ЗАО «Связь инжиниринг М»

Набор тестов для автоматизированного тестирования текстового протокола обмена данными с УСПД УМ-RTU

Версия 1.0

2016

Содержание

[1. Состав набора 3](#_Toc468461644)

[2. Системные требования 3](#_Toc468461645)

[3. Быстрый старт 3](#_Toc468461646)

[3.1. Подготовка к запуску тестов 3](#_Toc468461647)

[3.2. Подготовительный этап 3](#_Toc468461648)

[3.3. Запуск тестов 4](#_Toc468461649)

[3.3.1. Запуск одиночного теста: 4](#_Toc468461650)

[3.3.2. Массовый запуск тестов: 4](#_Toc468461651)

[3.3.3. Запуск только проваленных тестов: 4](#_Toc468461652)

[4. Подробное описание настроек тестов 4](#_Toc468461653)

[4.1. Настройки связи с УСПД 4](#_Toc468461654)

[4.2. Пароли доступа к УСПД 4](#_Toc468461655)

[4.3. Параметры ВПО УСПД 4](#_Toc468461656)

[4.4. Аппаратные параметры УСПД 4](#_Toc468461657)

[4.5. Параметры оператора связи 5](#_Toc468461658)

[5. Подробное описание Набора 5](#_Toc468461659)

[5.1. Работа с COM-портом и УСПД 5](#_Toc468461660)

[5.1.1. Драйвер COM-порта (serialPort.py) 5](#_Toc468461661)

[5.1.2. Слой совместимости текстового протокола и драйвера (port.py) 6](#_Toc468461662)

[5.1.3. Модуль работы с УСПД (actions.py) 6](#_Toc468461663)

[5.1.4. Интерфейс записи логов обмена по COM-порту (logger.py) 9](#_Toc468461664)

[5.2. Интерпретация протокола обмена УМ-RTU 10](#_Toc468461665)

[5.2.1. Интерпретатор протокола(txtProto.py) 10](#_Toc468461666)

[5.2.2. Счетчик контрольной суммы (CRC16RTU.py) 11](#_Toc468461667)

[5.3. Настройки Набора 12](#_Toc468461668)

[5.3.1. Парсер настроек (setup.py) 12](#_Toc468461669)

[5.4. Описание шаблона теста 12](#_Toc468461670)

[5.4.1. Блок параметризации 12](#_Toc468461671)

[5.4.2. Класс Test\_proto 12](#_Toc468461672)

[Приложение 1. Описание служебного протокола Набора. 13](#_Toc468461673)

[Приложение 2. Статус покрытия команд протокола УМ-RTU тестами 14](#_Toc468461674)

# Состав набора

Автоматизированные тесты представляют собой скрипты, написанные на языке Python.

В состав набора тестов входят следующие компоненты:

1. Скрипты инициализации и завершения тестирования (000\_PREPARE.py, XXX\_RESET.py);
2. Скрипты тестов (Файлы имеют вид «XXX\_NNNNN\_type\_test.py», где XXX – порядковый номер теста, NNNNN – имя теста, type – тип теста, test – ссылка для фреймворка тестирования);
3. Файлы для работы на «низком» уровне (папка Drivers):
   1. serialPort.py – драйвер COM-порта;
   2. port.py – слой совместимости текстового протокола и драйвера COM-порта;
   3. logger.py – интерфейс записи логов обмена;
   4. actions.py – модуль работы с УСПД;
4. Интерпретатор протокола УСПД (папка Proto):
   1. txtProto.py – интерпретатор протокола;
   2. CRC16RTU.py – счетчик контрольной суммы;
5. Конфигуратор тестов:
   1. setup.py – парсер настроек;
   2. settings.xml – настройки тестов;

# Системные требования

Машина для запуска тестов должна удовлетворять следующим системны требованиям:

1. Python версии не ниже 3.5.
2. Фреймворк Pytest.
3. Фреймворк Pyserial.
4. COM-порт.

# Быстрый старт

## Подготовка к запуску тестов

Чтобы приступить к выполнению тестов необходимо произвести подготовку окружения:

1. Записать настройки системы и проверяемого УСПД в файл settings.xml.
2. Выполнить прошивку УСПД тестируемым ВПО с полной очисткой настроек УСПД.
3. Установить в УСПД SIM-карту.
4. Подключить УСПД к ПК с помощью COM-порта.
5. Убедиться, что COM-порт не занят другим ПО.

