

АННОТАЦИЯ

Настоящий документ содержит описание ПП «СКАДА А-СОФТ» (далее по тексту СКАДА).

Руководство оператора ПП «СКАДА А-СОФТ» состоит из пяти частей (отдельных документов):

- 1) Руководство оператора, часть 1 - Общее описание ПП «СКАДА А-СОФТ»;
- 2) Руководство оператора, часть 2 - Редактор пользовательского интерфейса ПП «СКАДА А-СОФТ»;
- 3) Руководство оператора, часть 3 - Создание проекта в ПП «СКАДА А-СОФТ». Организация работы с источниками данных;
- 4) Руководство оператора, часть 4 - Работа с протоколом EN ПП «СКАДА А-СОФТ»;
- 5) Руководство оператора, часть 5 - Общесистемное API пользовательского программирования.

СОДЕРЖАНИЕ

1. Назначение программы	5
2. Условия выполнения программы	6
2.1 Состав аппаратных средств	6
2.2 Состав программных средств	6
2.3 Требования к квалификации персонала	6
3. Выполнение программы	6
3.1 Запуск программы	7
3.2 Конфигурация системных параметров	7
3.3 Подсистема "БД"	14
3.3.1 Общие сведения.....	14
3.3.2 Конфигурирование подсистемы «БД»	15
3.4 Подсистема "Безопасность"	23
3.4.1 Общие сведения.....	23
3.4.2 Конфигурирование подсистемы «Безопасность»	23
3.5 Подсистема "Транспорты"	39
3.5.1 Общие сведения.....	39
3.5.2 Конфигурирование подсистемы «Транспорты»	41
3.6 Подсистема "Транспортные протоколы"	50
3.6.1 Общие сведения.....	50
3.6.2 Модуль «OPC UA».....	51
3.6.3 Модуль «HTTP»	53
3.6.4 Модуль «ModBUS»	54
3.6.5 Модуль «Собственный протокол системы СКАДА» (SelfSystem)	56
3.6.6 Модуль «Пользовательский протокол» (UserProtocol).....	56
3.6.7 Конфигурирование подсистемы «Транспортные протоколы»	58
3.7 Подсистема "Сбор данных"	59
3.7.1 Общие сведения.....	59
3.7.2 Конфигурирование подсистемы «Сбор данных».....	60
3.8 Подсистема «Проверка данных».....	81
3.9 Подсистема «Архивы»	86
3.9.1 Общие сведения.....	86
3.9.2 Архиватор на БД	87
3.9.2.1 Архиватор сообщений.....	87
3.9.2.2 Архиватор значений	88
3.9.3 Архиватор на ФС.....	89
3.9.3.1 Архиватор сообщений.....	89
3.9.3.2 Формат файлов архива сообщений	91
3.9.3.3 Архиватор значений	92
3.9.3.4 Формат файлов архива значений	92
3.9.4 Конфигурирование подсистемы «Архивы».....	95
3.10 Подсистема "Пользовательские интерфейсы"	109

3.10.1	Общие сведения.....	109
3.10.2	Рабочий пользовательский интерфейс (QT).....	110
3.10.3	Конфигурирование модуля «Рабочий пользовательский интерфейс».....	112
3.10.4	Системный конфигуриратор (QT).....	114
3.10.5	QT GUI пускатель.....	115
3.10.6	Движок среды визуализации и управления.....	116
3.10.7	Модули, реализованные на основе WEB-технологий.....	120
3.11	Подсистема "Специальные".....	123
3.11.1	Общее описание.....	123
3.11.2	Библиотека стандартных математических функций.....	125
3.11.3	Библиотека функций системного API.....	126
3.11.4	Тесты ПП «СКАДА А-СОФТ».....	127
3.12	Подсистема "Управление модулями".....	129
3.13	Конфигурационный файл СКАДА и параметры командной строки вызова СКАДА.....	131
4.	Дополнительные возможности пользовательского интерфейса.....	133
4.1	Описание журнала активных тревог и журнала тревог и событий.....	133
4.1.1	Функциональное назначение.....	133
4.1.2	Описание журнала активных тревог и журнал тревог и событий.....	133
4.1.3	Взаимодействие журнала активных тревог и журнала тревог и событий с модулями СКАДА.....	133
4.1.4	Элементы интерфейса журнала активных тревог и журнал тревог и событий.....	134
4.2	Описание быстрого перехода по видеокадрам.....	139
4.2.1	Функциональное назначение.....	139
4.2.2	Элементы интерфейса для быстрого перехода по видеокадрам.....	139
4.2.3	Основные особенности быстрого перехода между видеокадрами.....	139
	Приложение 1.....	140
	Приложение 2.....	148
	Перечень принятых сокращений.....	154
	2 – Программная платформа «СКАДА А-СОФТ». Редактор пользовательского интерфейса ПП «СКАДА А-СОФТ». Руководство оператора.	
	3 – Программная платформа «СКАДА А-СОФТ». Создание проекта. Организация работы с источниками данных. Руководство оператора.	
	4 – Программная платформа «СКАДА А-СОФТ». Работа с протоколом EN. Руководство оператора.	
	5 – Программная платформа «СКАДА А-СОФТ». Общесистемное API пользовательского программирования. Руководство оператора.	

