

        </member>
    </struct>
</value>

    </data>
</array>
</value>

    </member>
</struct>
</value>

    </member>
</struct>
</value>

    </member>
</struct>
</value>

    </member>
</struct>
</value>

    </member>
</struct>
</value>

    </param>

</params>

</methodCall>

```

В модуле опроса опубликованы ряд информационных процедур ответ на которые возвращается синхронно:

- GetListMethods, позволяет получить список методов, комментарии для них и примеры их использования;
- GetListError, позволяет получить список ошибок возможных при вызове RPC методов и комментарии к ошибкам;
- GetListCommand, позволяет получить шаблоны команд для метода RsCommand;
- GetInfo, позволяет получить информацию о работающем модуле.

Параметры метода GetListMethods:

- **INT64 GUID** – жетон безопасности;

Параметры структуры **RESULTDATA** ответа:

- **STRING METHODNAME** – имя метода на который был дан ответ = «GetListMethods».
- **ARRAY METHODS** – список методов и их описание, элементами являются структуры данных:
 - **INT64 METHODSID** – номер метода;
 - **STRING METHODSNAME** – имя метода;
 - **STRING METHODSDESCRIPTION** – описание параметров для метода;
 - **INT64 METHODSTYPES** – тип метода;

- 0 – синхронный входящий вызов;
- 1- событие генерируемое модулем на стороне сервера;
- 2-асинхронные команды;
- 3-асинхронные команды.

Пример запроса и ответа:

Request Server(от 23.08.2006 14:04:08):<?xml version="1.0" encoding="Windows-1251" ?>

```
<methodCall>

  <methodName>GETLISTMETHODS</methodName>

  <params>

    <param>

      <value>
        <struct>
          <member>
            <name>GUID</name>
            <value><string>{5A12805E-0428-4871-BE44-8B53C2386004}</string></value>
          </member>
        </struct>
      </value>

    </param>

  </params>

</methodCall>
```

Response Client{от 23.08.2006 14:04:08}:<?xml version="1.0" encoding="Windows-1251" ?>

```
<methodResponse>

  <params>

    <param>

      <value>
        <struct>
          <member>
            <name>RESULT</name>
            <value><string>METHOD IS EXECUTE</string></value>
          </member>
          <member>
            <name>RESULTDATA</name>
            <value>
              <struct>
                <member>
                  <name>METHODNAME</name>
                  <value><string>GETLISTMETHODS</string></value>
                </member>
                <member>
                  <name>METHODS</name>
                  <value>
                    <array>
                      <data>
                        <value>
                          <struct>
                            <member>
                              <name>METHODSID</name>
                              <value><int>1</int></value>
                            </member>
                          </struct>
                        </value>
                      </data>
                    </array>
                  </value>
                </member>
              </struct>
            </value>
          </member>
        </struct>
      </value>

    </param>

  </params>

</methodResponse>
```

```

<member>
  <name>METHODSNAME</name>
  <value><string>GETLISTDEVICE</string></value>
</member>
<member>
  <name>METHODSDescription</name>
  <value><string>(IPSERVER: String; PORTSERVER: Integer; STATE : Integer{0-now result; 1-asynch result});</string></value>
</member>
<member>
  <name>METHODSTYPES</name>
  <value><int>0</int></value>
</member>
</struct>
</value>

<value>
  <struct>
    <member>
      <name>METHODSID</name>
      <value><int>2</int></value>
    </member>
    <member>
      <name>METHODSNAME</name>
      <value><string>SETSUBSCRIBE</string></value>
    </member>
    <member>
      <name>METHODSDescription</name>
      <value><string>(IPSERVER: String; PORTSERVER: Integer; LOGIN:String; {логин} PASSWORD: String; TYPECLIENT : Integer; {тип
клиента} SCRIBE : Integer; {подпись } DEVICES: array of Int64; {приборы на которые распространяется подпись});</string></value>
    </member>
    <member>
      <name>METHODSTYPES</name>
      <value><int>0</int></value>
    </member>
  </struct>
</value>

<value>
  <struct>
    <member>
      <name>METHODSID</name>
      <value><int>3</int></value>
    </member>
    <member>
      <name>METHODSNAME</name>
      <value><string>ONREVENT</string></value>
    </member>
    <member>
      <name>METHODSDescription</name>
      <value><string></string></value>
    </member>
    <member>
      <name>METHODSTYPES</name>
      <value><int>1</int></value>
    </member>
  </struct>
</value>

<value>
  <struct>
    <member>
      <name>METHODSID</name>
      <value><int>4</int></value>
    </member>
    <member>
      <name>METHODSNAME</name>
      <value><string>SETCONFIGURATIONHWSRV</string></value>
    </member>
  </struct>
</value>

```

```

    <member>
      <name>METHODSDESCRIPTION</name>
      <value><string></string></value>
    </member>
    <member>
      <name>METHODSTYPES</name>
      <value><int>0</int></value>
    </member>
  </struct>
</value>

<value>
  <struct>
    <member>
      <name>METHODSID</name>
      <value><int>5</int></value>
    </member>
    <member>
      <name>METHODSNAME</name>
      <value><string>GETLISTERROR</string></value>
    </member>
    <member>
      <name>METHODSDESCRIPTION</name>
      <value><string></string></value>
    </member>
    <member>
      <name>METHODSTYPES</name>
      <value><int>0</int></value>
    </member>
  </struct>
</value>

<value>
  <struct>
    <member>
      <name>METHODSID</name>
      <value><int>6</int></value>
    </member>
    <member>
      <name>METHODSNAME</name>
      <value><string>GETLISTMETHODS</string></value>
    </member>
    <member>
      <name>METHODSDESCRIPTION</name>
      <value><string></string></value>
    </member>
    <member>
      <name>METHODSTYPES</name>
      <value><int>0</int></value>
    </member>
  </struct>
</value>

<value>
  <struct>
    <member>
      <name>METHODSID</name>
      <value><int>7</int></value>
    </member>
    <member>
      <name>METHODSNAME</name>
      <value><string>GETCONFIGURATIONHWSERVER</string></value>
    </member>
    <member>
      <name>METHODSDESCRIPTION</name>
      <value><string></string></value>
    </member>
    <member>

```

```

    <name>METHODSTYPES</name>
    <value><int>0</int></value>
  </member>
</struct>
</value>

<value>
  <struct>
    <member>
      <name>METHODSID</name>
      <value><int>8</int></value>
    </member>
    <member>
      <name>METHODSNAME</name>
      <value><string>GETDEVICE</string></value>
    </member>
    <member>
      <name>METHODSDESCRIPTION</name>
      <value><string>Узнать состояние ус-ва</string></value>
    </member>
    <member>
      <name>METHODSTYPES</name>
      <value><int>2</int></value>
    </member>
  </struct>
</value>

<value>
  <struct>
    <member>
      <name>METHODSID</name>
      <value><int>9</int></value>
    </member>
    <member>
      <name>METHODSNAME</name>
      <value><string>GETSTATEELEMENTS</string></value>
    </member>
    <member>
      <name>METHODSDESCRIPTION</name>
      <value><string>Узнать состояние элемента</string></value>
    </member>
    <member>
      <name>METHODSTYPES</name>
      <value><int>0</int></value>
    </member>
  </struct>
</value>

<value>
  <struct>
    <member>
      <name>METHODSID</name>
      <value><int>10</int></value>
    </member>
    <member>
      <name>METHODSNAME</name>
      <value><string>RSCOMMAND</string></value>
    </member>
    <member>
      <name>METHODSDESCRIPTION</name>
      <value><string>RS команды:</string></value>
    </member>
    <member>
      <name>METHODSTYPES</name>
      <value><int>3</int></value>
    </member>
  </struct>
</value>

```

```

<value>
<struct>
  <member>
    <name>METHODSID</name>
    <value><int>11</int></value>
  </member>
  <member>
    <name>METHODSNAME</name>
    <value><string>ReadConfiguration</string></value>
  </member>
  <member>
    <name>METHODSDESCRIPTION</name>
    <value><string>Чтение конфигурации: 1.RPCGETSTATEELEMENT-запрос состояния элементов</string></value>
  </member>
  <member>
    <name>METHODSTYPES</name>
    <value><int>2</int></value>
  </member>
</struct>
</value>

<value>
<struct>
  <member>
    <name>METHODSID</name>
    <value><int>12</int></value>
  </member>
  <member>
    <name>METHODSNAME</name>
    <value><string>CLOSESCRIBE</string></value>
  </member>
  <member>
    <name>METHODSDESCRIPTION</name>
    <value><string>Отписаться</string></value>
  </member>
  <member>
    <name>METHODSTYPES</name>
    <value><int>0</int></value>
  </member>
</struct>
</value>

<value>
<struct>
  <member>
    <name>METHODSID</name>
    <value><int>13</int></value>
  </member>
  <member>
    <name>METHODSNAME</name>
    <value><string>ONCHANGESTATEELEMENTS</string></value>
  </member>
  <member>
    <name>METHODSDESCRIPTION</name>
    <value><string>Изменение состояния элемента</string></value>
  </member>
  <member>
    <name>METHODSTYPES</name>
    <value><int>1</int></value>
  </member>
</struct>
</value>

<value>
<struct>
  <member>
    <name>METHODSID</name>

```

```

    <value><int>14</int></value>
  </member>
  <member>
    <name>METHODSNAME</name>
    <value><string>ReadKeyCodeFromReader</string></value>
  </member>
  <member>
    <name>METHODSDESCRIPTION</name>
    <value><string>Считать код ключа</string></value>
  </member>
  <member>
    <name>METHODSTYPES</name>
    <value><int>3</int></value>
  </member>
</struct>
</value>

<value>
  <struct>
    <member>
      <name>METHODSID</name>
      <value><int>15</int></value>
    </member>
    <member>
      <name>METHODSNAME</name>
      <value><string>GETKEYLISTFROMDEVICE</string></value>
    </member>
    <member>
      <name>METHODSDESCRIPTION</name>
      <value><string>Получить список кодов ключей прибора</string></value>
    </member>
    <member>
      <name>METHODSTYPES</name>
      <value><int>3</int></value>
    </member>
  </struct>
</value>

<value>
  <struct>
    <member>
      <name>METHODSID</name>
      <value><int>16</int></value>
    </member>
    <member>
      <name>METHODSNAME</name>
      <value><string>SETKEY</string></value>
    </member>
    <member>
      <name>METHODSDESCRIPTION</name>
      <value><string>Записать ключ в прибор</string></value>
    </member>
    <member>
      <name>METHODSTYPES</name>
      <value><int>3</int></value>
    </member>
  </struct>
</value>

<value>
  <struct>
    <member>
      <name>METHODSID</name>
      <value><int>17</int></value>
    </member>
    <member>
      <name>METHODSNAME</name>
      <value><string>ONINITIATIVE</string></value>

```

```

</member>
<member>
  <name>METHODSDESCRIPTION</name>
  <value><string>Инициатива управления</string></value>
</member>
<member>
  <name>METHODSTYPES</name>
  <value><int>1</int></value>
</member>
</struct>
</value>

<value>
<struct>
  <member>
    <name>METHODSID</name>
    <value><int>18</int></value>
  </member>
  <member>
    <name>METHODSNAME</name>
    <value><string>TRANSLATERIGHTSOBJECTS</string></value>
  </member>
  <member>
    <name>METHODSDESCRIPTION</name>
    <value><string>Передача на считыватель полномочий управления</string></value>
  </member>
  <member>
    <name>METHODSTYPES</name>
    <value><int>0</int></value>
  </member>
</struct>
</value>

<value>
<struct>
  <member>
    <name>METHODSID</name>
    <value><int>19</int></value>
  </member>
  <member>
    <name>METHODSNAME</name>
    <value><string>TRANSLATESTATUSOBJECTS</string></value>
  </member>
  <member>
    <name>METHODSDESCRIPTION</name>
    <value><string>Передача на считыватель состояния раздела и других объектов</string></value>
  </member>
  <member>
    <name>METHODSTYPES</name>
    <value><int>0</int></value>
  </member>
</struct>
</value>

<value>
<struct>
  <member>
    <name>METHODSID</name>
    <value><int>20</int></value>
  </member>
  <member>
    <name>METHODSNAME</name>
    <value><string>CONTROBJECTS</string></value>
  </member>
  <member>
    <name>METHODSDESCRIPTION</name>
    <value><string>Управление объектом</string></value>
  </member>
</struct>

```



```

    <member>
      <name>METHODSTYPES</name>
      <value><int>0</int></value>
    </member>
  </struct>
</value>

<value>
  <struct>
    <member>
      <name>METHODSID</name>
      <value><int>21</int></value>
    </member>
    <member>
      <name>METHODSNAME</name>
      <value><string>GETINFO</string></value>
    </member>
    <member>
      <name>METHODSDESCRIPTION</name>
      <value><string>Запрос информации о сервере</string></value>
    </member>
    <member>
      <name>METHODSTYPES</name>
      <value><int>0</int></value>
    </member>
  </struct>
</value>

<value>
  <struct>
    <member>
      <name>METHODSID</name>
      <value><int>22</int></value>
    </member>
    <member>
      <name>METHODSNAME</name>
      <value><string>ONNEWDEVICE</string></value>
    </member>
    <member>
      <name>METHODSDESCRIPTION</name>
      <value><string>Найдено новое устройство</string></value>
    </member>
    <member>
      <name>METHODSTYPES</name>
      <value><int>1</int></value>
    </member>
  </struct>
</value>

<value>
  <struct>
    <member>
      <name>METHODSID</name>
      <value><int>23</int></value>
    </member>
    <member>
      <name>METHODSNAME</name>
      <value><string>ONRESULTCOMMAND</string></value>
    </member>
    <member>
      <name>METHODSDESCRIPTION</name>
      <value><string>Ответ на команду </string></value>
    </member>
    <member>
      <name>METHODSTYPES</name>
      <value><int>1</int></value>
    </member>
  </struct>

```

```

</value>

<value>
<struct>
  <member>
    <name>METHODSID</name>
    <value><int>24</int></value>
  </member>
  <member>
    <name>METHODSNAME</name>
    <value><string>ONCLOSEDESCRIBE</string></value>
  </member>
  <member>
    <name>METHODSDESCRIPTION</name>
    <value><string>Отписаться с генерацией события</string></value>
  </member>
  <member>
    <name>METHODSTYPES</name>
    <value><int>0</int></value>
  </member>
</struct>
</value>

<value>
<struct>
  <member>
    <name>METHODSID</name>
    <value><int>25</int></value>
  </member>
  <member>
    <name>METHODSNAME</name>
    <value><string>GETLISTDEVICES</string></value>
  </member>
  <member>
    <name>METHODSDESCRIPTION</name>
    <value><string>Получить список устройств с которыми работает HWServer</string></value>
  </member>
  <member>
    <name>METHODSTYPES</name>
    <value><int>0</int></value>
  </member>
</struct>
</value>

<value>
<struct>
  <member>
    <name>METHODSID</name>
    <value><int>26</int></value>
  </member>
  <member>
    <name>METHODSNAME</name>
    <value><string>GETLISTRSCOMMAND</string></value>
  </member>
  <member>
    <name>METHODSDESCRIPTION</name>
    <value><string>Получить шаблоны RS команд</string></value>
  </member>
  <member>
    <name>METHODSTYPES</name>
    <value><int>0</int></value>
  </member>
</struct>
</value>

<value>
<struct>
  <member>

```

```

    <name>METHODSID</name>
    <value><int>27</int></value>
  </member>
  <member>
    <name>METHODSNAME</name>
    <value><string>TRANSLATERSEVENT</string></value>
  </member>
  <member>
    <name>METHODSDESCRIPTION</name>
    <value><string>Трансляция события на устройство</string></value>
  </member>
  <member>
    <name>METHODSTYPES</name>
    <value><int>0</int></value>
  </member>
</struct>
</value>

</data>
</array>
</value>

  </member>
</struct>
</value>

  </member>
</struct>
</value>

</param>

</params>

</methodResponse>

```

Параметры метода *GetListError*:

- **INT64 GUID** – жетон безопасности;

Параметры структуры **RESULTDATA** для ответа:

- **STRING METHODNAME** – имя метода на который был дан ответ = «GetListError»;
- **ARRAY ERRORS** – список ошибок и их описание, элементами являются структуры данных:
 - **INT64 CODERROR** – код ошибки;
 - **STRING NAMEERROR** – наименование ошибки;
 - **STRING DESCRIPTIONERROR** – описание ошибки.

Пример запроса и ответа/

Request Server(от 23.08.2006 14:37:36):<?xml version="1.0" encoding="Windows-1251" ?>

```

<methodCall>

  <methodName>GETLISTERROR</methodName>

  <params>

    <param>

      <value>
        <struct>

```

```
<member>
  <name>GUID</name>
  <value><string>{5A12805E-0428-4871-BE44-8B53C2386004}</string></value>
</member>
</struct>
</value>
```

```
</param>

</params>

</methodCall>
```

Response Client{or 23.08.2006 14:37:36};<?xml version="1.0" encoding="Windows-1251" ?>

```
<methodResponse>

  <params>

    <param>

      <value>
        <struct>
          <member>
            <name>RESULT</name>
            <value><string>METHOD IS EXECUTE</string></value>
          </member>
          <member>
            <name>RESULTDATA</name>
            <value>
              <struct>
                <member>
                  <name>METHODNAME</name>
                  <value><string>GETLISTERROR</string></value>
                </member>
                <member>
                  <name>ERRORS</name>
                  <value>
                    <array>
                      <data>
                        <value>
                          <struct>
                            <member>
                              <name>CODEERROR</name>
                              <value><int>111</int></value>
                            </member>
                            <member>
                              <name>NAMEERROR</name>
                              <value><string>Wrong parameters</string></value>
                            </member>
                            <member>
                              <name>DESCRIPTIONERROR</name>
                              <value><string>Ошибка переданных параметров</string></value>
                            </member>
                          </struct>
                        </value>
                      </data>
                    </array>
                  </value>
                </member>
                <member>
                  <name>CODEERROR</name>
                  <value><int>112</int></value>
                </member>
                <member>
                  <name>NAMEERROR</name>
                  <value><string>Not enough rights</string></value>
                </member>
              </struct>
            </value>
          </member>
        </struct>
      </value>
    </param>
  </params>
</methodResponse>
```

```

</member>
<member>
  <name>DESCRIPTIONERROR</name>
  <value><string>Не достаточно полномочий</string></value>
</member>
</struct>
</value>

<value>
<struct>
  <member>
    <name>CODEERROR</name>
    <value><int>113</int></value>
  </member>
  <member>
    <name>NAMEERROR</name>
    <value><string>Client has to be subscribed</string></value>
  </member>
  <member>
    <name>DESCRIPTIONERROR</name>
    <value><string>Клиент не подписан на указанную операцию</string></value>
  </member>
</struct>
</value>