## Подготовительный этап

Перед непосредственным запуском тестов следует подготовить УСПД к работе, записав в него тестовые пароли:

1. Запустить CMD (Terminal) (далее считается, что работа ведется через c использованием командной строки и данный пункт будет пропущен).
2. Осуществить навигацию в папку с Набором.
3. Выполнить скрипт «000\_PREPARE.py» с помощью команды: **python 000\_PREPARE.py.**
4. Убедиться, что скрипт выполнен без ошибок и в логе (папка Log отсутствуют ошибки выполнения команд).

## Запуск тестов

Тесты можно запускать как по одиночке, так и все доступные в порядке очереди. Каждый тест может содержать в себе несколько итераций.

### Запуск одиночного теста:

1. Из папки с Набором выполнить команду: **py.test XXX\_NNNNN\_type\_test.py -s -v** .
2. Проверить папку Log на предмет ошибок в выполненном тесте.

### Массовый запуск тестов:

1. Из папки с Набором выполнить команду: **py.test -s -v** .
2. Проверить папку Log на предмет ошибок в выполненных тестах.

### Запуск только проваленных тестов:

1. Из папки с Набором выполнить команду: **py.test -s -v --lf** .
2. Проверить папку Log на предмет ошибок в выполненных тестах.

# Подробное описание настроек тестов

Настройки содержатся в файле settings.xml, который находится в папке Settings.

## Настройки связи с УСПД

* **port** – номер COM-порта для связи с УСПД;

## Пароли доступа к УСПД

Пароли доступа следует задавать отличающиеся друг от друга. Пароли доступа записываются в УСПД при выполнении скрипта подготовки к тестированию. Не следует выполнять запись паролей в УСПД вручную.

* **pwd1** – пароль уровня Пользователь;
* **pwd2** – пароль уровня Конфигуратор;
* **pwd3** – пароль уровня Завод;
* **pwde** – ошибочный пароль. В данный момент не используется;

## Параметры ВПО УСПД

Для успешного тестирования требуется указать параметры ВПО в УСПД.

* **swver** – версия ВПО;
* **hwver** – версия аппаратного обеспечения;
* **blver** – версия загрузчика;

## Аппаратные параметры УСПД

Для успешного тестирования требуется указать аппаратные параметры проверяемого УСПД.

* **dev** – тип УСПД (Допустимо: UM-31RTU, UM-40RTU.2);
* **modem\_name** – производитель модема;
* **modem\_model** – модель модема, например EHS5-E;
* **imei** – IMEI модема УСПД;
* **UMTS** – поддержка 3G (Допустимо: 1 – есть поддержка, 0 – нет поддержки);
* **cover\_di** – наличие дискретного датчика крышки УСПД (0 – нет, 1 - есть);
* **protective\_cover\_di** – наличие дискретного датчика защитной крышки УСПД (0 – нет, 1 - есть);
* **power\_di** – наличие дискретного датчика основного питания УСПД (0 – нет, 1 - есть);
* **failsafe\_power\_di** – наличие дискретного датчика резервного питания УСПД. В данный момент не используется;
* **di\_count** – количество дискретных датчиков УСПД общего назначения;
* **pl\_count** – количество линий питания УСПД;
* **ifaces** – интерфейсы УСПД (Номера интерфейсов по протоколу. Разделитель «;»). Не следует указывать интерфейс RS-232;
* **pls** – линии питания УСПД (Номера линий питания по протоколу. Разделитель «;»);
* **real\_time** – наличие часов реального времени (0 – нет, 1 - есть);
* **df1** – тип микросхемы памяти 1 (Например, «32Mbit»), пусто, если отсутствует;
* **df2** – тип микросхемы памяти 2 (Например, «32Mbit»), пусто, если отсутствует;
* **FRAM** – тип микросхемы FRAM, если известно (Если производиель не может быть определен – «UNKNOWN»);
* **HW\_INFO** – строка информации об исполнении устройства (Например, «1;B9;2;1F;7F;1F;7;31»)

## Параметры оператора связи

Для успешного тестирования требуется указать параметры ВПО в УСПД.