1. НАЗНАЧЕНИЕ ПРОГРАММЫ

ПП «СКАДА А-СОФТ» предназначена для разработки и исполнения прикладного программного обеспечения, предназначенного для сбора, архивирования, визуализации информации, выдачи управляющих воздействий, а также других родственных операций, характерных для полнофункциональной системы управления и сбора данных.

2. УСЛОВИЯ ВЫПОЛНЕНИЯ ПРОГРАММЫ

2.1 Состав аппаратных средств

Аппаратные требования СКАДА для её исполнения в зависимости от сложности проекта приведены в таблице 1.

Таблица 1

Требования к аппаратной части	
<i>Проекты более 30000 точек</i>	
АРМ	Процессор: Intel Core i3 2 ГГц; ОЗУ: 8 Гб; Жесткий диск: 40 Гб
Сервер	Процессор: Intel Xeon E5; ОЗУ: 128 Гб; Жесткий диск: 40 Гб
Сервер БД	Процессор: Intel Xeon E5; ОЗУ: 64 Гб; Жесткий диск: 1 Тб, RAID 10
Шлюз	Процессор: Intel Core i3 2 ГГц; ОЗУ: 2 Гб; Жесткий диск: 40 Гб
<i>Проекты до 5000 точек</i>	
АРМ-сервер	Процессор: Intel Core i7 3 ГГц; ОЗУ: 16 Гб; Жесткий диск: 300 Гб, RAID 5
Шлюз	Процессор: Intel Core i3 2 ГГц; ОЗУ: 2 Гб; Жесткий диск: 40 Гб
<i>Проекты до 500 точек</i>	
АРМ-сервер	Процессор: Intel Core i5 2 ГГц; ОЗУ: 16 Гб; Жесткий диск: 300 Гб

2.2 Состав программных средств

Для обеспечения работоспособности ПП «СКАДА А-СОФТ» необходимо наличие операционной системы Astra Linux SE Smolensk 1.5 (x64) или Astra Linux SE Smolensk 1.6.

2.3 Требования к квалификации персонала

Специалисты, занимающиеся настройкой программного обеспечения «СКАДА А-СОФТ», должны иметь знания для работы с системой Astra Linux на уровне пользователя.

3. ВЫПОЛНЕНИЕ ПРОГРАММЫ

Конфигурирование СКАДА осуществляется при помощи модуля конфигурации - UI.QTScfg. Данный модуль предоставляет развитый интерфейс конфигурации позволяющий

управлять, как локальной станцией, так и удалёнными станциями в локальной и глобальной сетях, включая безопасное соединение.

Значения конфигурации, изменённые в конфигураторе, а также большинство данных сохраняются в базах данных (БД). Учитывая модульность подсистемы "БД", ими могут быть различные БД. Причём предоставляется возможность хранения разных частей СКАДА как в разных БД одного типа, так и в БД разных типов.

Кроме БД данные о конфигурации могут содержаться в конфигурационном файле СКАДА, а также передаваться посредством параметров командной строки при вызове СКАДА. Сохранение конфигурации в конфигурационном файле осуществляется наравне с БД. Типовым именем конфигурационного файла является `/etc/scada.xml`. Формат конфигурационного файла и параметры командной строки рассматриваются в 3.13.

Многие настройки и конфигурация объектов СКАДА, которые исполняются или уже включены, не применяются сразу же по внесению изменений, поскольку конфигурация читается/применяется обычно только при включении или запуске. Следовательно, для применения изменений в таких случаях, достаточно включить/выключить включенный объект или перезапустить исполняющийся — остановить/запустить.