<value>
<struct>
  <member>
    <name>CODEERROR</name>
    <value><int>114</int></value>
  </member>
  <member>
    <name>NAMEERROR</name>
    <value><string>Unknown method</string></value>
  </member>
  <member>
    <name>DESCRIPTIONERROR</name>
    <value><string>Неизвестный метод</string></value>
  </member>
</struct>
</value>

<value>
<struct>
  <member>
    <name>CODEERROR</name>
    <value><int>115</int></value>
  </member>
  <member>
    <name>NAMEERROR</name>
    <value><string>Unknown attribute type</string></value>
  </member>
  <member>
    <name>DESCRIPTIONERROR</name>
    <value><string>Неизвестный тип атрибута</string></value>
  </member>
</struct>
</value>

<value>
<struct>
  <member>
    <name>CODEERROR</name>
    <value><int>116</int></value>
  </member>
  <member>
    <name>NAMEERROR</name>
    <value><string>Range check error</string></value>

```

```

</member>
<member>
  <name>DESCRIPTIONERROR</name>
  <value><string>Значение за пределами допустимого</string></value>
</member>
</struct>
</value>

<value>
<struct>
  <member>
    <name>CODEERROR</name>
    <value><int>117</int></value>
  </member>
  <member>
    <name>NAMEERROR</name>
    <value><string>Resource is not accessible</string></value>
  </member>
  <member>
    <name>DESCRIPTIONERROR</name>
    <value><string>Ресурс недоступен</string></value>
  </member>
</struct>
</value>

<value>
<struct>
  <member>
    <name>CODEERROR</name>
    <value><int>118</int></value>
  </member>
  <member>
    <name>NAMEERROR</name>
    <value><string>Authorization required</string></value>
  </member>
  <member>
    <name>DESCRIPTIONERROR</name>
    <value><string>Не авторизованный вызов метода</string></value>
  </member>
</struct>
</value>

<value>
<struct>
  <member>
    <name>CODEERROR</name>
    <value><int>119</int></value>
  </member>
  <member>
    <name>NAMEERROR</name>
    <value><string>Obligatory parameter required</string></value>
  </member>
  <member>
    <name>DESCRIPTIONERROR</name>
    <value><string>Нет обязательных тегов</string></value>
  </member>
</struct>
</value>

<value>
<struct>
  <member>
    <name>CODEERROR</name>
    <value><int>120</int></value>
  </member>
  <member>
    <name>NAMEERROR</name>
    <value><string>Client port or address is not initialized</string></value>
  </member>

```

```

</member>
<member>
  <name>DESCRIPTIONERROR</name>
  <value><string>Не инициализирован порт или адрес клиента</string></value>
</member>
</struct>
</value>

<value>
<struct>
  <member>
    <name>CODEERROR</name>
    <value><int>121</int></value>
  </member>
  <member>
    <name>NAMEERROR</name>
    <value><string>Port can not be open</string></value>
  </member>
  <member>
    <name>DESCRIPTIONERROR</name>
    <value><string>Порт не может быть открыт</string></value>
  </member>
</struct>
</value>

<value>
<struct>
  <member>
    <name>CODEERROR</name>
    <value><int>122</int></value>
  </member>
  <member>
    <name>NAMEERROR</name>
    <value><string>Server port and address unknown</string></value>
  </member>
  <member>
    <name>DESCRIPTIONERROR</name>
    <value><string>Не определен порт и адрес сервера для которого будет возвращен результат</string></value>
  </member>
</struct>
</value>

<value>
<struct>
  <member>
    <name>CODEERROR</name>
    <value><int>123</int></value>
  </member>
  <member>
    <name>NAMEERROR</name>
    <value><string>Device is unknown for HwServer</string></value>
  </member>
  <member>
    <name>DESCRIPTIONERROR</name>
    <value><string>Устройство не добавлено в сервер</string></value>
  </member>
</struct>
</value>

<value>
<struct>
  <member>
    <name>CODEERROR</name>
    <value><int>124</int></value>
  </member>
  <member>
    <name>NAMEERROR</name>
    <value><string>Port is not active</string></value>
  </member>

```

```

</member>
<member>
  <name>DESCRIPTIONERROR</name>
  <value><string>Порт не активен</string></value>
</member>
</struct>
</value>

<value>
<struct>
  <member>
    <name>CODEERROR</name>
    <value><int>125</int></value>
  </member>
  <member>
    <name>NAMEERROR</name>
    <value><string>Result method is undefined</string></value>
  </member>
  <member>
    <name>DESCRIPTIONERROR</name>
    <value><string>Не определен метод возврата результата для асинхронного вызова</string></value>
  </member>
</struct>
</value>

<value>
<struct>
  <member>
    <name>CODEERROR</name>
    <value><int>126</int></value>
  </member>
  <member>
    <name>NAMEERROR</name>
    <value><string>HWServer is not configured</string></value>
  </member>
  <member>
    <name>DESCRIPTIONERROR</name>
    <value><string>Опросчик не сконфигурирован</string></value>
  </member>
</struct>
</value>

<value>
<struct>
  <member>
    <name>CODEERROR</name>
    <value><int>127</int></value>
  </member>
  <member>
    <name>NAMEERROR</name>
    <value><string>Query can not be completed</string></value>
  </member>
  <member>
    <name>DESCRIPTIONERROR</name>
    <value><string>Запрос не может быть выполнен</string></value>
  </member>
</struct>
</value>

<value>
<struct>
  <member>
    <name>CODEERROR</name>
    <value><int>128</int></value>
  </member>
  <member>
    <name>NAMEERROR</name>
    <value><string>GUID is not correct</string></value>
  </member>

```



```

</member>
<member>
  <name>DESCRIPTIONERROR</name>
  <value><string>Не верный жетон безопасности</string></value>
</member>
</struct>
</value>

<value>
<struct>
  <member>
    <name>CODEERROR</name>
    <value><int>129</int></value>
  </member>
  <member>
    <name>NAMEERROR</name>
    <value><string>RsCommand parameters are not correct</string></value>
  </member>
  <member>
    <name>DESCRIPTIONERROR</name>
    <value><string>Параметры для команды устройству не корректны</string></value>
  </member>
</struct>
</value>

<value>
<struct>
  <member>
    <name>CODEERROR</name>
    <value><int>130</int></value>
  </member>
  <member>
    <name>NAMEERROR</name>
    <value><string>Answer type is not correct</string></value>
  </member>
  <member>
    <name>DESCRIPTIONERROR</name>
    <value><string>Не корректный тип ответа</string></value>
  </member>
</struct>
</value>

<value>
<struct>
  <member>
    <name>CODEERROR</name>
    <value><int>131</int></value>
  </member>
  <member>
    <name>NAMEERROR</name>
    <value><string>Unknown answer on command</string></value>
  </member>
  <member>
    <name>DESCRIPTIONERROR</name>
    <value><string>Не известный ответ на команду</string></value>
  </member>
</struct>
</value>

<value>
<struct>
  <member>
    <name>CODEERROR</name>
    <value><int>132</int></value>
  </member>
  <member>
    <name>NAMEERROR</name>
    <value><string>Wrong answer length</string></value>
  </member>

```

```
</member>
<member>
  <name>DESCRIPTIONERROR</name>
  <value><string>Не верная длина ответа</string></value>
</member>
</struct>
</value>
```

```
<value>
<struct>
  <member>
    <name>CODEERROR</name>
    <value><int>133</int></value>
  </member>
  <member>
    <name>NAMEERROR</name>
    <value><string>No connection to device</string></value>
  </member>
  <member>
    <name>DESCRIPTIONERROR</name>
    <value><string>Нет связи с устройством</string></value>
  </member>
</struct>
</value>
```

```
<value>
<struct>
  <member>
    <name>CODEERROR</name>
    <value><int>134</int></value>
  </member>
  <member>
    <name>NAMEERROR</name>
    <value><string>Answer is too long</string></value>
  </member>
  <member>
    <name>DESCRIPTIONERROR</name>
    <value><string>Слишком большая длина ответа</string></value>
  </member>
</struct>
</value>
```

```
<value>
<struct>
  <member>
    <name>CODEERROR</name>
    <value><int>135</int></value>
  </member>
  <member>
    <name>NAMEERROR</name>
    <value><string>No answer from device</string></value>
  </member>
  <member>
    <name>DESCRIPTIONERROR</name>
    <value><string>Устройство не ответило на запрос</string></value>
  </member>
</struct>
</value>
```

```
<value>
<struct>
  <member>
    <name>CODEERROR</name>
    <value><int>136</int></value>
  </member>
  <member>
    <name>NAMEERROR</name>
    <value><string>Answer is not correct</string></value>
```

```

</member>
<member>
  <name>DESCRIPTIONERROR</name>
  <value><string>Не корректный ответ на команду</string></value>
</member>
</struct>
</value>

<value>
<struct>
  <member>
    <name>CODEERROR</name>
    <value><int>386</int></value>
  </member>
  <member>
    <name>NAMEERROR</name>
    <value><string>Controlled device is not subscribed</string></value>
  </member>
  <member>
    <name>DESCRIPTIONERROR</name>
    <value><string>Управление на устройстве которое не подписано</string></value>
  </member>
</struct>
</value>

<value>
<struct>
  <member>
    <name>CODEERROR</name>
    <value><int>387</int></value>
  </member>
  <member>
    <name>NAMEERROR</name>
    <value><string>Address is not correct</string></value>
  </member>
  <member>
    <name>DESCRIPTIONERROR</name>
    <value><string>Не корректный адрес устройства</string></value>
  </member>
</struct>
</value>

<value>
<struct>
  <member>
    <name>CODEERROR</name>
    <value><int>388</int></value>
  </member>
  <member>
    <name>NAMEERROR</name>
    <value><string>Method can not be executed</string></value>
  </member>
  <member>
    <name>DESCRIPTIONERROR</name>
    <value><string>Метод не может быть выполнен</string></value>
  </member>
</struct>
</value>

<value>
<struct>
  <member>
    <name>CODEERROR</name>
    <value><int>389</int></value>
  </member>

  <member>
    <name>NAMEERROR</name>

```

```

    <value><string>Control word required</string></value>
  </member>
  <member>
    <name>DESCRIPTIONERROR</name>
    <value><string>Нет управляющего слова</string></value>
  </member>
</struct>
</value>

</data>
</array>
</value>

</member>
</struct>
</value>

</member>
</struct>
</value>

</param>

</params>

</methodResponse>

```

Параметры метода *GetListRsCommand*:

- **INT64 GUID** – жетон безопасности;

Параметры структуры **RESULTDATA** для ответа:

- **STRING METHODNAME** – имя метода на который был дан ответ = «GetListRsCommand»;
- **STRUCT RSDESCRIPTION** – структура с описанием команд и параметров RSCommand.
-

Параметры метода *GetInfo*

- **INT64 GUID** – жетон безопасности;

Параметры структуры **RESULTDATA** ответа:

- **STRING METHODNAME** – имя метода на который был дан ответ = «GetInfo»;
- **BASE64 LOGOCOMPANY** – зашифрованное в BASE64 файл формата bmp с логотипом компании;
- **STRUCT RSXMLSERVER** – структура с данными в которых хранятся атрибуты «Модуля цправления».
 - **STRING FILETYPE** – тип файла;
 - **STRING COMPANYNAME** – имя компании;
 - **STRING FILEDESCRIPTION** – описание файла;
 - **STRING FILEVERSION** – версия файла.

Параметры метода *GetDeviceListAsync*:

Вызов содержит один параметр типа STRUCT(структура), в котором передаются следующие данные:

- **STRING IPSERVER** – Ip адрес сервера, на который необходимо передать результат исполнения команды;
- **INT64 PORTSERVER** – порт сервера, на который необходимо передать результат исполнения команды;
- **STRING MethodNameForAnswer** – имя метода на вызывающей стороне для передачи результата исполнения команды;

Пример вызова метода *GetDeviceListAsync*:

```
<?xml version="1.0" encoding="windows-1251"?>
<methodCall>
  <methodName>GetDeviceListAsync</methodName>
  <params>
    <param>
      <value>
        <struct>
          <member>
            <name>IPSERVER</name>
            <value>
              <string>127.0.0.1</string>
            </value>
          </member>
          <member>
            <name>MethodNameForAnswer</name>
            <value>
              <string>xmlRpcResult</string>
            </value>
          </member>
          <member>
            <name>PORTSERVER</name>
            <value>
              <int>8082</int>
            </value>
          </member>
        </struct>
      </value>
    </param>
  </params>
</methodCall>
```

Ответ на вызов *GetDeviceListAsync*:

- **ARRAY ComPortList** – список COM-портов. Элемент списка является структурой данных (STRUCT);
 - ⤴ **ARRAY DeviceList** – список подключенных пультов. Элемент списка является структурой данных (STRUCT).
 - ⤴ **INT64 DeviceAddress** – адрес пульта по RS-232;
 - ⤴ **INT64 DeviceType** – тип устройства (см. таблицу 1 в приложении А);
 - ⤴ **INT64 DeviceVersion** – версия устройства;
 - ⤴ **INT State** – состояние устройства (см. таблицу 2 в приложении А);
 - ⤴ **ARRAY DeviceList** – список подключенных к пульту устройств. Элемент списка является структурой данных (STRUCT).
 - ⤴ **INT64 DeviceAddress** – адрес прибора по RS-485;

- ✧ **INT64 DeviceType** – тип устройства (см. таблицу 1 в приложении А);
- ✧ **INT64 DeviceVersion** – версия устройства;
- ✧ **INT State** — состояние устройства (см. таблицу 2 в приложении А);
- ✧ **ARRAY ShleifList** – список шлейфов устройства. Элемент списка является структурой данных (STRUCT);
 - ✧ **INT64 ID** – идентификатор шлейфа;
 - ✧ **INT64 Address** – номер шлейфа;
 - ✧ **INT State** – состояние шлейфа (см. таблицу 2 в приложении А);
- ✧ **ARRAY RelayList** – список реле устройства. Элемент списка является структурой данных (STRUCT);
 - ✧ **INT64 ID** – идентификатор реле;
 - ✧ **INT64 Address** – номер реле;
 - ✧ **INT State** – состояние реле (см. таблицу 2 в приложении А);
- ✧ **ARRAY ReaderList** – список считывателей устройства. Элемент списка является структурой данных (STRUCT);
 - ✧ **INT64 ID** – идентификатор считывателя;
 - ✧ **INT64 Address** – номер считывателя;
 - ✧ **INT State** – состояние считывателя (см. таблицу 2 в приложении А);
- ✧ **INT64 ComPort** – номер Com-порта;

Пример ответа на вызов GetDeviceListAsync:

```
<?xml version="1.0" encoding="windows-1251"?>
<methodCall>
  <methodName>xmlRpcResult</methodName>
  <params>
    <param>
      <value>
        <struct>
          <member>
            <name>ComPortList</name>
            <value>
              <array>
                <data>
                  <value>
                    <struct>
                      <member>
                        <name>ComPort</name>
                        <value>
                          <int>5</int>
                        </value>
                      </member>
                    </struct>
                  </value>
                </data>
              </array>
            </value>
          </member>
          <member>
            <name>DeviceList</name>
            <value>
              <array>
                <data>
                  <value>
                    <struct>
                      <member>
                        <name>DeviceAddress</name>
                        <value>
                          <int>127</int>
                        </value>
                      </member>
                      <member>
                        <name>DeviceType</name>
                        <value>
```

```

    <int>0</int>
  </value>
</member>
<member>
  <name>DeviceVersion</name>
  <value>
    <int>204</int>
  </value>
</member>
<member>
  <name>DeviceList</name>
  <value>
    <array>
      <data>
        <value>
          <struct>
            <member>
              <name>
                DeviceAddress</name>
              <value>
                <int>5</int>
              </value>
            </member>
            <member>
              <name>
                DeviceType</name>
              <value>
                <int>9</int>
              </value>
            </member>
            <member>
              <name>
                DeviceVersion</name>
              <value>
                <int>136</int>
              </value>
            </member>
            <member>
              <name>
                ReaderList</name>
              <value>
                <array>
                  <data>
                    <value>
                      <struct>
                        <member>
                          <name>
                            ID</name>
                          <value>
                            <int>
                              0</int>
                          </value>
                        </member>
                        <member>
                          <name>
                            Address</name>
                          <value>
                            <int>
                              1</int>
                          </value>
                        </member>
                      </struct>
                    </value>
                  </data>
                </array>
              </value>
            </member>
          </struct>
        </data>
      </array>
    </value>
  </member>
</struct>

```

```

</value>
<value>
<struct>
  <member>
    <name>
      DeviceAddress</name>
    <value>
      <int>8</int>
    </value>
  </member>
  <member>
    <name>
      DeviceType</name>
    <value>
      <int>10</int>
    </value>
  </member>
  <member>
    <name>
      DeviceVersion</name>
    <value>
      <int>220</int>
    </value>
  </member>
  <member>
    <name>
      ReaderList</name>
    <value>
      <array>
        <data>
          <value>
            <struct>
              <member>
                <name>
                  ID</name>
                <value>
                  <int>
                    0</int>
                </value>
              </member>
              <member>
                <name>
                  Address</name>
                <value>
                  <int>
                    1</int>
                </value>
              </member>
            </struct>
          </value>
        </data>
      </array>
    </value>
  </member>
</struct>
</value>
</data>
</array>
</value>
</member>
</struct>
</value>
</data>
</array>
</value>
</member>
</struct>
</value>

```