* **OPS** – оператор установленной SIM-карты;
* **SCID** – номер установленной SIM-карты;

# Подробное описание Набора

## Работа с COM-портом и УСПД

### Драйвер COM-порта (serialPort.py)

Драйвер порта представляет собой класс SerialPort, наследованный от класса Serialмодуля **pyserial**. Класс может использовать все методы и свойства родительского класса. Основными используемыми методами являются:

##### Встроенный метод **write(data):**

Метод осуществляет запись данных в формате байтовой строки в порт.

Входные параметры:

* **data** – байтовая строка (bytes);

Выходные параметры:

* метод ничего не возвращает;

##### Метод **read\_data(answer\_timeout)**:

Метод осуществляет чтение данных с порта и возвращает их в формате байтовой строки (bytes).

Входные параметры:

* **answer\_timeout** – время ожидания ответа. Целочисленное значение (integer) в секундах;

Выходные параметры:

* Байтовая строка считанных данных (bytes);

##### Метод **read\_data\_flush(answer\_timeout)**:

Метод осуществляет чтение данных с порта без их обработки и передачи куда-либо. Цель метода считать мусор.

Входные параметры:

* **answer\_timeout** – время ожидания ответа. Целочисленное значение (integer) в секундах;

### Слой совместимости текстового протокола и драйвера (port.py)

Представляет собой модуль port.py. Модуль содержит функции, переводящие строковые данные команд (string) в байтовые строки (bytes), добавляющие или удаляющие специальные символы протокола УСПД и отсылающие или принимающие данные, используя драйвер serialPort.py.

##### Функция **sendcommand(command)**

Функция осуществляет перевод строки (string) в байтовую строку, добавление специальных символов, отправку в порт.

Входные параметры:

* **command** – строка (string) команды для передачи в порт;

Выходные параметры:

* Функция ничего не возвращает;

##### Функция **readanswer(timeout)**

Функция осуществляет чтение данных из порта, отрезает специальные символы, переводит данные из байтовой строки (bytes) в строку (string).

Входные параметры:

* **timeout** – время ожидания ответа;

Выходные параметры:

* Строковые данные (string);

##### Функция **portclose()**

Функция закрывает порт. Функция не принимает и не возвращает параметров.

### Модуль работы с УСПД (actions.py)

Модуль представляет собой класс Action. Содержит методы работы с УСПД по протоколу УМ-RTU.

Свойства класса:

* **answer\_command** – ответная команда в строковом формате (string);
* **test\_name** – имя теста в строковом формате (string);
* **pwd** – пароль доступа к УСПД в строковом формате (string);
* **write\_command** – команда записи в строковом формате (string) или **None**;
* **read\_command** – команда чтения в строковом формате (string) или **None**;
* **timeout** – время ожидания ответа в секундах. Целочисленное значение (integer) или **None**;

##### Метод **write\_to\_device(\*params)**

Осуществляет отправку команды записи на УСПД, получает ответ от УСПД.

Входные параметры:

* Неограниченное количество строковых параметров (string) или **None**. Если **None** – первый параметр, то запись команды будет произведена без параметров. В остальных случаях, **None** игнорируется при записи.

Выходные параметры:

* Массив:
  + Строковый статус выполнения команды по внутреннему протоколу Набора;
  + Ответ УСПД (если есть), пропущенный через транслятор протокола УСПД;

##### Метод **write\_to\_device\_counter(\*params)**

Осуществляет отправку подготовку к записи счечтика и отправляет команду записи счетчика в таблицу счетчиков УСПД.

Входные параметры:

* Неограниченное количество строковых параметров (string) или **None**. Если **None** – первый параметр, то запись команды будет произведена без параметров. В остальных случаях, **None** игнорируется при записи.