3.1 Запуск программы

Запуск СКАДА осуществляется командой `scada` из командной строки терминала или из графического меню «Пуск» → «Графика» → «SCADA» с помощью мыши.

После загрузки системы появится окно регистрации (рисунок 1).

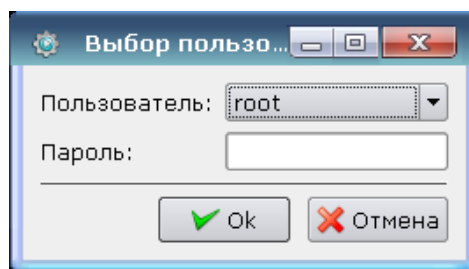


Рисунок 1

После регистрации в СКАДА (пользователь: `root`, пароль: `root`) на экране появится окно системного конфигуратора СКАДА (рисунок 2).

3.2 Конфигурация системных параметров

Конфигурация системных параметров размещается на пяти вкладках корневой страницы станции.

Вкладка "Станция" содержит основные информационные и конфигурационные параметры станции. Ее вид показан на рисунке 2.

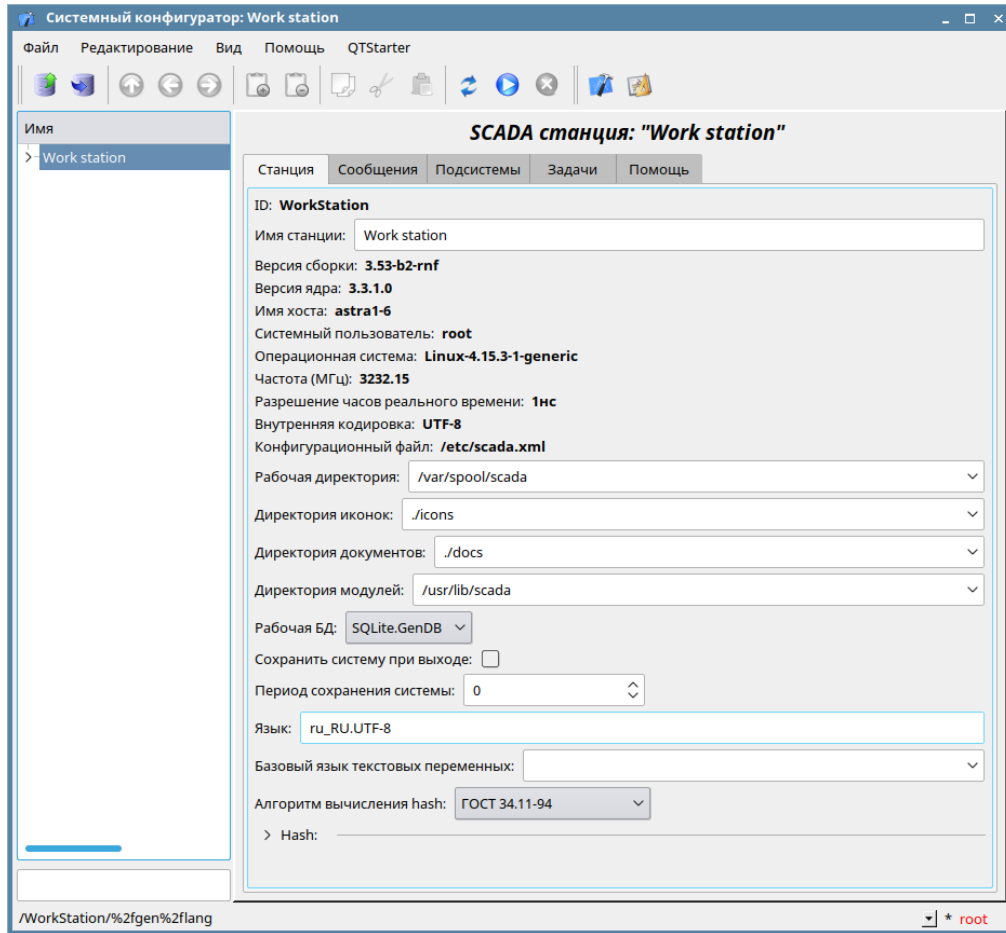


Рисунок 2

На вкладке "Станция" представлены следующие поля:

- *ID* - содержит информацию об идентификаторе станции. Указывается параметром командной строки - Station. При загрузке ищется соответствующий идентификатору станции раздел в конфигурационном файле и, если не обнаруживается, то используется первый доступный;
- *Имя станции* - указывает локализованное имя станции;
- *Версия сборки* - содержит информацию о текущей версии программы;
- *Версия ядра* - содержит информацию о текущей версии ядра системы;
- *Имя хоста* - содержит информацию об имени машины, на которой запущена станция.
- *Системный пользователь* - содержит информацию о пользователе, от имени которого выполняется программа в ОС;
- *Операционная система* - содержит информацию об имени и версии ОС, ядре ОС, на которой исполняется программа;

- *Частота (МГц)* - содержит информацию о частоте центрального процессора, которым исполняется программа. Значение частоты проверяется раз в 10 секунд и позволяет отслеживать её изменение, например, механизмами управления питанием;

- *Разрешение часов реального времени (нс)* - содержит информацию о возможности или разрешении часов реального времени ОС. Позволяет сориентироваться с минимальным интервалом времени периодических задач, например, для задач сбора данных;

- *Внутренняя кодировка* - содержит информацию о кодировке, в которой хранятся текстовые сообщения внутри программы;

- *Конфигурационный файл* - содержит информацию о конфигурационном файле, используемом программой. Устанавливается параметром командной строки – Config;

- *Рабочая директория* - указывает на рабочую директорию станции. Используется в относительной адресации объектов на файловой системе, например, файлов БД. Допускает изменение пользователем для сохранения данных системы в другую БД. При этом значение этого поля не сохраняется в БД, а может быть изменено только в секции "WorkDB" конфигурационного файла;

- *Директория иконок* - указывает на директорию, содержащую иконки программы. Если в дереве навигации конфигуратора отсутствуют иконки, то неправильно указано значение этого поля;

- *Директория документов* - указывает на директорию, содержащую справочную документацию по СКАДА;

- *Директория модулей* - указывает на директорию модулей для СКАДА. Если значение этого поля некорректно, то при запуске на экране будет отображена только информация в консоли о корректном запуске ядра СКАДА;

- *Рабочая БД* - указывает на рабочую базу данных (БД), а именно на БД, используемую для хранения основных данных программы. Изменение этого поля отмечает все объекты программы как модифицированные, что позволяет сохранить или загрузить данные станции из указанной основной БД;

- *Сохранять систему при выходе* - указывает на необходимость сохранения изменённых данных при завершении работы;

- *Период сохранения системы* - указывает период в секундах, с которым необходимо сохранять изменённые данные станции;

- *Язык* - указывает на язык сообщений программы. Изменение этого поля допустимо, однако приводит к изменению языка сообщений только для интерфейса и динамических сообщений;

- *Базовый язык текстовых переменных* - используется для включения режима поддержки многоязыковых текстовых переменных. Значение базового языка выбирается из списка двухсимвольных кодов языков, обычно только текущий и базовый языки в списке. Далее для текстовых переменных на небазовом языке в таблицах БД будут создаваться отдельные колонки. Под текстовыми переменными подразумеваются все текстовые поля конфигулятора, которые могут быть переведены на другой язык. Числа и другие символьные значения к их числу не относятся и не переводятся;

- *Алгоритм вычисления hash* – позволяет выбрать алгоритм вычисления контрольной суммы элемента. Для выбора доступны алгоритмы в соответствии с ГОСТ 34.11-94 и ГОСТ 34.11-2012. Подсчет контрольных сумм осуществляется в режиме исполнения и выводится в окно системного конфигулятора для следующих модулей: Сбор данных – Modbus, OPC, Логический уровень; Базы данных; Транспорт; Редактор графического интерфейса, а также вычисляется общая контрольная сумма для всех контролируемых конфигураций. Вкладка "Сообщения" - это раздел группы параметров, управляющих работой с сообщениями станции (рисунок 3), включает в себя:

- *Таблица «Уровни сообщений»* – указывает на уровень сообщений, начиная с которого необходимо их обрабатывать. Сообщения ниже этого уровня будут игнорироваться. Необходимо, например, для исключения из обработки отладочных сообщений уровня 0. Для каждого уровня сообщения возможно выбрать логирование – указав «True» в соответствующей ячейке;

- *В системный логер (syslog)* – указывает на необходимость направления сообщений в системный логер, механизм ОС для работы с сообщениями системы и ПО. При включении этого параметра появляется возможность управлять и контролировать сообщения СКАДА механизмами ОС;

- *На стандартный выход (stdout)* – указывает на использование стандартного механизмы вывода в консоль. Выключение этого свойства исключит весь вывод в консоль, если не указан следующий параметр;

- *На стандартный выход ошибок(stderr)* – указывает на использование стандартного механизма вывода ошибок, обычно тоже направляется в консоль;

- *В архив* – указывает на необходимость вывода сообщений в архив сообщений СКАДА. Этот параметр обычно включен, а его выключение приводит к фактическому отключению архивирования сообщений на станции.