```

        </data>
    </array>
</value>
</member>
<member>
    <name>IPSERVER</name>
    <value>
        <string>192.168.10.127</string>
    </value>
</member>
<member>
    <name>MethodName</name>
    <value>
        <string>GetDeviceList</string>
    </value>
</member>
<member>
    <name>Result</name>
    <value>
        <boolean>1</boolean>
    </value>
</member>
<member>
    <name>MessageType</name>
    <value>
        <int>2</int>
    </value>
</member>
<member>
    <name>Query</name>
    <value>
        <struct>
            <member>
                <name>IPSERVER</name>
                <value>
                    <string>127.0.0.1</string>
                </value>
            </member>
            <member>
                <name>MethodNameForAnswer</name>
                <value>
                    <string>xmlRpcResult</string>
                </value>
            </member>
            <member>
                <name>PORTSERVER</name>
                <value>
                    <int>8082</int>
                </value>
            </member>
        </struct>
    </value>
</member>
</struct>
</value>
</param>
</params>
</methodCall>

```

Параметры метода *GetDeviceList*:

Вызов содержит параметры, абсолютно идентичные асинхронному вызову GetDeviceListAsync.

Для загрузки в модуль управления данными, необходимых для конфигурирования СКУД и управления взятием/снятием используется метод LoadConfData.

Параметры метода LoadConfData (LoadAccessData):

Вызов содержит один параметр типа STRUCT(структура), в котором передаются следующие данные:

- **STRING GUID** – жетон безопасности;
- **ARRAY AccessZones** – список зон доступа. Элемент списка является структурой данных (STRUCT).
 - **INT ID** – ID зоны доступа. значения от 1 до $2^{31}-1$.
 - **STRING Name** – название зоны доступа.
- **ARRAY AccessPoints** – список точек доступа. Элемент списка является структурой данных (STRUCT).
 - **INT ID** – ID точки доступа. Значения от 1 до $2^{31}-1$.
 - **STRING NAME** – название точки доступа.
 - **INT PointType** – тип точки доступа.
 - 1 – однонаправленная дверь;
 - 2 – одна дверь на вход/выход;
 - 3 – турникет;
 - 4 – шлагбаун;
 - 5 – шлюз.
 - **INT InKeyID** – реле на вход. Значения от 1 до $2^{31}-1$.
 - **INT InDuration** – время управления на вход.
 - **INT InCommand** – команда на вход.
 - 0 – включить;
 - 1 – выключить.
 - **INT OutKeyID** – ID реле на выход. Значения от 1 до $2^{31}-1$.
 - **INT OutDuration** – время управления на выход.
 - **INT OutCommand** – команда на выход.
 - 0 – включить;
 - 1 – выключить.
 - **INT IndexZone1** – ID зоны доступа на вход. Значения от 1 до $2^{31}-1$.
 - **INT IndexZone2** – ID зоны доступа на выход. Значения от 1 до $2^{31}-1$.
 - **INT Mode** – режим работы точки доступа.
 - 0 – проход;
 - 1 – вход;
 - 2 – выход;
 - 3 – вход/выход.
- **ARRAY Groups** – список групп доступа. Элемент списка является структурой данных (STRUCT).
 - **INT ID** – ID группы доступа. Значения от 1 до $2^{31}-1$.

- **STRING Name** – имя группы доступа.
- **ARRAY Parts** – список разделов для взятия/снятия. Элемент списка является структурой данных (STRUCT).
 - **INT ID** – ID раздела. Значения от 1 до $2^{31}-1$.
 - **STRING Name** – имя раздела.
 - **INT Config** – является ли раздел особо охраняемым.
 - 0 – нет;
 - 1 – да.
 - **ARRAY Objects** – состав раздела. Элемент списка является структурой данных (STRUCT).
 - **INT ID** – ID элемента. Значения от 1 до $2^{31}-1$.
 - **INT Type** – тип элемента.
 - 3 – шлейф.
 - 9 – реле.
- **ARRAY PartGroups** – список групп разделов для взятия/снятия. Элемент списка является структурой данных (STRUCT).
 - **INT ID** – ID группы разделов. Значения от 1 до $2^{31}-1$.
 - **ARRAY Objects** – состав группы разделов. Элемент списка является структурой данных (STRUCT).
 - **INT ID** – ID элемента. Значения от 1 до $2^{31}-1$.
- **ARRAY Passwords** – список ключей. Элемент списка является структурой данных (STRUCT).
 - **INT ID** – ID ключа.
 - значения от 1 до $2^{31}-1$.
 - **INT CodeType** – тип кода.
 - 2 – PIN;
 - 3 – TouchMemory;
 - 4 – Proximity.
 - **INT AddCodeType** – тип кода.
 - 2 – PIN;
 - 3 – TouchMemory;
 - 4 – Proximity.
 - **INT Config** – параметры ключа.
 - бит 7 – хранить в приборе;
 - бит 15 – ключ заблокирован;
 - бит 16 – с дополнительным кодом;
 - бит 17-19 – тип ключа (три бита).
 - 0 – основной;
 - 3 – мастер;
 - 4 – открывающий;
 - 5 – закрывающий.
 - бит 30 – хранить в пульте C2000;

- **ARRAY Code** – код ключа. Каждый элемент массива представляет собой 1 байт данных.
- **ARRAY AddCode** – дополнительный код ключа. Каждый элемент массива представляет собой 1 байт данных.
- **INT Owner** – ID владельца ключа. Значения от 1 до $2^{31}-1$.
- **INT GroupID** – ID группы доступа. Значения от 1 до $2^{31}-1$.
- **STRING Start** – дата начала действия ключа (“12.11.2010 15:38:49”).
- **STRING Finish** – дата окончания действия ключа (“12.11.2010 15:38:49”).
- **ARRAY Persons** – список сотрудников. Элемент списка является структурой данных (STRUCT).
 - **INT ID** – ID сотрудника. Значения от 1 до $2^{31}-1$.
 - **STRING Name** – фамилия сотрудника.
 - **STRING FirstName** – имя сотрудника.
 - **STRING MidName** – отчество сотрудника.
- **ARRAY GroupAccesses** – список уровней доступа. Элемент списка является структурой данных (STRUCT).
 - **INT ID** – ID уровня доступа. Значения от 1 до $2^{31}-1$.
 - **INT GroupID** – ID группы доступа. Значения от 1 до $2^{31}-1$.
 - **INT Mode** – тип элемента доступа.
 - 2 – раздел;
 - 11 – зона доступа;
 - 12 – точка доступа;
 - 17 – группа разделов.
 - **INT AccessID** – ID элемента доступа. Значения от 1 до $2^{31}-1$.
 - 0 – все элементы заданного типа доступа.
 - **INT Config** – режим.
 - Для зон доступа и точек доступа:
 - 0 – проход;
 - 1 – вход;
 - 2 – выход;
 - 3 – вход/выход.
 - Для разделов и групп разделов:
 - 0 – просмотр;
 - 1 – взятие;
 - 2 – снятие;
 - 3 – взятие/снятие.
 - **INT Antipassback** – запрет повторного прохода в зону доступа.
 - 0 – нет;
 - 1 – строгий;
 - 2 – временной;
 - 3 – мягкий.
 - **STRING PardonTime** – время временного запрета повторного прохода в зону

- доступа.
- **INT ConfirmID** – режим прохода по правилу двух лиц.
0 – нет;
значение ID группы доступа.
 - **INT ConfirmID2** – режим прохода по правилу трех лиц.
0 – нет;
значение ID группы доступа.
 - **INT Flags** – доп. параметры.
бит 0 – подтверждающий режим прохода.
бит 10 – зональный antipassback.
 - **INT TimeZone** – ID окна времени. Значения от 1 до $2^{31}-1$.
0 – всегда.
- **ARRAY TimeWindows** – список временных окон. Элемент списка является структурой данных (STRUCT).
 - **INT ID** – ID окна времени. Значения от 1 до $2^{31}-1$.
 - **STRING NAME** – название окна времени.
 - **ARRAY Calendar** – календарь.
Каждый элемент массива представляет собой 1 байт данных.
Каждому месяцу соответствует 16 байт данных.
Каждый байт содержит информацию о двух днях месяца: младший полубайт — нечетный день месяца, старший полубайт — четный день месяца.
В календаре для каждого дня года указывается номер дня в «неделе», который может принимать значения:
0..13 – день «недели» (0 = Пн, 1 = Вт, 2 = Ср, ..., 6 = Вс);
14 – «праздник»;
15 – день соответствует дню недели по календарю (не переопределен).
 - **ARRAY TimeIntervals** – список временных интервалов. Элемент списка является структурой данных (STRUCT).
 - **INT ID** – ID временного интервала. Значения от 1 до $2^{31}-1$.
 - **INT TimeID** – ID окна времени. Значения от 1 до $2^{31}-1$.
 - **STRING Start** – начало временного интервала (“12.11.2010 15:38:49”).
 - **STRING Finish** – конец временного интервала (“12.11.2010 15:38:49”).
 - **INT Days** – битовая маска активности интервалов на 15 дней.
 - **BOOL EnterActivity** – активность интервала на вход.
 - **BOOL ExitActivity** – активность интервала на выход.
 - **ARRAY RdrAccessPoints** – список структур, связывающих считыватели с точками доступа/разделами. Элемент списка является структурой данных (STRUCT).
 - **INT ID** – ID связывающей структуры. Значения от 1 до $2^{31}-1$.
 - **INT ReaderID** – ID считывателя. Значения от 1 до $2^{31}-1$.

- **INT AccessPointID** – ID точки доступа/раздела. Значения от 1 до $2^{31}-1$.
- **INT AccessMode** – режим доступа.
 - 0 – проход;
 - 1 – вход;
 - 2 – выход.
- **INT Mode** – задает тип элемента, привязывающегося к считывателю.
 - 2 – раздел;
 - 11 – зона доступа;
 - 12 – точка доступа;
 - 17 – группа разделов.
- **ARRAY Relays** – список реле. Элемент списка является структурой данных (STRUCT).
 - **INT ID** – ID реле. Значения от 1 до $2^{31}-1$.
 - **STRUCT DeviceAddress** – адрес устройства.
 - **INT ComPort** – номер Com-порта.
 - **INT PKUAddress** – адрес пульта C2000.
 - **INT DeviceAddress** – адрес прибора.
 - **INT AggregateAddress** – адрес реле.
- **ARRAY Readers** – список считывателей. Элемент списка является структурой данных (STRUCT).
 - **INT ID** – ID считывателя. Значения от 1 до $2^{31}-1$.
 - **STRUCT DeviceAddress** – адрес устройства.
 - **INT ComPort** – номер Com-порта.
 - **INT PKUAddress** – адрес пульта C2000.
 - **INT DeviceAddress** – адрес прибора.
 - **INT AggregateAddress** – адрес считывателя.
- **ARRAY Shleifes** – список шлейфов. Элемент списка является структурой данных (STRUCT).
 - **INT ID** – ID шлейфа. Значения от 1 до $2^{31}-1$.
 - **STRUCT DeviceAddress** – адрес устройства.
 - **INT ComPort** – номер Com-порта.
 - **INT PKUAddress** – адрес пульта C2000.
 - **INT DeviceAddress** – адрес прибора.
 - **INT AggregateAddress** – адрес шлейфа.

Пример вызова LoadAccess:

```
<?xml version="1.0" encoding="windows-1251"?>
<methodCall>
  <methodName>LoadAccessData</methodName>
  <params>
    <param>
      <value>
        <struct>
```

```

<member>
  <name>AccessZones</name>
  <value>
    <array>
      <data>
        <value>
          <struct>
            <member>
              <name>ID</name>
              <value>
                <int>1</int>
              </value>
            </member>
            <member>
              <name>UserIndex</name>
              <value>
                <int>1</int>
              </value>
            </member>
            <member>
              <name>Name</name>
              <value>
                <string>
<string>
</string>
              </value>
            </member>
          </struct>
        </value>
        <value>
          <struct>
            <member>
              <name>ID</name>
              <value>
                <int>2</int>
              </value>
            </member>
            <member>
              <name>UserIndex</name>
              <value>
                <int>2</int>
              </value>
            </member>
            <member>
              <name>Name</name>
              <value>
                <string>
<string>
</string>
              </value>
            </member>
          </struct>
        </value>
        <value>
          <struct>
            <member>
              <name>ID</name>
              <value>
                <int>3</int>
              </value>
            </member>
            <member>
              <name>UserIndex</name>
              <value>
                <int>3</int>
              </value>
            </member>
            <member>
              <name>Name</name>
              <value>

```

```

        <string>
        </string>
    </value>
</member>
</struct>
</value>
<value>
<struct>
    <member>
        <name>ID</name>
        <value>
            <int>4</int>
        </value>
    </member>
    <member>
        <name>UserIndex</name>
        <value>
            <int>4</int>
        </value>
    </member>
    <member>
        <name>Name</name>
        <value>
            <string>
            </string>
        </value>
    </member>
</struct>
</value>
<value>
<struct>
    <member>
        <name>ID</name>
        <value>
            <int>5</int>
        </value>
    </member>
    <member>
        <name>UserIndex</name>
        <value>
            <int>5</int>
        </value>
    </member>
    <member>
        <name>Name</name>
        <value>
            <string>
            </string>
        </value>
    </member>
</struct>
</value>
<value>
<struct>
    <member>
        <name>ID</name>
        <value>
            <int>6</int>
        </value>
    </member>
    <member>
        <name>UserIndex</name>
        <value>
            <int>6</int>
        </value>
    </member>
    <member>
        <name>Name</name>

```



```

    <value>
      <string>
    </string>
  </value>
</member>
</struct>
</value>
<value>
  <struct>
    <member>
      <name>ID</name>
      <value>
        <int>7</int>
      </value>
    </member>
    <member>
      <name>UserIndex</name>
      <value>
        <int>7</int>
      </value>
    </member>
    <member>
      <name>Name</name>
      <value>
        <string>
      </string>
    </value>
  </member>
</struct>
</value>
<value>
  <struct>
    <member>
      <name>ID</name>
      <value>
        <int>8</int>
      </value>
    </member>
    <member>
      <name>UserIndex</name>
      <value>
        <int>8</int>
      </value>
    </member>
    <member>
      <name>Name</name>
      <value>
        <string>
      </string>
    </value>
  </member>
</struct>
</value>
<value>
  <struct>
    <member>
      <name>ID</name>
      <value>
        <int>9</int>
      </value>
    </member>
    <member>
      <name>UserIndex</name>
      <value>
        <int>9</int>
      </value>
    </member>
  </struct>

```

```

    <name>Name</name>
    <value>
      <string>
      </string>
    </value>
  </member>
</struct>
</value>
<value>
  <struct>
    <member>
      <name>ID</name>
      <value>
        <int>10</int>
      </value>
    </member>
    <member>
      <name>UserIndex</name>
      <value>
        <int>10</int>
      </value>
    </member>
    <member>
      <name>Name</name>
      <value>
        <string>
        </string>
      </value>
    </member>
  </struct>
</value>
</data>
</array>
</value>
</member>
<member>
  <name>AccessPoints</name>
  <value>
    <array>
      <data>
        <value>
          <struct>
            <member>
              <name>ID</name>
              <value>
                <int>1</int>
              </value>
            </member>
            <member>
              <name>ComputerID</name>
              <value>
                <int>1</int>
              </value>
            </member>
            <member>
              <name>Name</name>
              <value>
                <string>
                </string>
              </value>
            </member>
            <member>
              <name>UserIndex</name>
              <value>
                <int>0</int>
              </value>
            </member>
          </struct>
        </value>
      </data>
    </array>
  </value>
</member>

```

```

    <name>PointType</name>
    <value>
      <int>0</int>
    </value>
  </member>
  <member>
    <name>InKeyID</name>
    <value>
      <int>1</int>
    </value>
  </member>
  <member>
    <name>InDuration</name>
    <value>
      <int>2</int>
    </value>
  </member>
  <member>
    <name>InCommand</name>
    <value>
      <int>0</int>
    </value>
  </member>
  <member>
    <name>OutKeyID</name>
    <value>
      <int>2</int>
    </value>
  </member>
  <member>
    <name>OutDuration</name>
    <value>
      <int>2</int>
    </value>
  </member>
  <member>
    <name>OutCommand</name>
    <value>
      <int>0</int>
    </value>
  </member>
  <member>
    <name>IndexZone1</name>
    <value>
      <int>2</int>
    </value>
  </member>
  <member>
    <name>IndexZone2</name>
    <value>
      <int>4</int>
    </value>
  </member>
  <member>
    <name>Mode</name>
    <value>
      <int>0</int>
    </value>
  </member>
</struct>
</value>
</data>
</array>
</value>
</member>
<member>
  <name>Groups</name>
  <value>

```

```

<array>
  <data>
    <value>
      <struct>
        <member>
          <name>ID</name>
          <value>
            <int>1</int>
          </value>
        </member>
        <member>
          <name>Name</name>
          <value>
            <string>
            </string>
          </value>
        </member>
      </struct>
    </value>
    <value>
      <struct>
        <member>
          <name>ID</name>
          <value>
            <int>2</int>
          </value>
        </member>
        <member>
          <name>Name</name>
          <value>
            <string>
            </string>
          </value>
        </member>
      </struct>
    </value>
    <value>
      <struct>
        <member>
          <name>ID</name>
          <value>
            <int>3</int>
          </value>
        </member>
        <member>
          <name>Name</name>
          <value>
            <string>
            </string>
          </value>
        </member>
      </struct>
    </value>
  </data>
</array>
</value>
</member>
<member>
  <name>Passwords</name>
  <value>
    <array>
      <data>
        <value>
          <struct>
            <member>
              <name>ID</name>
              <value>
                <int>1</int>
              </value>
            </member>
          </struct>
        </value>
      </data>
    </array>
  </value>
</member>

```

```

    </value>
  </member>
  <member>
    <name>CodeType</name>
    <value>
      <int>4</int>
    </value>
  </member>
  <member>
    <name>AddCodeType</name>
    <value>
      <int>0</int>
    </value>
  </member>
  <member>
    <name>Config</name>
    <value>
      <int>128</int>
    </value>
  </member>
  <member>
    <name>Code</name>
    <value>
      <array>
        <data>
          <value>
            <int>1</int>
          </value>
          <value>
            <int>26</int>
          </value>
          <value>
            <int>105</int>
          </value>
          <value>
            <int>118</int>
          </value>
          <value>
            <int>0</int>
          </value>
          <value>
            <int>0</int>
          </value>
          <value>
            <int>0</int>
          </value>
          <value>
            <int>234</int>
          </value>
        </data>
      </array>
    </value>
  </member>
  <member>
    <name>AddCode</name>
  </member>
  <member>
    <name>Owner</name>
    <value>
      <int>1</int>
    </value>
  </member>
  <member>
    <name>GroupID</name>
    <value>
      <int>3</int>
    </value>
  </member>

```

```

    <member>
      <name>Start</name>
      <value>
        <string>23.04.2011 12:35:45</string>
      </value>
    </member>
    <member>
      <name>Finish</name>
      <value>
        <string>01.01.2040 00:00:00</string>
      </value>
    </member>
  </struct>
</value>
</data>
</array>
</value>
</member>
<member>
  <name>Persons</name>
  <value>
    <array>
      <data>
        <value>
          <struct>
            <member>
              <name>ID</name>
              <value>
                <int>1</int>
              </value>
            </member>
            <member>
              <name>Name</name>
              <value>
                <string>
                  </string>
                </value>
            </member>
            <member>
              <name>FirstName</name>
              <value>
                <string>
                  </string>
                </value>
            </member>
            <member>
              <name>MidName</name>
              <value>
                <string>
                  </string>
                </value>
            </member>
            <member>
              <name>Status</name>
              <value>
                <int>0</int>
              </value>
            </member>
            <member>
              <name>BirthDate</name>
              <value>
                <string>01.01.2001 00:00:00</string>
              </value>
            </member>
          </struct>
        </value>
      </data>
    </array>
  </value>
</member>

```