Выходные параметры:

* Массив:
  + Строковый статус выполнения команды по внутреннему протоколу Набора;
  + Ответ УСПД (если есть), пропущенный через транслятор протокола УСПД;

##### Метод **read\_from\_device(\*params)**

Осуществляет отправку команды чтения на УСПД, получает ответ от УСПД. Успешность выполнения определяется по соответствию ответа УСПД ответной команде (answer\_command)

Входные параметры:

* Неограниченное количество строковых параметров (string) или **None** (Если **None** – первый параметр, то запись команды будет произведена без параметров. В остальных случаях, **None** игнорируется при записи);

Выходные параметры:

* Массив:
  + Строковый статус выполнения команды по внутреннему протоколу Набора;
  + Ответ УСПД (если есть), пропущенный через транслятор протокола УСПД;

##### Метод **read\_from\_device\_old(\*params)**

Осуществляет отправку команды чтения на УСПД, получает ответ от УСПД. Успешность выполнения определяется по соответствию ответа УСПД ответной команде (answer\_command). Осуществляет разбор строки, с разделителями в виде знаков «пробел», «.», «:».

Входные параметры:

* Неограниченное количество строковых параметров (string) или **None** (Если **None** – первый параметр, то запись команды будет произведена без параметров. В остальных случаях, **None** игнорируется при записи);

Выходные параметры:

* Массив:
  + Строковый статус выполнения команды по внутреннему протоколу Набора;
  + Ответ УСПД (если есть), пропущенный через транслятор протокола УСПД;

##### Метод **read\_from\_device\_rdiagn(\*params)**

Метод чтения для команды диагностики. Отправляет на УСПД команду диагностики, получает и обрабатывает ответ.

Входные параметры:

* Неограниченное количество строковых параметров (string) или **None** (Если **None** – первый параметр, то запись команды будет произведена без параметров. В остальных случаях, **None** игнорируется при записи);

Выходные параметры:

* Массив:
  + Строковый статус выполнения команды по внутреннему протоколу Набора;
  + Ответ УСПД (если есть), пропущенный через транслятор протокола УСПД, представляет собой словарь (dictionary) со следующими элементами:
    - **dev** – тип УСПД;
    - **fwrev** – ревизия ВПО;
    - **ops** – оператор связи;
    - **scid** – номер SIM-карты;
    - **imei** – IMEI модема;
    - **reg** – тип регистрации в сети;
    - **df1** – тип микросхемы памяти 1;
    - **df2** – тип микросхемы памяти 2;
    - **fram** – тип микросхемы FRAM;
    - **hw\_info** – строка информации об исполнении;

##### Метод **reboot\_to\_fw()**

Метод вывода УСПД из режима загрузчика в режим основной программы.

Входные параметры:

* Нет входных параметров;

Выходные параметры:

* Массив:
  + Строковый статус выполнения команды по внутреннему протоколу Набора;

##### Метод **defaultconfig(\*params)**

Метод выполняет сброс УСПД до заводских настроек и восстанавливает пароли доступа на заданные в файле настроек после успешного сброса настроек.

Входные параметры:

* Неограниченное количество строковых параметров (string) или **None** (Если **None** – первый параметр, то запись команды будет произведена без параметров. В остальных случаях, **None** игнорируется при записи);

Выходные параметры:

* Массив:
  + Строковый статус выполнения команды по внутреннему протоколу Набора;

##### Метод **tranzit\_off(speed, length, parity, stop)**

Метод отключения транзитного режима.

Входные параметры:

* **speed** – кодированное согласно протоколу УМ-RTU строковое (string) значение скорости открытого транзитного режима;
* **length** – кодированное согласно протоколу УМ-RTU строковое (string) значение длина кадра данных;
* **parity** – кодированное согласно протоколу УМ-RTU строковое (string) значение четности;
* **stop** – кодированное согласно протоколу УМ-RTU строковое (string) значение количества стоповых бит данных;

Выходные параметры:

* Массив:
  + Строковый статус выполнения команды по внутреннему протоколу Набора;
  + Ответ УСПД (если есть), пропущенный через транслятор протокола УСПД;

##### Метод **port\_flush()**

Служит для очистки порта от мусорных данных. Например, после перезагрузки УСПД.

Входные параметры:

* Нет входных параметров;

Выходные параметры:

* Метод ничего не возвращает;

##### Метод **end\_session()**

Закрывает порт.

Входные параметры:

* Нет входных параметров;

Выходные параметры:

* Метод ничего не возвращает;

### Интерфейс записи логов обмена по COM-порту (logger.py)

Интерфейс представляет собой класс Log, содержащий метод записи лога в файл. Файл именуется согласно ответной команде и типа теста. Файлы создаются во вложенной папке Log.

Свойства класса:

* **answer\_command** – ответная команда в строковом формате (string);
* **test\_name** – тип теста в строковом формате (string);

##### Метод **write\_log(request, answer)**

Служит для записи отправленных и полученных с COM-порта данных в файл лога.

Входные параметры:

* **request** – данные запроса к УСПД в строковом формате (string);
* **answer** – данные ответа от УСПД в строковом формате (string);

Выходные параметры:

* Метод ничего не возвращает;

##### Метод **write\_log\_info(info)**

Служит для записи произвольного сообщения в файл лога.

Входные параметры:

* **info** – данные запроса к УСПД в строковом формате (string);

Выходные параметры:

* Метод ничего не возвращает;

## Интерпретация протокола обмена УМ-RTU

### Интерпретатор протокола(txtProto.py)

Интерпретатор представляет собой класс TxtProto. Содержит методы генерации команд к УСПД и разбора ответов УСПД. Методы класса учитывают контрольную сумму, согласно протоколу УМ-RTU.

Свойства класса:

* **pwd** – пароль доступа к УСПД в строковом формате (string);

##### Метод **parse(respond)**

Разбирает строку ответа УСПД.

Входные параметры:

* Строка ответа УСПД (string);

Выходные параметры:

* Массив:
  + При успешном разборе - ответная команда УСПД, при ошибке – строковый статус выполнения согласно внутреннему протоколу Набора;
  + Аргументы ответной команды в строковом формате (string): каждый аргумент как отельный элемент массива в порядке следования их в ответе;

##### Метод **parse\_old(respond)**

Разбирает строку ответа УСПД, где разделителями являются знаки «пробел», «.», «:». Применимо для команд даты-времени.

Входные параметры:

* Строка ответа УСПД (string)

Выходные параметры:

* Массив:
  + При успешном разборе - ответная команда УСПД, при ошибке – строковый статус выполнения согласно внутреннему протоколу Набора;
  + Аргументы ответной команды в строковом формате (string): каждый аргумент как отельный элемент массива в порядке следования их в ответе;

##### Метод **parse\_rdiagn(respond)**

Разбирает строку ответа УСПД на команду запроса диагностики.

Входные параметры:

* Строка ответа УСПД (string);

Выходные параметры:

* Массив:
  + При успешном разборе - ответная команда УСПД, при ошибке – строковый статус выполнения согласно внутреннему протоколу Набора;
  + Свойства дигностики в формате словаря (dictionary):
    - **dev** – тип УСПД;
    - **fwrev** – ревизия ВПО;
    - **ops** – оператор связи;
    - **scid** – номер SIM-карты;
    - **imei** – IMEI модема;
    - **reg** – тип регистрации в сети;
    - **df1** – тип микросхемы памяти 1;
    - **df2** – тип микросхемы памяти 2;
    - **fram** – тип микросхемы FRAM;
    - **hw\_info** – строка информации об исполнении;

##### Метод **make(\*command)**

Генерирует команду протокола УСПД из данных параметров.

Входные параметры:

* Массив, где первый элемент – команда текстового протоколу УМ-RTU в строковом формате (string), последующие – аргументы команды текстового протоколу УМ-RTU в строковом формате (string) или **None**;

Выходные параметры:

* Строка с командой (string), соответствующая текстовому протоколу УМ-RTU;

### Счетчик контрольной суммы (CRC16RTU.py)

Представляет собой класс Crc16. Осуществляет расчет контрольной суммы полученной строки согласно протоколу УМ-RTU.

##### Метод **makeCRC16(buf)**

Рассчитывает контрольную сумму полученной строки.