```

</value>
</member>
<member>
  <name>GroupAccesses</name>
  <value>
    <array>
      <data>
        <value>
          <struct>
            <member>
              <name>ID</name>
              <value>
                <int>1</int>
              </value>
            </member>
            <member>
              <name>GroupID</name>
              <value>
                <int>3</int>
              </value>
            </member>
            <member>
              <name>Mode</name>
              <value>
                <int>12</int>
              </value>
            </member>
            <member>
              <name>AccessID</name>
              <value>
                <int>1</int>
              </value>
            </member>
            <member>
              <name>TimeZone</name>
              <value>
                <int>1</int>
              </value>
            </member>
            <member>
              <name>Config</name>
              <value>
                <int>0</int>
              </value>
            </member>
            <member>
              <name>Antipassback</name>
              <value>
                <int>2</int>
              </value>
            </member>
            <member>
              <name>PardonTime</name>
              <value>
                <string>30.12.1899</string>
              </value>
            </member>
            <member>
              <name>ConfirmID</name>
              <value>
                <int>0</int>
              </value>
            </member>
            <member>
              <name>ConfirmID2</name>
              <value>
                <int>0</int>
              </value>
            </member>
          </struct>
        </value>
      </data>
    </array>
  </value>
</member>

```

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

```
</value>
<value>
  <int>0</int>
</value>
<value>
  <int>0</int>
</value>
<value>
  <int>0</int>
</value>
<value>
  <int>0</int>
</value>
<value>
  <int>0</int>
</value>
<value>
  <int>0</int>
</value>
<value>
  <int>0</int>
</value>
<value>
  <int>0</int>
</value>
<value>
  <int>0</int>
</value>
<value>
  <int>0</int>
</value>
<value>
  <int>0</int>
</value>
<value>
  <int>0</int>
</value>
<value>
  <int>0</int>
</value>
<value>
  <int>0</int>
</value>
<value>
  <int>0</int>
</value>
<value>
  <int>0</int>
</value>
<value>
  <int>0</int>
</value>
</data>
</array>
</value>
</member>
</string>
<value>
</member>
</struct>
</value>
</data>
</array>
</value>
</member>
<member>
<name>TimeIntervals</name>
<value>
  <array>
    <data>
      <value>
```



```

<struct>
  <member>
    <name>ID</name>
    <value>
      <int>0</int>
    </value>
  </member>
  <member>
    <name>TimeID</name>
    <value>
      <int>1</int>
    </value>
  </member>
  <member>
    <name>Number</name>
    <value>
      <int>0</int>
    </value>
  </member>
  <member>
    <name>Start</name>
    <value>
      <string>8:00:00</string>
    </value>
  </member>
  <member>
    <name>Finish</name>
    <value>
      <string>23:00:00</string>
    </value>
  </member>
  <member>
    <name>Days</name>
    <value>
      <int>16383</int>
    </value>
  </member>
  <member>
    <name>EnterActivity</name>
    <value>
      <int>1</int>
    </value>
  </member>
  <member>
    <name>ExitActivity</name>
    <value>
      <int>1</int>
    </value>
  </member>
</struct>
</value>
</data>
</array>
</value>
</member>
<member>
  <name>RdrAccessPoints</name>
  <value>
    <array>
      <data>
        <value>
          <struct>
            <member>
              <name>ID</name>
              <value>
                <int>1</int>
              </value>
            </member>

```

```

    <member>
      <name>ReaderID</name>
      <value>
        <int>1</int>
      </value>
    </member>
    <member>
      <name>AccessPointID</name>
      <value>
        <int>1</int>
      </value>
    </member>
    <member>
      <name>AccessMode</name>
      <value>
        <int>0</int>
      </value>
    </member>
  </struct>
</value>
<value>
  <struct>
    <member>
      <name>ID</name>
      <value>
        <int>2</int>
      </value>
    </member>
    <member>
      <name>ReaderID</name>
      <value>
        <int>2</int>
      </value>
    </member>
    <member>
      <name>AccessPointID</name>
      <value>
        <int>1</int>
      </value>
    </member>
    <member>
      <name>AccessMode</name>
      <value>
        <int>0</int>
      </value>
    </member>
  </struct>
</value>
</data>
</array>
</value>
</member>
<member>
  <name>Relays</name>
  <value>
    <array>
      <data>
        <value>
          <struct>
            <member>
              <name>ID</name>
              <value>
                <int>1</int>
              </value>
            </member>
            <member>
              <name>DeviceAddress</name>
              <value>

```

```

<struct>
  <member>
    <name>ComPort</name>
    <value>
      <int>5</int>
    </value>
  </member>
  <member>
    <name>PKUAddress</name>
    <value>
      <int>127</int>
    </value>
  </member>
  <member>
    <name>DeviceAddress</name>
    <value>
      <int>6</int>
    </value>
  </member>
  <member>
    <name>AggregateAddress</name>
    <value>
      <int>1</int>
    </value>
  </member>
</struct>
</value>
<value>
  <struct>
    <member>
      <name>ID</name>
      <value>
        <int>2</int>
      </value>
    </member>
    <member>
      <name>DeviceAddress</name>
      <value>
        <struct>
          <member>
            <name>ComPort</name>
            <value>
              <int>5</int>
            </value>
          </member>
          <member>
            <name>PKUAddress</name>
            <value>
              <int>127</int>
            </value>
          </member>
          <member>
            <name>DeviceAddress</name>
            <value>
              <int>6</int>
            </value>
          </member>
          <member>
            <name>AggregateAddress</name>
            <value>
              <int>2</int>
            </value>
          </member>
        </struct>
      </value>
    </member>
  </struct>
</value>
<value>
  <struct>
    <member>

```

```

<name>ID</name>
<value>
  <int>3</int>
</value>
</member>
<member>
  <name>DeviceAddress</name>
  <value>
    <struct>
      <member>
        <name>ComPort</name>
        <value>
          <int>5</int>
        </value>
      </member>
      <member>
        <name>PKUAddress</name>
        <value>
          <int>127</int>
        </value>
      </member>
      <member>
        <name>
          DeviceAddress</name>
        <value>
          <int>4</int>
        </value>
      </member>
      <member>
        <name>
          AggregateAddress</name>
        <value>
          <int>1</int>
        </value>
      </member>
    </struct>
  </value>
</value>
<struct>
  <member>
    <name>ID</name>
    <value>
      <int>4</int>
    </value>
  </member>
  <member>
    <name>
      DeviceAddress</name>
    <value>
      <struct>
        <member>
          <name>
            ComPort</name>
          <value>
            <int>5</int>
          </value>
        </member>
        <member>
          <name>
            PKUAddress</name>
          <value>
            <int>127</int>
          </value>
        </member>
        <member>
          <name>
            DeviceAddress</name>
          <value>

```

```

        <int>4</int>
      </value>
    </member>
  <member>
    <name>
      AggregateAddress</name>
    <value>
      <int>2</int>
    </value>
  </member>
</struct>
</value>
</member>
<member>
  <name>Readers</name>
  <value>
    <array>
      <data>
        <value>
          <struct>
            <member>
              <name>
                ID</name>
              <value>
                <int>
                  1</int>
                </value>
            </member>
            <member>
              <name>
                DeviceAddress</name>
              <value>
                <struct>
                  <member>
                    <name>
                      ComPort</name>
                    <value>
                      <int>
                        5</int>
                      </value>
                  </member>
                  <member>
                    <name>
                      PKUAddress</name>
                    <value>
                      <int>
                        127</int>
                      </value>
                  </member>
                  <member>
                    <name>
                      DeviceAddress</name>
                    <value>
                      <int>
                        6</int>
                      </value>
                  </member>
                </struct>
              </value>
            </member>
            <member>
              <name>
                AggregateAddress</name>
              <value>
                <int>
                  1</int>
                </value>
            </member>
          </struct>
        </value>
      </data>
    </array>
  </value>
</member>

```

```

<struct>
  <member>
    <name>
      ID</name>
    <value>
      <int>
        2</int>
      </value>
    </member>
    <member>
      <name>
        DeviceAddress</name>
      <value>
        <struct>

          <member>

            <name>
              ComPort</name>
            <value>

              <int>
                5</int>
              </value>
            </member>
            <member>

              <name>
                PKUAddress</name>
              <value>

                <int>
                  127</int>
                </value>
              </member>
            </member>

            <name>
              DeviceAddress</name>
            <value>

              <int>
                6</int>
              </value>
            </member>
            <member>

              <name>
                AggregateAddress</name>
              <value>

                <int>
                  2</int>
                </value>
              </member>
            </member>
          </struct>
        </value>
      </member>
    </struct>
  </value>
</data>
</array>
</value>
</member>
</struct>

```

```

        </value>
      </member>
    </struct>
  </value>
</member>
</struct>
</value>
</member>
</struct>
</value>
</data>
</array>
</value>
</member>
</struct>
</value>
</param>
</params>
</methodCall>

```

Для того, чтобы синхронизировать ключ, находящийся в таблице модуля управления, с прибором используется метод `SynchronizeOneKey`.

Параметры метода `SynchronizeOneKey`:

Вызов содержит один параметр типа `STRUCT`(структура), в котором передаются следующие данные:

- **STRING GUID** – жетон безопасности;
- **STRING IPSEVER** – IP адрес сервера, на который необходимо передать результат исполнения команды;
- **INT64 PORTSERVER** – порт сервера, на который необходимо передать результат исполнения команды;
- **STRING MethodNameForAnswer** – имя метода на вызывающей стороне для передачи результата исполнения команды;
- **INT ID** – ID ключа из списка ключей (Passwords).
- **INT AutoWriting** – записывать окно времени и уровень доступа автоматом.
- **INT RewriteDeleted** – записывать ключ на месте первого найденного ключа с атрибутом «удален».
- **INT RewriteBlocked** – записывать ключ на месте первого найденного ключа с атрибутом «заблокирован».

Ответ на вызов `SynchronizeOneKey`:

Ответ содержит один параметр типа `STRUCT`(структура), в котором передаются следующие данные.

- **INT PasswordID** - ID ключа из списка ключей (Passwords).
- **BOOL PasswordStatus** - состояние ключа(TRUE – ключ везде записан правильно, FALSE – где-то записан неверно).
- **BOOL KeyCode** – код ключа.
- **INT PersonID** – ID владельца ключа.
- **STRING PersonName** – ФИО владельца ключа.

Для того, чтобы синхронизировать все ключи, находящиеся в таблице модуля управления с прибором, используется метод `SynchronizeKeys`.

Параметры метода `SynchronizeKeys`:

Вызов содержит один параметр типа `STRUCT`(структура), в котором передаются следующие данные:

- **STRING GUID** – жетон безопасности;
- **STRING IPSERVER** – Ip адрес сервера, на который необходимо передать результат исполнения команды;
- **INT64 PORTSERVER** – порт сервера, на который необходимо передать результат исполнения команды;
- **STRING MethodNameForAnswer** – имя метода на вызывающей стороне для передачи результата исполнения команды;
- **INT AutoWriting** – записывать окно времени и уровень доступа автоматом.
- **INT RewriteDeleted** – записывать ключ на месте первого найденного ключа с атрибутом «удален».
- **INT RewriteBlocked** – записывать ключ на месте первого найденного ключа с атрибутом «заблокирован».

Пример вызова `SynchronizeKeys`:

```
<?xml version="1.0" encoding="windows-1251"?>
<methodCall>
  <methodName>SynchronizeKeys</methodName>
  <params>
    <param>
      <value>
        <struct>
          <member>
            <name>IPSERVER</name>
            <value>
              <string>127.0.0.1</string>
            </value>
          </member>
          <member>
            <name>PORTSERVER</name>
            <value>
              <int>8092</int>
            </value>
          </member>
          <member>
            <name>MethodNameForAnswer</name>
            <value>
              <string>SyncKeysResult</string>
            </value>
          </member>
          <member>
            <name>AutoWriting</name>
            <value>
              <int>1</int>
            </value>
          </member>
          <member>
            <name>RewriteDeleted</name>
            <value>
              <int>1</int>
            </value>
          </member>
        </struct>
      </value>
    </param>
  </params>
</methodCall>
```



```

</member>
<member>
  <name>RewriteBlocked</name>
  <value>
    <int>1</int>
  </value>
</member>
</struct>
</value>
</param>
</params>
</methodCall>

```

Ответ на вызов *SynchronizeKeys*:

Ответ содержит один параметр типа STRUCT(структура), в котором передаются следующие данные.

- **INT PasswordID** - ID ключа из списка ключей (Passwords).
- **BOOL PasswordStatus** - состояние ключа(TRUE – ключ везде записан правильно, FALSE – где-то записан неверно).
- **BOOL KeyCode** – код ключа.
- **INT PersonID** – ID владельца ключа.
- **STRING PersonName** – ФИО владельца ключа.

Пример ответа на вызов *SynchronizeKeys*:

```

<?xml version="1.0" encoding="windows-1251"?>
<methodCall>
  <methodName>SyncKeysResult</methodName>
  <params>
    <param>
      <value>
        <struct>
          <member>
            <name>Comment</name>
            <value>
              <base64></base64>
            </value>
          </member>
          <member>
            <name>Progress</name>
            <value>
              <int>100</int>
            </value>
          </member>
          <member>
            <name>PasswordID</name>
            <value>
              <int>1</int>
            </value>
          </member>
          <member>
            <name>PasswordStatus</name>
            <value>
              <boolean>1</boolean>
            </value>
          </member>
          <member>
            <name>KeyCode</name>
            <value>
              <string>EA00000076691A01</string>
            </value>
          </member>
          <member>
            <name>PersonID</name>

```

```

<value>
  <int>1</int>
</value>
</member>
<member>
  <name>PersonName</name>
  <value>
    <string>Гаврилин В.А.</string>
  </value>
</member>
<member>
  <name>CommandCount</name>
  <value>
    <int>18</int>
  </value>
</member>
<member>
  <name>CommandIndex</name>
  <value>
    <int>18</int>
  </value>
</member>
<member>
  <name>IPSERVER</name>
  <value>
    <string>192.168.10.127</string>
  </value>
</member>
<member>
  <name>MethodName</name>
  <value>
    <string>Uknown method</string>
  </value>
</member>
<member>
  <name>Result</name>
  <value>
    <boolean>0</boolean>
  </value>
</member>
<member>
  <name>Query</name>
  <value>
    <struct>
      <member>
        <name>IPSERVER</name>
        <value>
          <string>127.0.0.1</string>
        </value>
      </member>
      <member>
        <name>PORTSERVER</name>
        <value>
          <int>8092</int>
        </value>
      </member>
      <member>
        <name>MethodNameForAnswer</name>
        <value>
          <string>SyncKeysResult</string>
        </value>
      </member>
      <member>
        <name>AutoWriting</name>
        <value>
          <int>1</int>
        </value>
      </member>
    </struct>
  </value>
</member>

```

```

    <member>
      <name>RewriteDeleted</name>
      <value>
        <int>1</int>
      </value>
    </member>
    <member>
      <name>RewriteBlocked</name>
      <value>
        <int>1</int>
      </value>
    </member>
  </struct>
</value>
</member>
<member>
  <name>ComPort</name>
  <value>
    <int>5</int>
  </value>
</member>
<member>
  <name>PKUAddress</name>
  <value>
    <int>127</int>
  </value>
</member>
<member>
  <name>DeviceAddress</name>
  <value>
    <int>6</int>
  </value>
</member>
<member>
  <name>ID</name>
  <value>
    <int>-1</int>
  </value>
</member>
</struct>
</value>
</param>
</params>
</methodCall>

```

Для получения информации о хранении ключа используется метод `GetKeyFullInformation`.

Параметры метода `GetKeyFullInformation`:

Вызов содержит один параметр типа `STRUCT`(структура), в котором передаются следующие данные:

- **STRING GUID** – жетон безопасности;
- **STRING IPSERVER** – Ip адрес сервера, на который необходимо передать результат исполнения команды;
- **INT64 PORTSERVER** – порт сервера, на который необходимо передать результат исполнения команды;
- **STRING MethodNameForAnswer** – имя метода на вызывающей стороне для передачи результата исполнения команды;
- **INT ID** - ID ключа из списка ключей (Passwords).

Пример вызова GetKeyFullInformation:

```
<?xml version="1.0" encoding="windows-1251"?>
<methodCall>
  <methodName>GetKeyFullInformation</methodName>
  <params>
    <param>
      <value>
        <struct>
          <member>
            <name>ID</name>
            <value>
              <int>1</int>
            </value>
          </member>
          <member>
            <name>MethodNameForAnswer</name>
            <value>
              <string>GetKeyInfoResult</string>
            </value>
          </member>
          <member>
            <name>IPSERVER</name>
            <value>
              <string>192.168.10.127</string>
            </value>
          </member>
          <member>
            <name>PORTSERVER</name>
            <value>
              <int>8092</int>
            </value>
          </member>
        </struct>
      </value>
    </param>
  </params>
</methodCall>
```

Ответ на вызов GetKeyFullInformation:

Ответ содержит один параметр типа STRUCT(структура), в котором передаются следующие данные.

- **ARRAY DeviceList** – список приборов. Элемент списка является структурой данных (STRUCT).
 - **INT ID** – ID прибора.
 - **INT ComPort** – номер Com-порта.
 - **INT PKUAddress** – адрес пульта C2000.
 - **INT DeviceAddress** – адрес прибора.
 - **STRING Name** – название прибора.
 - **STRING DeviceVersion** – версия прибора.
 - **STRUCT InGroupInfo** – как должен храниться на основе уровня доступа.
 - **STRING TimeName** – название окна времени.
 - **INT Status** – статус.
 - **INT MustBeAccess** – должен работать для доступа.
 - **INT MustBeControl** – должен работать для управления.
- Только для C2000-4 версий до 1.20:
- **INT Shleif1** – взятие/снятие для ШС1.

- **INT Shleif2** – взятие/снятие для ШС2.
- **INT Shleif3** – взятие/снятие для ШС3.
- **INT Shleif4** – взятие/снятие для ШС4.

Только для С2000-4 версий до 2.0 и выше:

- **INT Shleif1** – взятие/снятие для ШС1.
- **INT Shleif2** – взятие/снятие для ШС2.
- **INT Shleif3** – взятие/снятие для ШС3.
- **INT Shleif4** – взятие/снятие для ШС4.
- **INT TimeNameControl** - окно времени для управления.

Только для С2000-2 :

- **INT PasswordType** – тип ключа.
- **INT Enter** – режим работы для входа.
- **INT Exit** – режим работы для выхода.
- **INT Antipassback** – тип запрета повторного прохода в зону доступа.
- **AntipassbackUnlockTime** – время запрета повторного прохода в зону доступа.
- **INT ConfirmGroupEnter1** – уровень подтверждения для входа.
- **INT ConfirmGroupEnter2** – уровень подтверждения для входа.
- **INT ConfirmGroupExit1** – уровень подтверждения для выхода.
- **INT ConfirmGroupExit2** – уровень подтверждения для выхода.
- **INT ZonalAntipassback** – зональный антипассбэк.
- **INT TimeNameControl** – окно времени для управления.
- **INT Shleif1** - взятие/снятие для ШС1.
- **INT Shleif2** - взятие/снятие для ШС2.
- **STRUCT InDeviceInfo** – как ключ записан в прибор.
 - **STRING TimeName** – название окна времени.
 - **INT Status** – статус.
 - **INT MustBeAccess** – должен работать для доступа.
 - **INT MustBeControl** – должен работать для управления.
 - **INT PasswordBlocked** – ключ заблокирован.
 - **STRING PasswordDateStart** – начало действия ключа.
 - **STRING PasswordDateFinish** – окончание действия ключа.

Только для С2000-4 версий до 1.20:

- **INT Shleif1** – взятие/снятие для ШС1.
- **INT Shleif2** – взятие/снятие для ШС2.
- **INT Shleif3** – взятие/снятие для ШС3.
- **INT Shleif4** – взятие/снятие для ШС4.

Только для С2000-4 версий до 2.0 и выше:

- **INT Shleif1** – взятие/снятие для ШС1.
- **INT Shleif2** – взятие/снятие для ШС2.
- **INT Shleif3** – взятие/снятие для ШС3.
- **INT Shleif4** – взятие/снятие для ШС4.
- **INT TimeNameControl** - окно времени для управления.

Только для C2000-2 :

- **INT PasswordType** – тип ключа.
- **INT Enter** – режим работы для входа.
- **INT Exit** – режим работы для выхода.
- **INT Antipassback** – тип запрета повторного прохода в зону доступа.
- **AntipassbackUnlockTime** – время запрета повторного прохода в зону доступа.
- **INT ConfirmGroupEnter1** – уровень подтверждения для входа.
- **INT ConfirmGroupEnter2** – уровень подтверждения для входа.
- **INT ConfirmGroupExit1** – уровень подтверждения для выхода.
- **INT ConfirmGroupExit2** – уровень подтверждения для выхода.
- **INT ZonalAntipassback** – зональный антипассбэк.
- **INT TimeNameControl** – окно времени для управления.
- **INT Shleif1** - взятие/снятие для ШС1.
- **INT Shleif2** - взятие/снятие для ШС2.

Пример ответа на вызов GetKeyFullInformation:

```
<?xml version="1.0" encoding="windows-1251"?>
<methodCall>
  <methodName>GetKeyInfoResult</methodName>
  <params>
    <param>
      <value>
        <struct>
          <member>
            <name>DeviceList</name>
            <value>
              <array>
                <data>
                  <value>
                    <struct>
                      <member>
                        <name>Status</name>
                        <value>
                          <boolean>0</boolean>
                        </value>
                      </member>
                      <member>
                        <name>Comment</name>
                        <value>
                          <string>Не записан уровень доступа</string>
                        </value>
                      </member>
                      <member>
                        <name>ID</name>
                        <value>
                          <int>-1</int>
                        </value>
                      </member>
                      <member>
                        <name>ComPort</name>
                        <value>
                          <int>5</int>
                        </value>
                      </member>
                      <member>
                        <name>PKUAddress</name>
                        <value>
```

```

    <int>127</int>
  </value>
</member>
<member>
  <name>DeviceAddress</name>
  <value>
    <int>6</int>
  </value>
</member>
<member>
  <name>Name</name>
  <value>
    <string>[5.127.6 - C2000-2]:</string>
  </value>
</member>
<member>
  <name>DeviceVersion</name>
  <value>
    <int>110</int>
  </value>
</member>
<member>
  <name>InGroupInfo</name>
  <value>
    <struct>
      <member>
        <name>PasswordBlocked</name>
        <value>
          <string></string>
        </value>
      </member>
      <member>
        <name>PasswordType</name>
        <value>
          <string>Основной</string>
        </value>
      </member>
      <member>
        <name>PasswordDateStart</name>
        <value>
          <string>23.04.2011</string>
        </value>
      </member>
      <member>
        <name>PasswordDateFinish</name>
        <value>
          <string>01.01.2040</string>
        </value>
      </member>
      <member>
        <name>MustBeAccess</name>
        <value>
          <string>Да</string>
        </value>
      </member>
      <member>
        <name>MustBeControl</name>
        <value>
          <string></string>
        </value>
      </member>
      <member>
        <name>Status</name>
        <value>
          <string>Должен храниться в приборе</string>
        </value>
      </member>
    </struct>
  </value>
</member>

```

```

    <name>Enter</name>
    <value>
      <string>Подтверждающий</string>
    </value>
  </member>
  <member>
    <name>Exit</name>
    <value>
      <string>Her</string>
    </value>
  </member>
  <member>
    <name>TimeName</name>
    <value>
      <string></string>
    </value>
  </member>
  <member>
    <name>Antipassback</name>
    <value>
      <string>Her</string>
    </value>
  </member>
  <member>
    <name>ZonalAntipassback</name>
    <value>
      <string></string>
    </value>
  </member>
</struct>
</value>
</member>
<member>
  <name>InDeviceInfo</name>
  <value>
    <struct>
      <member>
        <name>Status</name>
        <value>
          <string>Хранится в приборе</string>
        </value>
      </member>
      <member>
        <name>PasswordType</name>
        <value>
          <string>Основной</string>
        </value>
      </member>
      <member>
        <name>ZonalAntipassback</name>
        <value>
          <string></string>
        </value>
      </member>
      <member>
        <name>AdditionalCode</name>
        <value>
          <boolean>1</boolean>
        </value>
      </member>
      <member>
        <name>MustBeAccess</name>
        <value>
          <string>Да</string>
        </value>
      </member>
      <member>
        <name>MustBeControl</name>

```



```

    <value>
      <string></string>
    </value>
  </member>
  <member>
    <name>TimeName</name>
    <value>
      <string>Не найдено</string>
    </value>
  </member>
  <member>
    <name>Enter</name>
    <value>
      <string>Подтверждающий</string>
    </value>
  </member>
  <member>
    <name>Exit</name>
    <value>
      <string>Her</string>
    </value>
  </member>
  <member>
    <name>Antipassback</name>
    <value>
      <string>Her</string>
    </value>
  </member>
  <member>
    <name>PasswordBlocked</name>
    <value>
      <string></string>
    </value>
  </member>
  <member>
    <name>PasswordDateStart</name>
    <value>
      <string>23.04.2011</string>
    </value>
  </member>
  <member>
    <name>PasswordDateFinish</name>
    <value>
      <string>01.01.2040</string>
    </value>
  </member>
</struct>
</value>
</member>
</struct>
</value>
</data>
</array>
</value>
</member>
<member>
  <name>IPSERVER</name>
  <value>
    <string>192.168.10.127</string>
  </value>
</member>
<member>
  <name>MethodName</name>
  <value>
    <string>SetKeyFullInformation</string>
  </value>
</member>
</member>

```

```

<name>Result</name>
<value>
  <boolean>1</boolean>
</value>
</member>
<member>
  <name>MessageType</name>
  <value>
    <int>2</int>
  </value>
</member>
<member>
  <name>Query</name>
  <value>
    <struct>
      <member>
        <name>ID</name>
        <value>
          <int>1</int>
        </value>
      </member>
      <member>
        <name>MethodNameForAnswer</name>
        <value>
          <string>GetKeyInfoResult</string>
        </value>
      </member>
      <member>
        <name>IPSERVER</name>
        <value>
          <string>192.168.10.127</string>
        </value>
      </member>
      <member>
        <name>PORTSERVER</name>
        <value>
          <int>8092</int>
        </value>
      </member>
    </struct>
  </value>
</member>
</struct>
</value>
</param>
</params>
</methodCall>

```

Для получения информации о состоянии ключей используется метод GetPasswordListWithStatus.

Параметры метода GetPasswordListWithStatus:

Вызов содержит один параметр типа STRUCT(структура), в котором передаются следующие данные:

- **STRING GUID** – жетон безопасности;
- **STRING IPSERVER** – Ip адрес сервера, на который необходимо передать результат исполнения команды;
- **INT64 PORTSERVER** – порт сервера, на который необходимо передать результат исполнения команды;
- **STRING MethodNameForAnswer** – имя метода на вызывающей стороне для

передачи результата исполнения команды;

Пример вызова GetPasswordListWithStatus:

```
<?xml version="1.0" encoding="windows-1251"?>
<methodCall>
  <methodName>GetPasswordListWithStatus</methodName>
  <params>
    <param>
      <value>
        <struct>
          <member>
            <name>MethodNameForAnswer</name>
            <value>
              <string>GetPswListAsync</string>
            </value>
          </member>
          <member>
            <name>IPSERVER</name>
            <value>
              <string>127.0.0.1</string>
            </value>
          </member>
          <member>
            <name>PORTSERVER</name>
            <value>
              <int>8092</int>
            </value>
          </member>
        </struct>
      </value>
    </param>
  </params>
</methodCall>
```

Ответ на вызов GetPasswordListWithStatus:

Ответ содержит один параметр типа STRUCT(структура), в котором передаются следующие данные.

- **STRING GUID** – жетон безопасности;
- **ARRAY KeyList** – список ключей. Элемент списка является структурой данных (STRUCT).
 - ⤴ **INT PasswordID** - ID ключа из списка ключей (Passwords).
 - ⤴ **INT PasswordStatus** - состояние ключа(TRUE – ключ везде записан правильно, FALSE – где-то записан неверно).
 - ⤴ **STRING Comment** – текст ошибки.

Пример ответа на вызов GetPasswordListWithStatus:

```
<?xml version="1.0" encoding="windows-1251"?>
<methodCall>
  <methodName>GetPswListAsync</methodName>
  <params>
    <param>
      <value>
        <struct>
          <member>
            <name>KeyList</name>
            <value>
              <array>
                <data>
                  <value>
```

```

    <struct>
      <member>
        <name>PasswordID</name>
        <value>
          <int>1</int>
        </value>
      </member>
      <member>
        <name>PasswordStatus</name>
        <value>
          <boolean>0</boolean>
        </value>
      </member>
    </struct>
  </value>
</data>
</array>
</value>
</member>
<member>
  <name>IPSERVER</name>
  <value>
    <string>192.168.10.127</string>
  </value>
</member>
<member>
  <name>MethodName</name>
  <value>
    <string>GetPasswordListWithStatus</string>
  </value>
</member>
<member>
  <name>Result</name>
  <value>
    <boolean>1</boolean>
  </value>
</member>
<member>
  <name>MessageType</name>
  <value>
    <int>2</int>
  </value>
</member>
<member>
  <name>Query</name>
  <value>
    <struct>
      <member>
        <name>MethodNameForAnswer</name>
        <value>
          <string>GetPswListAsync</string>
        </value>
      </member>
      <member>
        <name>IPSERVER</name>
        <value>
          <string>127.0.0.1</string>
        </value>
      </member>
      <member>
        <name>PORTSERVER</name>
        <value>
          <int>8092</int>
        </value>
      </member>
    </struct>
  </value>
</member>

```

```

    </struct>
  </value>
</param>
</params>
</methodCall>

```

Для чтения конфигурации из прибора используется метод ReadConfiguration.

Параметры метода ReadConfiguration:

Вызов содержит один параметр типа STRUCT(структура), в котором передаются следующие данные:

- **STRING GUID** – жетон безопасности;
- **STRING IPSERVER** – Ip адрес сервера, на который необходимо передать результат исполнения команды;
- **INT64 PORTSERVER** – порт сервера, на который необходимо передать результат исполнения команды;
- **STRING MethodNameForAnswer** – имя метода на вызывающей стороне для передачи результата исполнения команды;
- **INT ComPort** – номер Com-порта.
- **INT PKUAddress** – адрес пульта C2000.
- **INT DeviceAddress** – адрес прибора.

Пример вызова ReadConfiguration:

```

<?xml version="1.0" encoding="windows-1251"?>
<methodCall>
  <methodName>ReadConfiguration</methodName>
  <params>
    <param>
      <value>
        <struct>
          <member>
            <name>ComPort</name>
            <value>
              <int>5</int>
            </value>
          </member>
          <member>
            <name>PKUAddress</name>
            <value>
              <int>127</int>
            </value>
          </member>
          <member>
            <name>DeviceAddress</name>
            <value>
              <int>6</int>
            </value>
          </member>
          <member>
            <name>ReaderAddress</name>
            <value>
              <int>0</int>
            </value>
          </member>
          <member>
            <name>MethodNameForAnswer</name>
            <value>
              <string>ReadConfResult</string>
            </value>
          </member>
        </struct>
      </value>
    </param>
  </params>
</methodCall>

```

```

    </value>
  </member>
  <member>
    <name>IPSERVER</name>
    <value>
      <string>127.0.0.1</string>
    </value>
  </member>
  <member>
    <name>PORTSERVER</name>
    <value>
      <int>8092</int>
    </value>
  </member>
</struct>
</value>
</param>
</params>
</methodCall>

```

Ответ на вызов ReadConfiguration:

Ответ содержит один параметр типа STRUCT(структура), в котором передаются следующие данные:

- **BOOL ShowConfigurationProgress** – значение должно быть TRUE.
- **INT Progress** – состояние процессы считывания конфигурации (отображается в процентах).
- **BOOL EverythingIsComplete** – считывание конфигурации завершено. Значение должно быть TRUE.
- **BOOL Result** – в случае успешного окончания процесса — TRUE.

Пример ответа на вызов ReadConfiguration:

```

<?xml version="1.0" encoding="windows-1251"?>
<methodCall>
  <methodName>ReadConfResult</methodName>
  <params>
    <param>
      <value>
        <struct>
          <member>
            <name>Comment</name>
            <value>
              <base64></base64>
            </value>
          </member>
          <member>
            <name>Progress</name>
            <value>
              <int>3</int>
            </value>
          </member>
          <member>
            <name>ShowConfigurationProgress</name>
            <value>
              <boolean>1</boolean>
            </value>
          </member>
          <member>
            <name>CommandCount</name>
            <value>
              <int>305</int>
            </value>
          </member>
        </struct>
      </value>
    </param>
  </params>
</methodCall>

```

```

</value>
</member>
<member>
  <name>CommandIndex</name>
  <value>
    <int>10</int>
  </value>
</member>
<member>
  <name>IPSERVER</name>
  <value>
    <string>192.168.10.127</string>
  </value>
</member>
<member>
  <name>MethodName</name>
  <value>
    <string>Unknown method</string>
  </value>
</member>
<member>
  <name>Result</name>
  <value>
    <boolean>0</boolean>
  </value>
</member>
<member>
  <name>Query</name>
  <value>
    <struct>
      <member>
        <name>ComPort</name>
        <value>
          <int>5</int>
        </value>
      </member>
      <member>
        <name>PKUAddress</name>
        <value>
          <int>127</int>
        </value>
      </member>
      <member>
        <name>DeviceAddress</name>
        <value>
          <int>6</int>
        </value>
      </member>
      <member>
        <name>ReaderAddress</name>
        <value>
          <int>0</int>
        </value>
      </member>
      <member>
        <name>MethodNameForAnswer</name>
        <value>
          <string>ReadConfResult</string>
        </value>
      </member>
      <member>
        <name>IPSERVER</name>
        <value>
          <string>127.0.0.1</string>
        </value>
      </member>
      <member>
        <name>PORTSERVER</name>

```

```

        <value>
          <int>8092</int>
        </value>
      </member>
    </struct>
  </value>
</member>
<member>
  <name>ComPort</name>
  <value>
    <int>5</int>
  </value>
</member>
<member>
  <name>PKUAddress</name>
  <value>
    <int>127</int>
  </value>
</member>
<member>
  <name>DeviceAddress</name>
  <value>
    <int>6</int>
  </value>
</member>
<member>
  <name>ID</name>
  <value>
    <int>-1</int>
  </value>
</member>
</struct>
</value>
</param>
</params>
</methodCall>

```

Для считывания списка ключей из прибора используется метод ReadDeviceKeyList.

Параметры метода ReadDeviceKeyList:

Вызов содержит один параметр типа STRUCT(структура), в котором передаются следующие данные:

- **STRING GUID** – жетон безопасности;
- **STRING IPSERVER** – Ip адрес сервера, на который необходимо передать результат исполнения команды;
- **INT64 PORTSERVER** – порт сервера, на который необходимо передать результат исполнения команды;
- **STRING MethodNameForAnswer** – имя метода на вызывающей стороне для передачи результата исполнения команды;
- **INT ComPort** – номер Com-порта.
- **INT PKUAddress** – адрес пульта C2000.
- **INT DeviceAddress** – адрес прибора.

Пример вызова ReadDeviceKeyList:

```

<?xml version="1.0" encoding="windows-1251"?>
<methodCall>
  <methodName>ReadDeviceKeyList</methodName>
  <params>

```



```

<param>
  <value>
    <struct>
      <member>
        <name>ComPort</name>
        <value>
          <int>5</int>
        </value>
      </member>
      <member>
        <name>PKUAddress</name>
        <value>
          <int>127</int>
        </value>
      </member>
      <member>
        <name>DeviceAddress</name>
        <value>
          <int>6</int>
        </value>
      </member>
      <member>
        <name>ReaderAddress</name>
        <value>
          <int>0</int>
        </value>
      </member>
      <member>
        <name>MethodNameForAnswer</name>
        <value>
          <string>ReadDeviceKeyListResult</string>
        </value>
      </member>
      <member>
        <name>IPSERVER</name>
        <value>
          <string>127.0.0.1</string>
        </value>
      </member>
      <member>
        <name>PORTSERVER</name>
        <value>
          <int>8092</int>
        </value>
      </member>
    </struct>
  </value>
</param>
</params>
</methodCall>

```

Ответ на вызов ReadDeviceKeyList:

Ответ содержит один параметр типа STRUCT(структура), в котором передаются следующие данные:

- **BOOL ShowKeysProgress** – значение должно быть TRUE.
- **INT Progress** – состояние процессы считывания конфигурации (отображается в процентах).
- **INT KeysCount** – количество ключей в приборе.
- **BOOL EverythingIsComplete** – считывание ключей завершено. Значение должно быть TRUE.
- **BOOL Result** – в случае успешного окончания процесса — TRUE.

Пример ответа на вызов ReadDeviceKeyList:

```
<?xml version="1.0" encoding="windows-1251"?>
```

```
<methodCall>
```

```
<methodName>ReadDeviceKeyListResult</methodName>
```

```
<params>
```

```
<param>
```

```
<value>
```

```
<struct>
```

```
<member>
```

```
<name>Comment</name>
```

```
<value>
```

```
<base64></base64>
```

```
</value>
```

```
</member>
```

```
<member>
```

```
<name>Progress</name>
```

```
<value>
```

```
<int>100</int>
```

```
</value>
```

```
</member>
```

```
<member>
```

```
<name>ShowKeysProgress</name>
```

```
<value>
```

```
<boolean>1</boolean>
```

```
</value>
```

```
</member>
```

```
<member>
```

```
<name>KeysCount</name>
```

```
<value>
```

```
<int>1</int>
```

```
</value>
```

```
</member>
```

```
<member>
```

```
<name>CommandCount</name>
```

```
<value>
```

```
<int>2</int>
```

```
</value>
```

```
</member>
```

```
<member>
```

```
<name>CommandIndex</name>
```

```
<value>
```

```
<int>2</int>
```

```
</value>
```

```
</member>
```

```
<member>
```

```
<name>IPSERVER</name>
```

```
<value>
```

```
<string>192.168.10.127</string>
```

```
</value>
```

```
</member>
```

```
<member>
```

```
<name>MethodName</name>
```

```
<value>
```

```
<string>Unknown method</string>
```

```
</value>
```

```
</member>
```

```
<member>
```

```
<name>Result</name>
```

```
<value>
```

```
<boolean>0</boolean>
```

```
</value>
```

```
</member>
```

```
<member>
```

```
<name>Query</name>
```

```
<value>
```

```
<struct>
```

```

<member>
  <name>ComPort</name>
  <value>
    <int>5</int>
  </value>
</member>
<member>
  <name>PKUAddress</name>
  <value>
    <int>127</int>
  </value>
</member>
<member>
  <name>DeviceAddress</name>
  <value>
    <int>6</int>
  </value>
</member>
<member>
  <name>ReaderAddress</name>
  <value>
    <int>0</int>
  </value>
</member>
<member>
  <name>MethodNameForAnswer</name>
  <value>
    <string>ReadDeviceKeyListResult</string>
  </value>
</member>
<member>
  <name>IPSERVER</name>
  <value>
    <string>127.0.0.1</string>
  </value>
</member>
<member>
  <name>PORTSERVER</name>
  <value>
    <int>8092</int>
  </value>
</member>
</struct>
</value>
</member>
<member>
  <name>ComPort</name>
  <value>
    <int>5</int>
  </value>
</member>
<member>
  <name>PKUAddress</name>
  <value>
    <int>127</int>
  </value>
</member>
<member>
  <name>DeviceAddress</name>
  <value>
    <int>6</int>
  </value>
</member>
<member>
  <name>ID</name>
  <value>
    <int>-1</int>
  </value>

```

```

    </member>
  </struct>
</value>
</param>
</params>
</methodCall>

```

Для определения статуса конфигурации (считана или нет) используется метод `GetDeviceConfigurationStatus`.

Параметры метода `GetDeviceConfigurationStatus`:

Вызов содержит один параметр типа `STRUCT`(структура), в котором передаются следующие данные:

- **STRING GUID** – жетон безопасности;

В ответ на запрос модуль высылает стандартный ответ об исполнении, возвращая конфигурацию модуля в структуре данных **RESULTDATA**, параметры структуры указаны ниже:

- **ARRAY DeviceList** – список устройств, из которых считана конфигурация.
 - **INT ID** – ID устройства;
 - **INT ComPort** – номер Com-порта;
 - **INT PKUAddress** – адрес пульта C2000;
 - **INT DeviceAddress** – адрес прибора;
 - **BOOL DeviceConfigurationStatus** – должен быть TRUE;
 - **BOOL DeviceKeyListStatus** – должен быть TRUE;
 - **INT KeysCount** – количество ключей в приборе;

Пример ответа на вызов `GetDeviceConfigurationStatus`:

```

<value>
<struct>
  <member>
    <name>DeviceList</name>
    <value>
      <array>
        <data>
          <value>
            <struct>
              <member>
                <name>ID</name>
                <value>
                  <int>-1</int>
                </value>
              </member>
              <member>
                <name>ComPort</name>
                <value>
                  <int>5</int>
                </value>
              </member>
              <member>
                <name>PKUAddress</name>
                <value>
                  <int>127</int>
                </value>
              </member>
            </struct>
          </value>
        </data>
      </array>
    </value>
  </member>
</struct>
</value>

```

```

</member>
<member>
  <name>DeviceAddress</name>
  <value>
    <int>6</int>
  </value>
</member>
<member>
  <name>DeviceConfigurationStatus</name>
  <value>
    <boolean>1</boolean>
  </value>
</member>
<member>
  <name>DeviceKeyListStatus</name>
  <value>
    <boolean>1</boolean>
  </value>
</member>
<member>
  <name>KeysCount</name>
  <value>
    <int>1</int>
  </value>
</member>
</struct>
</value>
</data>
</array>
</value>
</member>
</struct>
</value>

```

Параметры метода ControlAccess:

Вызов содержит один параметр типа STRUCT(структура), в котором передаются следующие данные:

- **STRING GUID** – жетон безопасности;
- **STRING IPSERVER** – Ip адрес сервера, на который необходимо передать результат исполнения команды;
- **INT64 PORTSERVER** – порт сервера, на который необходимо передать результат исполнения команды;
- **STRING MethodNameForAnswer** – имя метода на вызывающей стороне для передачи результата исполнения команды;
- **INT ComPort** – номер Com-порта.
- **INT PKUAddress** – адрес пульта C2000.
- **INT DeviceAddress** – адрес прибора.
- **INT AggregateAddress** – номер реле.
- **INT Command** – команда управления.
 - 0 – предоставление доступа;
 - 1 – разрешение (восстановление) доступа;
 - 2 – разрешение входа;
 - 3 – разрешение выхода (по кнопке);
 - 4 – запрет доступа (входа и выхода);
 - 5 – запрет входа;

6 – запрет выхода (по кнопке);

7 – открытие доступа.

- **INT Violations** – доп. данные.
- **ARRAY AccessCode** – код доступа.

Пример вызова ControlAccess:

```
<?xml version="1.0" encoding="windows-1251"?>
<methodCall>
<methodName>ControlAccess</methodName>
<params>
<param>
<value>
<struct>
<member>
<name>ComPort</name>
<value>
<int>5</int>
</value>
</member>
<member>
<name>PKUAddress</name>
<value>
<int>127</int>
</value>
</member>
<member>
<name>DeviceAddress</name>
<value>
<int>6</int>
</value>
</member>
<member>
<name>AggregateAddress</name>
<value>
<int>1</int>
</value>
</member>
<member>
<name>Command</name>
<value>
<int>0</int>
</value>
</member>
<member>
<name>MethodNameForAnswer</name>
<value>
<string>Result</string>
</value>
</member>
<member>
<name>IPSERVER</name>
<value>
<string>127.0.0.1</string>
</value>
</member>
<member>
<name>PORTSERVER</name>
<value>
<int>8092</int>
</value>
</member>
</struct>
</value>
</param>
```

</params>
</methodCall>

Ответ на вызов ControlAccess:

Ответ содержит один параметр типа STRUCT(структура), в котором передаются следующие данные:

- **STRUCT Address** – адрес прибора.
 - ▲ **INT ADDRDEVICE** – адрес устройства.
 - ▲ **INT ADDRPULT** – адрес пульта С2000.
 - ▲ **INT ADDRPORT** – номер Com-порта.
- **STRING Answer** – текст ответа.

В случае успешного выполнения значение равно “Access control”, и “Command is not supported” если команда не поддерживается.
- **INT Code** – код ответа.

Если команда выполнена, значение соответствует номеру программы управления, если не выполнена, может содержать следующие значения:

 - 0 – причина не известна;
 - 1 – нет контакта;
 - 2 – не поддерживаемый тип обмена команды;
 - 3 – приняты не все необходимые данные;
 - 4 – недопустимое значение данных;
 - 5 – нет квитанции;
 - 6 – адресное устройство команду не поддерживает;
 - 7 – команда заблокирована;
 - 8 – неверный код доступа;
 - 9 – недостаточно полномочий.

Пример ответа на вызов ControlAccess:

```
<?xml version="1.0" encoding="windows-1251"?>
<methodCall>
  <methodName>Result</methodName>
  <params>
    <param>
      <value>
        <struct>
          <member>
            <name>ADDRESS</name>
            <value>
              <struct>
                <member>
                  <name>ADDRDEVICE</name>
                  <value>
                    <int>6</int>
                  </value>
                </member>
                <member>
                  <name>ADDRPULT</name>
                  <value>
                    <int>127</int>
                  </value>
                </member>
                <member>
                  <name>ADDRPORT</name>
                  <value>
```

```

        <int>5</int>
    </value>
</member>
</struct>
</value>
</member>
<member>
    <name>Answer</name>
    <value>
        <string>Access control</string>
    </value>
</member>
<member>
    <name>Code</name>
    <value>
        <int>1</int>
    </value>
</member>
<member>
    <name>IPSERVER</name>
    <value>
        <string>192.168.10.127</string>
    </value>
</member>
<member>
    <name>MethodName</name>
    <value>
        <string>ControlAccess</string>
    </value>
</member>
<member>
    <name>Result</name>
    <value>
        <boolean>1</boolean>
    </value>
</member>
<member>
    <name>MessageType</name>
    <value>
        <int>2</int>
    </value>
</member>
<member>
    <name>Query</name>
    <value>
        <struct>
            <member>
                <name>ComPort</name>
                <value>
                    <int>5</int>
                </value>
            </member>
            <member>
                <name>PKUAddress</name>
                <value>
                    <int>127</int>
                </value>
            </member>
            <member>
                <name>DeviceAddress</name>
                <value>
                    <int>6</int>
                </value>
            </member>
            <member>
                <name>AggregateAddress</name>
                <value>
                    <int>1</int>
                </value>
            </member>
        </struct>
    </value>
</member>

```



```

        </value>
    </member>
    <member>
        <name>Command</name>
        <value>
            <int>0</int>
        </value>
    </member>
    <member>
        <name>MethodNameForAnswer</name>
        <value>
            <string>Result</string>
        </value>
    </member>
    <member>
        <name>IPSERVER</name>
        <value>
            <string>127.0.0.1</string>
        </value>
    </member>
    <member>
        <name>PORTSERVER</name>
        <value>
            <int>8092</int>
        </value>
    </member>
    <member>
        <name>METHODNAME</name>
        <value>
            <string>ControlAccess</string>
        </value>
    </member>
</struct>
</value>
</member>
</struct>
</value>
</param>
</params>
</methodCall>

```

Для считывания ключа со считывателя предназначен метод ReadKeyCodeFromReader. Перед вызовом данного метода необходимо пульт перевести в режим программирования.

Параметры метода ReadKeyCodeFromReader:

Вызов содержит один параметр типа STRUCT(структура), в котором передаются следующие данные:

- **STRING GUID** – жетон безопасности;
- **STRING IPSERVER** – IP адрес сервера, на который необходимо передать результат исполнения команды;
- **INT64 PORTSERVER** – порт сервера, на который необходимо передать результат исполнения команды;
- **STRING MethodNameForAnswer** – имя метода на вызывающей стороне для передачи результата исполнения команды;
- **INT ComPort** – номер Com-порта.
- **INT PKUAddress** – адрес пульта C2000.
- **INT DeviceAddress** – адрес прибора.
- **INT AggregateAddress** – адрес реле.

Пример вызова ReadKeyCodeFromReader:

```
<?xml version="1.0" encoding="windows-1251"?>
```

```
<methodCall>
```

```
<methodName>ReadKeyCodeFromReader</methodName>
```

```
<params>
```

```
<param>
```

```
<value>
```

```
<struct>
```

```
<member>
```

```
<name>MethodNameForAnswer</name>
```

```
<value>
```

```
<string>ResultMethod</string>
```

```
</value>
```

```
</member>
```

```
<member>
```

```
<name>IPSERVER</name>
```

```
<value>
```

```
<string>192.168.42.123</string>
```

```
</value>
```

```
</member>
```

```
<member>
```

```
<name>PORTSERVER</name>
```

```
<value>
```

```
<int>8081</int>
```

```
</value>
```

```
</member>
```

```
<member>
```

```
<name>ComPort</name>
```

```
<value>
```

```
<int>5</int>
```

```
</value>
```

```
</member>
```

```
<member>
```

```
<name>PKUAddress</name>
```

```
<value>
```

```
<int>0</int>
```

```
</value>
```

```
</member>
```

```
<member>
```

```
<name>DeviceAddress</name>
```

```
<value>
```

```
<int>4</int>
```

```
</value>
```

```
</member>
```

```
<member>
```

```
<name>DeviceType</name>
```

```
<value>
```

```
<int>4</int>
```

```
</value>
```

```
</member>
```

```
<member>
```

```
<name>AggregateAddress</name>
```

```
<value>
```

```
<int>1</int>
```

```
</value>
```

```
</member>
```

```
</struct>
```

```
</value>
```

```
</param>
```

```
</params>
```

```
</methodCall>
```

```
<methodCall>
```

```
<methodName>ResultMethod</methodName>
```

```
<params>
```

```
<param>
```

```
<value>
```

```

<struct>
  <member>
    <name>Result</name>
    <value>
      <string>FE00000F3BF3C101</string>
    </value>
  </member>
  <member>
    <name>IPSERVER</name>
    <value>
      <string>192.168.42.123</string>
    </value>
  </member>
  <member>
    <name>MethodName</name>
    <value>
      <string>ReadKeyCodeFromReader</string>
    </value>
  </member>
  <member>
    <name>Result</name>
    <value>
      <boolean>1</boolean>
    </value>
  </member>
  <member>
    <name>MessageType</name>
    <value>
      <int>2</int>
    </value>
  </member>
  <member>
    <name>ComPort</name>
    <value>
      <int>5</int>
    </value>
  </member>
  <member>
    <name>PKUAddress</name>
    <value>
      <int>0</int>
    </value>
  </member>
  <member>
    <name>DeviceAddress</name>
    <value>
      <int>4</int>
    </value>
  </member>
  <member>
    <name>ID</name>
    <value>
      <int>-1</int>
    </value>
  </member>
</struct>
</value>
</param>
</params>
</methodCall>

```

Ответ на вызов ReadKeyCodeFromReader:

Ответ содержит один параметр типа STRUCT(структура), в котором передаются следующие данные:

- **STRING RESULT** – считанный код ключа;
- **STRING IPSEVER** – Ip адрес сервера, который прислал ответ;
- **INT ComPort** – номер Com-порта;
- **INT PKUAddress** – адрес пульта C2000;
- **INT DeviceAddress** – адрес прибора, с которого считан ключ;
- **INT ID** – идентификатор считывателя, с которого считан ключ.

Пример ответа на вызов ReadKeyCodeFromReader:

```
<?xml version="1.0" encoding="windows-1251"?>
<methodCall>
  <methodName>ResultMethod</methodName>
  <params>
    <param>
      <value>
        <struct>
          <member>
            <name>Result</name>
            <value>
              <string>FE0000F3BF3C101</string>
            </value>
          </member>
          <member>
            <name>IPSERVER</name>
            <value>
              <string>192.168.11.125</string>
            </value>
          </member>
          <member>
            <name>MethodName</name>
            <value>
              <string>ReadKeyCodeFromReader</string>
            </value>
          </member>
          <member>
            <name>Result</name>
            <value>
              <boolean>1</boolean>
            </value>
          </member>
          <member>
            <name>MessageType</name>
            <value>
              <int>2</int>
            </value>
          </member>
          <member>
            <name>ComPort</name>
            <value>
              <int>5</int>
            </value>
          </member>
          <member>
            <name>PKUAddress</name>
            <value>
              <int>0</int>
            </value>
          </member>
        </struct>
      </value>
    </param>
  </params>
</methodCall>
```

```

<name>DeviceAddress</name>
<value>
  <int>4</int>
</value>
</member>
<member>
  <name>ID</name>
  <value>
    <int>-1</int>
  </value>
</member>
</struct>
</value>
</param>
</params>
</methodCall>

```

Для записи конфигурации в пульт C2000 используется метод WriteConfToPKU.
Перед вызовом данного метода необходимо пульт перевести в режим программирования.

Параметры метода WriteConfToPKU:

Вызов содержит один параметр типа STRUCT(структура), в котором передаются следующие данные:

- **STRING GUID** – жетон безопасности;
- **STRING IPSEVER** – Ip адрес сервера, на который необходимо передать результат исполнения команды;
- **INT64 PORTSERVER** – порт сервера, на который необходимо передать результат исполнения команды;
- **STRING MethodNameForAnswer** – имя метода на вызывающей стороне для передачи результата исполнения команды;
- **STRUCT DeviceAddress** – адрес пульта C2000 для экспорта конфигурации.
 - ⤴ **INT ADDRPUIT** – адрес пульта C2000;
 - ⤴ **INT ADDRPORT** – номер Com-порта;
- **STRING AuthorityPassword** – пароль хоз.органа;
- **STRING ProgrammerPassword** – пароль настройщика;
- **ARRAY BroadcastObjectList** – список приборов;
 - ⤴ **INT DeviceAddress** – адрес устройства;
 - ⤴ **INT BroadcastConfig** – транслируемые события;
 - бит 0 – пожары;
 - бит 1 – тревоги;
 - бит 2 – неисправности;
 - бит 3 – разделы;
 - бит 4 – шлейфы;
 - бит 6 – служебные;
 - бит 7 – технологические;
 - бит 8 – доступ;
 - бит 9 – реле;
 - ⤴ **ARRAY BroadcastObjectList** – список транслируемых разделов. Элемент списка является структурой данных (STRUCT).

- ✧ **INT ID** – идентификатор раздела/группы разделов;
- ✧ **INT ItemType** – тип элемента (2 – раздел; 17 – группа разделов);

Пример вызова WriteConfToPKU:

```
<?xml version="1.0" encoding="windows-1251"?>
<methodCall>
  <methodName>WriteConfToPKU</methodName>
  <params>
    <param>
      <value>
        <struct>
          <member>
            <name>GUID</name>
            <value>
              <string></string>
            </value>
          </member>
          <member>
            <name>DeviceAddress</name>
            <value>
              <struct>
                <member>
                  <name>ComPort</name>
                  <value>
                    <int>5</int>
                  </value>
                </member>
                <member>
                  <name>PKUAddress</name>
                  <value>
                    <int>127</int>
                  </value>
                </member>
                <member>
                  <name>DeviceAddress</name>
                  <value>
                    <int>0</int>
                  </value>
                </member>
                <member>
                  <name>AggregateAddress</name>
                  <value>
                    <int>0</int>
                  </value>
                </member>
              </struct>
            </value>
          </member>
          <member>
            <name>MethodNameForAnswer</name>
            <value>
              <string>OnRsEvent</string>
            </value>
          </member>
          <member>
            <name>IPSERVER</name>
            <value>
              <string>127.0.0.1</string>
            </value>
          </member>
          <member>
            <name>PORTSERVER</name>
            <value>
              <int>8092</int>
            </value>
          </member>
        </struct>
      </value>
    </param>
  </params>
</methodCall>
```

```

<member>
  <name>ProgrammerPassword</name>
  <value>
    <string>123</string>
  </value>
</member>
<member>
  <name>AuthorityPassword</name>
  <value>
    <string>4321</string>
  </value>
</member>
<member>
  <name>BroadcastObjectList</name>
  <value>
    <array>
      <data>
        <value>
          <struct>
            <member>
              <name>DeviceAddress</name>
              <value>
                <int>0</int>
              </value>
            </member>
            <member>
              <name>BroadcastConfig</name>
              <value>
                <int>0</int>
              </value>
            </member>
          </struct>
        </value>
      </data>
    </array>
  </value>
</member>
</struct>
</value>
</param>
</params>
</methodCall>

```

Ответ на вызов WriteConfToPKU:

Ответ содержит один параметр типа STRUCT(структура), в котором передаются следующие данные:

- **INT Progress** – состояние процессы считывания конфигурации (отображается в процентах).
- **BOOL EverythingIsComplete** – считывание конфигурации завершено. Значение должно быть TRUE.
- **BOOL Result** – в случае успешного окончания процесса — TRUE.

Пример ответа на вызов WriteConfToPKU:

```

<?xml version="1.0" encoding="windows-1251"?>
<methodCall>
  <methodName>OnRsEvent</methodName>
  <params>
    <param>
      <value>
        <struct>

```

```

<member>
  <name>Comment</name>
  <value>
    <base64></base64>
  </value>
</member>
<member>
  <name>Progress</name>
  <value>
    <int>0</int>
  </value>
</member>
<member>
  <name>CommandCount</name>
  <value>
    <int>3328</int>
  </value>
</member>
<member>
  <name>CommandIndex</name>
  <value>
    <int>6</int>
  </value>
</member>
<member>
  <name>IPSERVER</name>
  <value>
    <string>192.168.11.125</string>
  </value>
</member>
<member>
  <name>MethodName</name>
  <value>
    <string>Uknown method</string>
  </value>
</member>
<member>
  <name>Result</name>
  <value>
    <boolean>0</boolean>
  </value>
</member>
<member>
  <name>ComPort</name>
  <value>
    <int>5</int>
  </value>
</member>
<member>
  <name>PKUAddress</name>
  <value>
    <int>127</int>
  </value>
</member>
<member>
  <name>DeviceAddress</name>
  <value>
    <int>0</int>
  </value>
</member>
<member>
  <name>ID</name>
  <value>
    <int>1</int>
  </value>
</member>
</struct>
</value>

```



```
</param>
</params>
</methodCall>
```

Для управления взятием/снятием шлейфа используется метод ArmDisarmShleif.

Параметры метода ArmDisarmShleif:

Вызов содержит один параметр типа STRUCT(структура), в котором передаются следующие данные:

- **STRING GUID** – жетон безопасности;
- **STRING IPSEVER** – Ip адрес сервера, на который необходимо передать результат исполнения команды;
- **INT64 PORTSERVER** – порт сервера, на который необходимо передать результат исполнения команды;
- **STRING MethodNameForAnswer** – имя метода на вызывающей стороне для передачи результата исполнения команды;
- **STRING MethodNameForAnswer** – имя метода на вызывающей стороне для возвращения результата работы команды;
- **INT ComPort** – номер Com-порта.
- **INT PKUAddress** – адрес пульта C2000.
- **INT DeviceAddress** – адрес прибора.
- **INT AggregateAddress** – номер реле.
- **INT Command** – вид взятия/снятия.
 - 0 – снятие ШС;
 - 1 – групповое снятие;
 - 2 – взятие ШС;
 - 3 – групповое взятие;
 - 4 – сброс тревог;
 - 5 – отключение автоматики;
 - 6 – включение автоматики;
 - 7 – сброс пуска АСПТ;
 - 8 – дистанционный пуск.

Пример вызова ArmDisarmShleif:

```
<?xml version="1.0"?>
<methodCall>
  <methodName>ArmDisarmShleif</methodName>
  <params>
    <param>
      <value>
        <struct>
          <member>
            <name>ComPort</name>
            <value>
              <i4>5</i4>
            </value>
          </member>
          <member>
            <name>PKUAddress</name>
            <value>
              <i4>127</i4>
            </value>
          </member>
        </struct>
      </value>
    </param>
  </params>
</methodCall>
```

```

</value>
</member>
<member>
  <name>DeviceAddress</name>
  <value>
    <i4>2</i4>
  </value>
</member>
<member>
  <name>AggregateAddress</name>
  <value>
    <i4>1</i4>
  </value>
</member>
<member>
  <name>Command</name>
  <value>
    <i4>2</i4>
  </value>
</member>
<member>
  <name>MethodNameForAnswer</name>
  <value>
    <string>ControlRelayResult</string>
  </value>
</member>
<member>
  <name>IPSERVER</name>
  <value>
    <string>127.0.0.1</string>
  </value>
</member>
<member>
  <name>PORTSERVER</name>
  <value>
    <i4>8084</i4>
  </value>
</member>
<member>
  <name>ENDPOINT</name>
  <value>
    <string>/test.rem</string>
  </value>
</member>
</struct>
</value>
</param>
</params>
</methodCall>

```

Ответ на вызов ArmDisarmShleif:

Ответ содержит один параметр типа STRUCT(структура), в котором передаются следующие данные:

- **STRUCT Address** – адрес прибора.
 - ✦ **INT ADDRDEVICE** – адрес устройства.
 - ✦ **INT ADDRPULT** – адрес пульта С2000.
 - ✦ **INT ADDRPORT** – номер Com-порта.
- **STRING Answer** – текст ответа.

В случае успешного выполнения значение равно “Arm/Disarm”, и “Command is not supported” если команда не поддерживается.
- **INT Code** – код ответа.

Если команда выполнена, значение соответствует номеру программы управления, если не выполнена, может содержать следующие значения:

- 0 – причина не известна;
- 1 – нет контакта;
- 2 – не поддерживаемый тип обмена команды;
- 3 – приняты не все необходимые данные;
- 4 – недопустимое значение данных;
- 5 – нет квитанции;
- 6 – адресное устройство команду не поддерживает;
- 7 – команда заблокирована;
- 8 – неверный код доступа;
- 9 – недостаточно полномочий.

Пример ответа на вызов ArmDisarmShleif:

```
<?xml version="1.0" encoding="windows-1251"?>
<methodCall>
<methodName>ControlRelayResult</methodName>
<params>
<param>
<value>
<struct>
<member>
<name>ADDRESS</name>
<value>
<struct>
<member>
<name>ADDRDEVICE</name>
<value>
<int>2</int>
</value>
</member>
<member>
<name>ADDRPULT</name>
<value>
<int>127</int>
</value>
</member>
<member>
<name>ADDRPORT</name>
<value>
<int>5</int>
</value>
</member>
</struct>
</value>
</member>
<member>
<name>Answer</name>
<value>
<string>Arm/Disarm</string>
</value>
</member>
<member>
<name>Code</name>
<value>
<int>2</int>
</value>
</member>
<member>
<name>IPSERVER</name>
<value>
```

```

    <string>192.168.10.127</string>
  </value>
</member>
<member>
  <name>MethodName</name>
  <value>
    <string>ArmDisarmShleif</string>
  </value>
</member>
<member>
  <name>Result</name>
  <value>
    <boolean>1</boolean>
  </value>
</member>
<member>
  <name>MessageType</name>
  <value>
    <int>2</int>
  </value>
</member>
<member>
  <name>Query</name>
  <value>
    <struct>
      <member>
        <name>ComPort</name>
        <value>
          <int>5</int>
        </value>
      </member>
      <member>
        <name>PKUAddress</name>
        <value>
          <int>127</int>
        </value>
      </member>
      <member>
        <name>DeviceAddress</name>
        <value>
          <int>2</int>
        </value>
      </member>
      <member>
        <name>AggregateAddress</name>
        <value>
          <int>1</int>
        </value>
      </member>
      <member>
        <name>Command</name>
        <value>
          <int>2</int>
        </value>
      </member>
      <member>
        <name>MethodNameForAnswer</name>
        <value>
          <string>ControlRelayResult</string>
        </value>
      </member>
      <member>
        <name>IPSERVER</name>
        <value>
          <string>127.0.0.1</string>
        </value>
      </member>
    </struct>
  </value>
</member>

```

```

    <name>PORTSERVER</name>
    <value>
      <int>8084</int>
    </value>
  </member>
  <member>
    <name>ENDPOINT</name>
    <value>
      <string>/test.rem</string>
    </value>
  </member>
  <member>
    <name>METHODNAME</name>
    <value>
      <string>ArmDisarmShleif</string>
    </value>
  </member>
</struct>
</value>
</member>
</struct>
</value>
</param>
</params>
</methodCall>

```

Для управления выходами используется метод ControlRelay.

Параметры метода ControlRelay:

Вызов содержит один параметр типа STRUCT(структура), в котором передаются следующие данные:

- **STRING GUID** – жетон безопасности;
- **STRING IPSERVER** – Ip адрес сервера, на который необходимо передать результат исполнения команды;
- **INT64 PORTSERVER** – порт сервера, на который необходимо передать результат исполнения команды;
- **STRING MethodNameForAnswer** – имя метода на вызывающей стороне для передачи результата исполнения команды;
- **STRING MethodNameForAnswer** – имя метода на вызывающей стороне для возвращения результата работы команды;
- **INT ComPort** – номер Com-порта.
- **INT PKUAddress** – адрес пульта C2000.
- **INT DeviceAddress** – адрес прибора.
- **INT AggregateAddress** – номер реоле.
- **INT Command** – команда управления.
 - 0 – возврат в исходное состояние;
 - 1 – включить;
 - 2 – выключить;
 - 3 – включить на время;
 - 4 – выключить на время;
 - 5 – мигать из состояния выключено;
 - 6 – мигать из состояния включено;

- 7 – мигать из состояния выключено на время;
- 8 – мигать из состояния включено на время;
- 9 – лампа;
- 10 – ПЦН;
- 11 – пуск АСПТ;
- 12 – пуск-стоп речевого оповещения.

- **INT MaskNumber** – номер маски мигания (значения от 1 до 64).
- **INT ControlTime** – время управления (с).
- **INT WaitTime** – задержка управления (с).

Пример вызова ControlRelay:

```
<?xml version="1.0"?>
<methodCall>
  <methodName>ControlRelay</methodName>
  <params>
    <param>
      <value>
        <struct>
          <member>
            <name>ComPort</name>
            <value>
              <i4>5</i4>
            </value>
          </member>
          <member>
            <name>PKUAddress</name>
            <value>
              <i4>127</i4>
            </value>
          </member>
          <member>
            <name>DeviceAddress</name>
            <value>
              <i4>1</i4>
            </value>
          </member>
          <member>
            <name>AggregateAddress</name>
            <value>
              <i4>1</i4>
            </value>
          </member>
          <member>
            <name>Command</name>
            <value>
              <i4>1</i4>
            </value>
          </member>
          <member>
            <name>MethodNameForAnswer</name>
            <value>
              <string>ControlRelayResult</string>
            </value>
          </member>
          <member>
            <name>IPSERVER</name>
            <value>
              <string>127.0.0.1</string>
            </value>
          </member>
          <member>
            <name>PORTSERVER</name>
```

```

    <value>
      <i4>8084</i4>
    </value>
  </member>
</member>
<name>ENDPOINT</name>
<value>
  <string>/test.rem</string>
</value>
</member>
</struct>
</value>
</param>
</params>
</methodCall>

```

Ответ на вызов ControlRelay:

Ответ содержит один параметр типа STRUCT(структура), в котором передаются следующие данные:

- **STRUCT Address** – адрес прибора.
 - ⤴ **INT ADDRDEVICE** – адрес устройства.
 - ⤴ **INT ADDRPULT** – адрес пульта C2000.
 - ⤴ **INT ADDRPORT** – номер Com-порта.
- **STRING Answer** – текст ответа.
 В случае успешного выполнения значение равно “Relay control”, и “Command is not supported” если команда не поддерживается.
- **INT Code** – код ответа.
 Если команда выполнена, значение соответствует номеру программы управления, если не выполнена, может содержать следующие значения:
 - 0 – причина не известна;
 - 1 – нет контакта;
 - 2 – не поддерживаемый тип обмена команды;
 - 3 – приняты не все необходимые данные;
 - 4 – недопустимое значение данных;
 - 5 – нет квитанции;
 - 6 – адресное устройство команду не поддерживает;
 - 7 – команда заблокирована;
 - 8 – неверный код доступа;
 - 9 – недостаточно полномочий.

Пример ответа на вызов ControlRelay:

```

<?xml version="1.0" encoding="windows-1251"?>
<methodCall>
  <methodName>ControlRelayResult</methodName>
  <params>
    <param>
      <value>
        <struct>
          <member>
            <name>ADDRESS</name>
            <value>
              <struct>
                <member>
                  <name>ADDRDEVICE</name>
                  <value>

```

```

        <int>1</int>
    </value>
</member>
<member>
    <name>ADDRPULT</name>
    <value>
        <int>127</int>
    </value>
</member>
<member>
    <name>ADDRPORT</name>
    <value>
        <int>5</int>
    </value>
</member>
</struct>
</value>
</member>
<member>
    <name>Answer</name>
    <value>
        <string>Relay control</string>
    </value>
</member>
<member>
    <name>Code</name>
    <value>
        <int>1</int>
    </value>
</member>
<member>
    <name>IPSERVER</name>
    <value>
        <string>192.168.10.127</string>
    </value>
</member>
<member>
    <name>MethodName</name>
    <value>
        <string>ControlRelay</string>
    </value>
</member>
<member>
    <name>Result</name>
    <value>
        <boolean>1</boolean>
    </value>
</member>
<member>
    <name>MessageType</name>
    <value>
        <int>2</int>
    </value>
</member>
<member>
    <name>Query</name>
    <value>
        <struct>
            <member>
                <name>ComPort</name>
                <value>
                    <int>5</int>
                </value>
            </member>
            <member>
                <name>PKUAddress</name>
                <value>
                    <int>127</int>

```



```

        </value>
    </member>
    <member>
        <name>DeviceAddress</name>
        <value>
            <int>1</int>
        </value>
    </member>
    <member>
        <name>AggregateAddress</name>
        <value>
            <int>1</int>
        </value>
    </member>
    <member>
        <name>Command</name>
        <value>
            <int>1</int>
        </value>
    </member>
    <member>
        <name>MethodNameForAnswer</name>
        <value>
            <string>ControlRelayResult</string>
        </value>
    </member>
    <member>
        <name>IPSERVER</name>
        <value>
            <string>127.0.0.1</string>
        </value>
    </member>
    <member>
        <name>PORTSERVER</name>
        <value>
            <int>8084</int>
        </value>
    </member>
    <member>
        <name>ENDPOINT</name>
        <value>
            <string>/test.rem</string>
        </value>
    </member>
    <member>
        <name>METHODNAME</name>
        <value>
            <string>ControlRelay</string>
        </value>
    </member>
</struct>
</value>
</member>
</struct>
</value>
</param>
</params>
</methodCall>

```

Для удаления ключей из прибора используется метод DeleteKeys.

Параметры метода DeleteKeys:

Вызов содержит один параметр типа STRUCT(структура), в котором передаются следующие

данные:

- **STRING GUID** – жетон безопасности;
- **STRING IPSERVER** – Ip адрес сервера, на который необходимо передать результат исполнения команды;
- **INT64 PORTSERVER** – порт сервера, на который необходимо передать результат исполнения команды;
- **STRING MethodNameForAnswer** – имя метода на вызывающей стороне для передачи результата исполнения команды;
- **INT ComPort** – номер Com-порта.
- **INT PKUAddress** – адрес пульта C2000.
- **INT DeviceAddress** – адрес прибора.

Пример вызова DeleteKeys:

```
<?xml version="1.0" encoding="windows-1251"?>
<methodCall>
  <methodName>DeleteKeys</methodName>
  <params>
    <param>
      <value>
        <struct>
          <member>
            <name>ComPort</name>
            <value>
              <int>5</int>
            </value>
          </member>
          <member>
            <name>PKUAddress</name>
            <value>
              <int>127</int>
            </value>
          </member>
          <member>
            <name>DeviceAddress</name>
            <value>
              <int>6</int>
            </value>
          </member>
          <member>
            <name>ReaderAddress</name>
            <value>
              <int>0</int>
            </value>
          </member>
          <member>
            <name>MethodNameForAnswer</name>
            <value>
              <string>ReadDeviceKeyListResult</string>
            </value>
          </member>
          <member>
            <name>IPSERVER</name>
            <value>
              <string>192.168.10.127</string>
            </value>
          </member>
          <member>
            <name>PORTSERVER</name>
            <value>
              <int>8092</int>
            </value>
          </member>
        </struct>
      </value>
    </param>
  </params>
</methodCall>
```

```

    </value>
  </member>
</struct>
</value>
</param>
</params>
</methodCall>

```

Ответ на вызов DeleteKeys:

Ответ содержит один параметр типа STRUCT(структура), в котором передаются следующие данные:

- **INT Progress** – состояние процессы считывания конфигурации (отображается в процентах).

Пример вызова DeleteKeys:

```

<?xml version="1.0" encoding="windows-1251"?>
<methodCall>
  <methodName>ReadDeviceKeyListResult</methodName>
  <params>
    <param>
      <value>
        <struct>
          <member>
            <name>Comment</name>
            <value>
              <base64></base64>
            </value>
          </member>
          <member>
            <name>Progress</name>
            <value>
              <int>100</int>
            </value>
          </member>
          <member>
            <name>CommandCount</name>
            <value>
              <int>2</int>
            </value>
          </member>
          <member>
            <name>CommandIndex</name>
            <value>
              <int>2</int>
            </value>
          </member>
          <member>
            <name>IPSERVER</name>
            <value>
              <string>192.168.10.127</string>
            </value>
          </member>
          <member>
            <name>MethodName</name>
            <value>
              <string>Uknown method</string>
            </value>
          </member>
          <member>
            <name>Result</name>
            <value>
              <boolean>0</boolean>
            </value>
          </member>
        </struct>
      </value>
    </param>
  </params>
</methodCall>

```