Входные параметры:

* **buf** – строка для расчета контрольной суммы в строковом формате (string) или формате байтовой строки (bytes);

Выходные параметры:

* Контрольная сумма в формате байтовой строки (bytes);

## Настройки Набора

### Парсер настроек (setup.py)

Парсер представляет собой класс Settings, осуществляющий парсинг XML документа settings.xml. Каждая настройка из файла settings.xml считывается в одноименное свойство класса Settings для дальнейшего использования по мере надобности. Настройки описаны в разделе [Подробное описание настроек тестов](#_Подробное_описание_настроек).

## Описание шаблона теста

Тесты основаны на фреймворке тестирования pytest. Возможно создавать тесты, удовлетворяющие формату фреймворка pytest, не ограничиваясь описанным шаблоном.

### Блок параметризации

Блок параметризации содержит фикстуры параметров и их значения для каждой итерации теста.

Пример для 2-х итераций с 2-мя фикстурами:

@pytest.mark.parametrize('фикстура1,фикстура2', [

('значение\_фикстуры1-1', 'значение\_фикстуры2-1'),

('значение\_фикстуры1-2', 'значение\_фикстуры2-2')])

### Класс Test\_proto

Класс теста содержит непосредственно вызываемыми при выполнении теста методы.

##### Метод **setup\_class()**

Метод служит для инициализации классов работы с УСПД и хранения глобальных параметров теста. Вызывается единожды при запуске теста.

##### Метод **setup\_method()**

Служит для подготовки окружения к выполнению теста. Вызывается перед каждой итерацией теста.

##### Метод **test\_run(фикстура1, фикстура2, …)**

Метод непосредственно теста. Единственный метод, способный принимать фикстуры параметризации в качестве входных параметров.

##### Метод **teardown\_method()**

Служит для совершения завершающих действий с окружением, не входящих в тест. Вызывается в после каждой итерации теста.

##### Метод **teardown\_class()**

Служит для совершения завершающих действий с окружением, не входящих в тест. Вызывается после всех итераций.

# Приложение 1. Описание служебного протокола Набора.

|  |  |
| --- | --- |
| **Строка** | **Значение** |
| OK | Ошибки отсутствуют |
| ERROR | Получена ошибка |
| CRC\_ERROR | Ошибка контрольной суммы полученной строки |
| NO\_DATA | Отсутствуют данные там, где они ожидались, или данные имеют неизвестный формат |

# Приложение 2. Статус покрытия команд протокола УМ-RTU тестами

|  |  |
| --- | --- |
| **Команда** | **Статус** |
| RNAME | OK |
| WNAME | OK |
| GETSWVER | OK |
| GETHWVER | OK |
| GETBLVER | OK |
| SPASS1 | OK |
| SPASS2 | OK |
| SPASS3 | OK |
| WSIMPIN | OK |
| UM\_READ\_ID | OK |
| WRTUPWD | OK |
| RRTUPWD | OK |
| WERROREX | OK |
| RERROREX | OK |
| DEFAULTCONFIG | OK |
| WCRCCTRL | OK |
| RCRCCTRL | OK |
| WCMDIFACE | Отложено |
| RCMDIFACE | Отложено |
| GETDATETIME | OK |
| GETDATETIMEEX | OK |
| SETDATETIME | OK |
| WTIMEZONE | OK |
| RTIMEZONE | OK |
| WTIMESYNCLIM | OK |
| RTIMESYNCLIM | OK |
| WTSYNSRVADDR | OK |
| RTSYNCSRVADDR | OK |
| TRANSOPEN | OK |
| TRANSROLLBACK | Отложено |
| TRANSCOMMIT | OK |
| TRANSADD | OK |
| TRANSUPD | OK |
| TRANSDEL | Отложено |
| READTABLX | OK |
| RTABLLEN | Отложено |
| RTABLMAXLEN | OK |
| CLEARTABL | OK |
| WGSMACT | OK |
| RGSMACT | OK |
| WAGPRS | OK |
| RAGPRS | OK |
| EXRAGPRS | OK |
| WADNS | OK |
| RADNS | OK |
| WRTUPORT | OK |
| RRTUPORT | OK |
| WHTTPAUTH | OK |
| RHTTPAUTH | OK |
| WHTTPSRVADDR | OK |
| RHTTPSRVADDR | OK |
| WHTTPOPS | OK |
| RHTTPOPS | OK |
| WSMTPAUTH | OK |
| RSMTPAUTH | OK |
| WSMTPSRVADDR | OK |
| RSMTPSRVADDR | OK |
| WSMTPMAIL | OK |
| RSMTPMAIL | OK |
| WMAILARCHPRM | OK |
| RMAILARCHPRM | OK |
| WTCPCLSRVADDR | OK |
| RTCPCLSRVADDR | OK |
| WTCPCLAUTH | OK |
| RTCPCLAUTH | OK |
| WTCPCLOPS | OK |
| RTCPCLOPS | OK |
| WARCHPRM | OK |
| WARCHPRMEX | OK |
| RARCHPRM | OK |
| WARCHDEPTH | OK |
| WARCHDEPTHEX | OK |
| RARCHDEPTH | OK |
| WRITEITEMPARAM | OK |
| READITEMPARAM | OK |
| READCQUAL | Отложено |
| READSERIAL | Отложено |
| READDAY | Отложено |
| READDAYCONS | Отложено |
| READMONTH | Отложено |
| READMONTHCONS | Отложено |
| READSTATE | Отложено |
| READAQUAL | Отложено |
| READINTJRNL | Отложено |
| WMTRJRNLMODE | OK |
| RMTRJRNLMODE | OK |
| READMTRJRNL | Отложено |
| SETRELAY | Отложено |
| GETRELAY | Отложено |
| SETMETERTIME | Отложено |
| WDISPPHONE | OK |
| RDISPPHONE | OK |
| RCURSMSSCA | OK |
| WSMSSCA | OK |
| RSMSSCA | OK |
| WSCHEDMODE | OK |
| RSCHEDMODE | OK |
| WSCHEDSYNMETER | OK |
| RSCHEDSYNMETER | OK |
| WSCHEDTIMESYNC | OK |
| RSCHEDTIMESYNC | OK |
| WSCHEDHTTP | OK |
| RSCHEDHTTP | OK |
| RSCHEDMAICUR | OK |
| WSCHEDMAICUR | OK |
| WSCHEDMAILDAY | OK |
| RSCHEDMAILDAY | OK |
| RSCHEDMAILDAYCONS | OK |
| WSCHEDMAILDAYCONS | OK |
| RSCHEDMAILMON | OK |
| WSCHEDMAILMON | OK |
| RSCHEDMAILMONCONS | OK |
| WSCHEDMAILMONCONS | OK |
| RSCHEDMAILHHCONS | OK |
| WSCHEDMAILHHCONS | OK |
| RSCHEDMAILAQUAL | OK |
| WSCHEDMAILAQUAL | OK |
| RSCHEDMAILMJRNL | OK |
| WSCHEDMAILMJRNL | OK |
| RSCHEDMAILIJRNL | OK |
| WSCHEDMAILIJRNL | OK |
| RSCHEDMAILDI | OK |
| WSCHEDMAILDI | OK |
| RSCHEDTCPCLIENT | OK |
| WSCHEDTCPCLIENT | OK |
| RSCHEDAQUAL | OK |
| WSCHEDAQUAL | OK |
| WDIINIT | OK |
| RDIINIT | OK |
| WDIALARM | OK |
| RDIALARM | OK |
| WDIMAILALARM | OK |
| RDIMAILALARM | OK |
| WDIHTTPALARM | OK |
| RDIHTTPALARM | OK |
| WDITCPALARM | OK |
| RDITCPALARM | OK |
| RDISTATE | OK |
| WPWRIFACEMODE | OK |
| RPWRIFACEMODE | OK |
| TRANZIT\_ON | OK |
| WDEFTRANZITPRM | OK |
| RDEFTRANZITPRM | OK |
| WTRANZITCFG | OK |
| RTRANZITCFG | OK |
| STARTSESSION | OK |
| SWRESET | OK |
| RDIAGN | OK |
| WVERIFYMODE | OK |
| RVERIFYMODE | OK |
| FWUPD | OK |
| WFWUPDAUTH | OK |
| RFWUPDAUTH | OK |
| WFWUPDSRVADDR | OK |
| RFWUPDSRVADDR | OK |
| WFWUPDOPS | OK |
| RFWUPDOPS | OK |