```

</member>
<member>
  <name>Query</name>
  <value>
    <struct>
      <member>
        <name>ComPort</name>
        <value>
          <int>5</int>
        </value>
      </member>
      <member>
        <name>PKUAddress</name>
        <value>
          <int>127</int>
        </value>
      </member>
      <member>
        <name>DeviceAddress</name>
        <value>
          <int>6</int>
        </value>
      </member>
      <member>
        <name>ReaderAddress</name>
        <value>
          <int>0</int>
        </value>
      </member>
      <member>
        <name>MethodNameForAnswer</name>
        <value>
          <string>ReadDeviceKeyListResult</string>
        </value>
      </member>
      <member>
        <name>IPSERVER</name>
        <value>
          <string>192.168.10.127</string>
        </value>
      </member>
      <member>
        <name>PORTSERVER</name>
        <value>
          <int>8092</int>
        </value>
      </member>
    </struct>
  </value>
</member>
<member>
  <name>ComPort</name>
  <value>
    <int>5</int>
  </value>
</member>
<member>
  <name>PKUAddress</name>
  <value>
    <int>127</int>
  </value>
</member>
<member>
  <name>DeviceAddress</name>
  <value>
    <int>6</int>
  </value>
</member>

```

```

<member>
  <name>ID</name>
  <value>
    <int>-1</int>
  </value>
</member>
</struct>
</value>
</param>
</params>
</methodCall>

```

Для редактирования таблиц в модуле управления используется метод RefreshTablesData.

Параметры метода RefreshTablesData:

Вызов содержит один параметр типа STRUCT(структура), в котором передаются следующие данные:

- **STRING GUID** – жетон безопасности;
- **STRING IPSERVER** – Ip адрес сервера, на который необходимо передать результат исполнения команды;
- **INT64 PORTSERVER** – порт сервера, на который необходимо передать результат исполнения команды;
- **STRING MethodNameForAnswer** – имя метода на вызывающей стороне для передачи результата исполнения команды;
- **ARRAY TableList** – список редактируемых таблиц. Элемент списка является структурой данных (STRUCT).
- **STRING TableName** – название таблицы. Совпадает с названием таблиц метода LoadConfData;
- **ARRAY DataList** – список строк изменяемой таблицы. Элемент списка является структурой данных (STRUCT), состоящей из полей таблицы.

Пример вызова RefreshTablesData:

```

<?xml version="1.0" encoding="windows-1251"?>
<methodCall>
  <methodName>RefreshTablesData</methodName>
  <params>
    <param>
      <value>
        <struct>
          <member>
            <name>TableList</name>
            <value>
              <array>
                <data>
                  <value>
                    <struct>
                      <member>
                        <name>DataList</name>
                        <value>
                          <array>
                            <data>
                              <value>
                                <struct>
                                  <member>
                                    <name>ID</name>
                                    <value>

```

```

    <int>1</int>
  </value>
</member>
<member>
  <name>CodeType</name>
  <value>
    <int>3</int>
  </value>
</member>
<member>
  <name>Config</name>
  <value>
    <int>128</int>
  </value>
</member>
<member>
  <name>Owner</name>
  <value>
    <int>1</int>
  </value>
</member>
<member>
  <name>GroupID</name>
  <value>
    <int>1</int>
  </value>
</member>
<member>
  <name>Start</name>
  <value>
    <string>04.09.2012
    8:52:25</string>
  </value>
</member>
<member>
  <name>Finish</name>
  <value>
    <string>04.09.2012
    8:52:25</string>
  </value>
</member>
<member>
  <name>Code</name>
  <value>
    <array>
      <data>
        <value>
          <int>1</int>
        </value>
        <value>
          <int>117</int>
        </value>
        <value>
          <int>234</int>
        </value>
        <value>
          <int>51</int>
        </value>
        <value>
          <int>15</int>
        </value>
        <value>
          <int>0</int>
        </value>
        <value>
          <int>0</int>
        </value>
      </data>
    </array>
  </value>
</member>

```

```

        <int>182</int>
      </value>
    </data>
  </array>
</value>
</member>
</struct>
</value>
</data>
</array>
</value>
</member>
<member>
  <name>Action</name>
  <value>
    <int>1</int>
  </value>
</member>
<member>
  <name>TableName</name>
  <value>
    <string>Passwords</string>
  </value>
</member>
</struct>
</value>
</data>
</array>
</value>
</member>
<member>
  <name>MethodNameForAnswer</name>
  <value>
    <string>ONRSEVENT</string>
  </value>
</member>
<member>
  <name>IPSERVER</name>
  <value>
    <string>127.0.0.1</string>
  </value>
</member>
<member>
  <name>PORTSERVER</name>
  <value>
    <int>8092</int>
  </value>
</member>
</struct>
</value>
</param>
</params>
</methodCall>

```

Конфигурирование СКУД и управления взятием/снятием ШС

Необходимо загрузить данные контроля доступа в модуль управления выполнив команду LoadConfData (LoadAccessData).

Затем считать конфигурацию, если она не была считана ранее. Это выполняется с помощью команд ReadConfiguration. В процессе чтения ядро будет высылать структуры со значением процентов считанной конфигурации либо кодов ключей (данные действия необходимо выполнить один раз, далее конфигурация будет подгружаться из кэша).

После того, как конфигурация и коды ключей считаны, ядру необходимо дать команду для получения состояния кодов ключей – GetPasswordListWithStatus. В результате ядро сравнит уровни доступа каждого ключа с конфигурацией приборов и вышлет в ответ список с ключами и их состояниями – правильно ли записан ключ либо нет.

После сравнения можно выполнять синхронизацию либо всего списка ключей либо одного отдельно взятого ключа с помощью методов SynchronizeOneKey, SynchronizeKeys. На основании уровня доступа ключ будет удален либо записан в каждый прибор, который входит в состав уровня доступа. После синхронизации ключа в ответ высылается состояние ключа.

Приложение А. Коды типов приборов, состояний и событий.

Таблица 1. Типы поддерживаемых приборов.

Код типа прибора	Наименование прибора
0	C2000
1	Сигнал 20
2	Сигнал 20П
4	C2000-4
7	C2000-К
8	C2000-ИТ
9	C2000-КДЛ
10	C2000-БИ
11	Сигнал-20 (серия 02)
13	C2000-КС
14	C2000-АСПТ
15	C2000-КПБ
16	C2000-2
20	Рупор
22	C2000-ПТ
23	C2000-ПЦН
24	УО-4С
25	Поток-3Н
26	Сигнал-20М
28	C2000-БИ исп. 01

30	Рупор исп. 01
31	C2000-Adem
33	РИП-12 RS
34	Сигнал-10
38	РИП-12-2А RS
39	РИП-24-2А RS
41	C2000-КДЛ-2И
43	C2000-PGE
44	C2000-БКИ
45	Поток-БКИ
46	Рупор-200

Таблица 2. Коды возможных состояний.

Объект	Состояние	Описание
Прибор	1	Норма сети 220
Прибор	2	Авария сети 220
Прибор	47	Норма ДПЛС
Прибор	149	Тревога взлома корпуса
Прибор	152	Норма корпуса
Прибор	189	Потеря связи по ветви ДПЛС1
Прибор	190	Потеря связи по ветви ДПЛС2
Прибор	191	Норма связи по ветви ДПЛС1
Прибор	198	Неисправность источника питания
Прибор	199	Норма источника питания
Прибор	201	Норма связи по ветви ДПЛС2
Прибор	202	Неисправность батареи
Прибор	215	КЗ ДПЛС
Прибор	222	Авария ДПЛС
Прибор	250	Нет контакта
Прибор	251	Норма контакта
-	-	-

Считыватель	0	Неопределенное
Считыватель	25	Доступ закрыт
Считыватель	30	Доступ восстановлен
Считыватель	219	Доступ открыт
Считыватель	250	Нет контакта
Считыватель	251	Норма контакта
-	-	-
Дверь	-1	Неопределенное
Дверь	25	Доступ закрыт
Дверь	26	Доступ отклонен
Дверь	27	Дверь взломана
Дверь	28	Доступ предоставлен
Дверь	29	Запрет доступа
Дверь	30	Доступ восстановлен
Дверь	31	Норма двери
Дверь	32	Проход
Дверь	33	Дверь заблокирована
Дверь	219	Свободный доступ открыт
Дверь	250	Нет контакта
Дверь	251	Норма контакта
-	-	-
Реле	0	Неопределенное
Реле	47	Норма ДПЛС
Реле	121	Обрыв цепи
Реле	122	Короткое замыкание цепи
Реле	123	Норма цепи
Реле	126	Отключение выхода
Реле	127	Подключение выхода
Реле	149	Тревога взлома корпуса
Реле	152	Норма корпуса
Реле	189	Потеря связи по ветви ДПЛС1
Реле	190	Потеря связи по ветви ДПЛС2

Реле	191	Норма связи по ветви ДПЛС1
Реле	201	Норма связи по ветви ДПЛС2
Реле	215	КЗ ДПЛС
Реле	222	Авария ДПЛС
Реле	250	Нет контакта
Реле	251	Норма контакта
Реле	401	Реле включено
Реле	402	Реле выключено
Реле	403..464	Реле мигает 3..64
-	-	-
Шлейф	0	Неопределенное
Шлейф	1	Норма сети 220
Шлейф	2	Авария сети
Шлейф	3	Тревога проникновения
Шлейф	17	Неудачное взятие
Шлейф	23	Задержка взятия
Шлейф	24	Взят
Шлейф	35	Норма технологического ШС
Шлейф	36	Нарушение технологического ШС
Шлейф	37	Пожар
Шлейф	38	Нарушение 2 технологического
Шлейф	39	Пожарное оборудование в норме
Шлейф	41	Неисправность пожарного оборудования
Шлейф	44	Внимание! Опасность пожара
Шлейф	45	Обрыв шлейфа
Шлейф	47	Норма ДПЛС
Шлейф	58	Тихая тревога
Шлейф	71	Понижение уровня
Шлейф	72	Уровень в норме
Шлейф	74	Повышение уровня
Шлейф	75	Аварийное повышение уровня
Шлейф	76	Повышение температуры

Шлейф	77	Аварийное понижение уровня
Шлейф	78	Температура в норме
Шлейф	82	Неисправность термометра
Шлейф	109	Снят
Шлейф	110	Сброс тревоги
Шлейф	117	Норма снятой зоны
Шлейф	118	Тревога входной зоны
Шлейф	119	Нарушение снятой зоны
Шлейф	130	Включение насоса
Шлейф	131	Выключение насоса
Шлейф	141	Задержка автоматического пуска ПТ
Шлейф	142	Автоматика выключена
Шлейф	143	Сброс пуска АСПТ
Шлейф	144	Тушение
Шлейф	145	Аварийный пуск
Шлейф	146	Пуск АСПТ
Шлейф	147	Блокировка пуска ПТ
Шлейф	148	Автоматика включена
Шлейф	149	Тревога взлома корпуса
Шлейф	152	Норма корпуса
Шлейф	158	Норма внутренней зоны
Шлейф	165	Ошибка параметров ШС
Шлейф	187	ШС отключен
Шлейф	188	ШС подключен
Шлейф	189	Потеря связи по ветви ДПЛС1
Шлейф	190	Потеря связи по ветви ДПЛС2
Шлейф	191	Норма связи по ветви ДПЛС1
Шлейф	192	Отключение выходного напряжения источника питания
Шлейф	193	Подключение выходного напряжения источника питания
Шлейф	193	Неудачный пуск ПТ
Шлейф	194	Перегрузка источника питания
Шлейф	196	Неисправность ЗУ

Шлейф	197	Норма ЗУ
Шлейф	198	Неисправность источника питания
Шлейф	199	Норма источника питания
Шлейф	200	Норма батареи
Шлейф	200	Норма батареи
Шлейф	201	Норма связи по ветви ДПЛС2
Шлейф	202	Неисправность батареи
Шлейф	205	Ошибка теста АКБ
Шлейф	206	Понижение температуры
Шлейф	211	Разряд батареи
Шлейф	212	Разряд резервной батареи
Шлейф	213	Норма резервной батареи
Шлейф	214	Короткое замыкание
Шлейф	215	Короткое замыкание ДПЛС
Шлейф	222	Авария ДПЛС
Шлейф	224	Некорректный ответ устройства в ДПЛС
Шлейф	225	Неустойчивый ответ устройства в ДПЛС
Шлейф	243	Идет взятие
Шлейф	244	Идет снятие
Шлейф	250	Нет контакта
Шлейф	251	Норма контакта

Таблица 3. Коды возможных событий.

Код события	Описание
1	Восстановление сети
2	Авария сети
3	Тревога проникновения
5	Реакция оператора
17	Неудачное взятие
19	Тест извещателя
20	Пожарное тестирование

21	Выключение пожарного тестирования
22	Восстановление контроля
23	Задержка взятия
24	Взятие зоны охраны
25	Доступ закрыт
26	Доступ отклонен
27	Дверь взломана
28	Доступ предоставлен
29	Запрет доступа
30	Восстановление доступ
31	Восстановление целостности двери
32	Проход
33	Дверь заблокирована
34	Идентификатор хозоргана
35	Восстановление технологического ШС
36	Нарушение технологического ШС
37	Тревога пожарного ШС
38	Нарушение 2 техн. ШС
39	Пожарное оборудование в норме
41	Неисправность пожарного оборудования
42	Нестандартное оборудование
44	Внимание! Опасность пожара
45	Обрыв шлейфа
46	Обрыв ДПЛС
47	Восстановление ДПЛС
58	Тихая тревога
67	Изменение даты
69	Журнал заполнен
70	Журнал переполнен
71	Понижение уровня
72	Уровень в норме
73	Изменение времени

74	Повышение уровня
75	Аварийное повышение уровня
76	Повышение температуры
77	Аварийное понижение уровня
78	Температура в норме
82	Неисправность термометра
83	Восстановление термометра
84	Локальное программирование
90	Неисправность телефонной линии
91	Восстановление телефонной линии
94	Нагрев калорифера
95	Угроза охлаждения
96	Угроза замерзания
97	Перегрев обратной воды
98	Загрязнение воздушного фильтра
99	Отказ вентилятора
100	Лето-день
101	Лето-ночь
102	Зима-день
103	Лето-ночь
109	Снятие ШС
110	Сброс тревоги
117	Восстановление снятой зоны
118	Тревога входной зоны
119	Нарушение снятой зоны
121	Обрыв цепи выхода (реле)
122	Короткое замыкание цепи выхода (реле)
123	Восстановление цепи выхода (реле)
126	Отключение выхода (реле)
127	Подключение выхода (реле)
128	Изменение состояния выхода (включение/выключение реле)
130	Включение насоса

131	Выключение насоса
135	Ошибка при автоматическом тестировании
136	Восстановление напряжения питания
137	Срабатывание цепи пуска
138	Отказ цепи пуска
139	Неудачный пуск ПТ
140	Ручной тест
141	Задержка автоматического пуска
142	Автоматика выключена
143	Сброс пуска АСПТ
144	Тушение
145	Аварийный пуск
146	Пуск АСПТ
147	Блокировка пуска
148	Автоматика включена
149	Тревога взлома
150	Пуск речевого оповещения
151	Сброс пуска речевого оповещения
152	Восстановление зоны контроля взлома
158	Восстановление внутренней зоны
165	Ошибка параметров ШС
172	Включение принтера
173	Выключение принтера
187	ШС отключен
188	ШС подключен
189	Потеря связи по ветви ДПЛС1
190	Потеря связи по ветви ДПЛС2
191	Восстановление связи по ветви ДПЛС1
192	Отключение выходного напряжения
193	Подключение выходного напряжения
194	Перегрузка источника питания
195	Перегрузка источника устранена

196	Неисправность ЗУ
197	Восстановление ЗУ
198	Неисправность источника питания
199	Восстановление источника питания
200	Восстановление батареи
201	Восстановление связи по ветви ДПЛС2
202	Неисправность батареи
203	Сброс сторожевого таймера
204	Требуется обслуживание
206	Понижение температуры
214	Короткое замыкание
215	Короткое замыкание ДПЛС
216	Сработка датчика
217	Отключение ветви интерфейса RS-485
218	Восстановление ветви интерфейса RS-485
219	Доступ открыт
220	Срабатывание СДУ
221	Отказ СДУ
222	Авария ДПЛС
223	Отметка наряда
224	Некорректный ответ устройства в ДПЛС
225	Неустойчивый ответ устройства в ДПЛС
238	Смена дежурства
241	Взятие раздела
242	Снятие раздела
243	Удаленный запрос на взятие
244	Удаленный запрос на снятие
245	Удаленный запрос доступа
246	Неверный пароль
247	Неверный раздел
248	Превышение полномочий
249	Программирование (произошло изменение параметров конфигурации)

250	Потерян контакт с устройством
251	Восстановлен контакт с прибором
252	Подмена прибора
253	Включение пульта С2000
254	Отметка даты
255	Отметка времени
265	Два пожара
270	Доступ предоставлен (по кнопке)
271	Проход (по кнопке)
272	Запрет доступа (по кнопке)
280	Взятие группы разделов
281	Снятие группы разделов
311	Включить
312	Выключить
313	Включить на время
314	Выключить на время
315	Мигать из состояния выключено
316	Мигать из состояния включено
317	Мигать из состояния выключено на время
318	Мигать из состояния включено на время
319	Лампа
320	ПЦН
321	АСПТ
322	Сирена
323	Пожарный ПЦН
324	Выход неисправности
325	Пожарная лампа
326	Старая тактика ПЦН
327	Включить на время перед взятием
328	Выключить на время перед взятием
329	Включить на время при взятии
330	Выключить на время при взятии

331	Включить на время при снятии
332	Выключить на время при снятии
333	Включить на время при невзятии
334	Выключить на время при невзятии
335	Включить на время при нарушении технологического ШС
336	Выключить на время при тушении технологического ШС
337	Включить при снятии
338	Выключить при снятии
339	Включить при взятии
340	Выключить при взятии
341	Включить при нарушении технологического ШС
342	Выключить при нарушении технологического ШС
343	АСПТ-1
344	АСПТ-А
345	АСПТ-А1
360	Запуск сценария управления
380	Передано сообщение
390	Запрос вкл. автоматики
391	Запрос выкл. автоматики
392	Запрос на пуск
393	Запрос на сброс пожаротушения
401	Включение реле
402	Выключение реле
403	Мигание реле
440	Изменение состояния