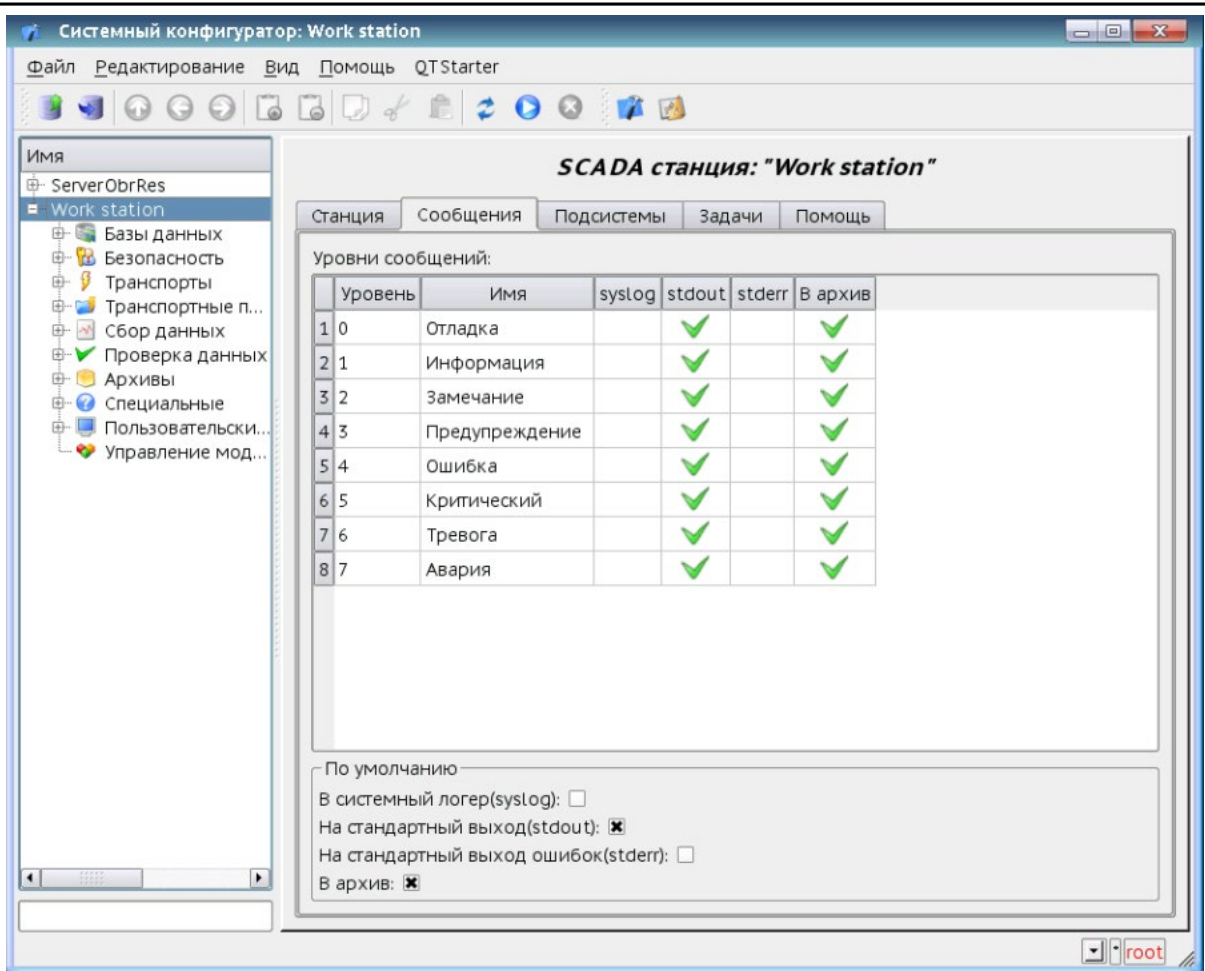


Рисунок 3

Вкладка "Подсистемы" содержит список подсистем и позволяет выполнять прямые переходы к ним с помощью контекстного меню. Ее вид показан на рисунке 4.

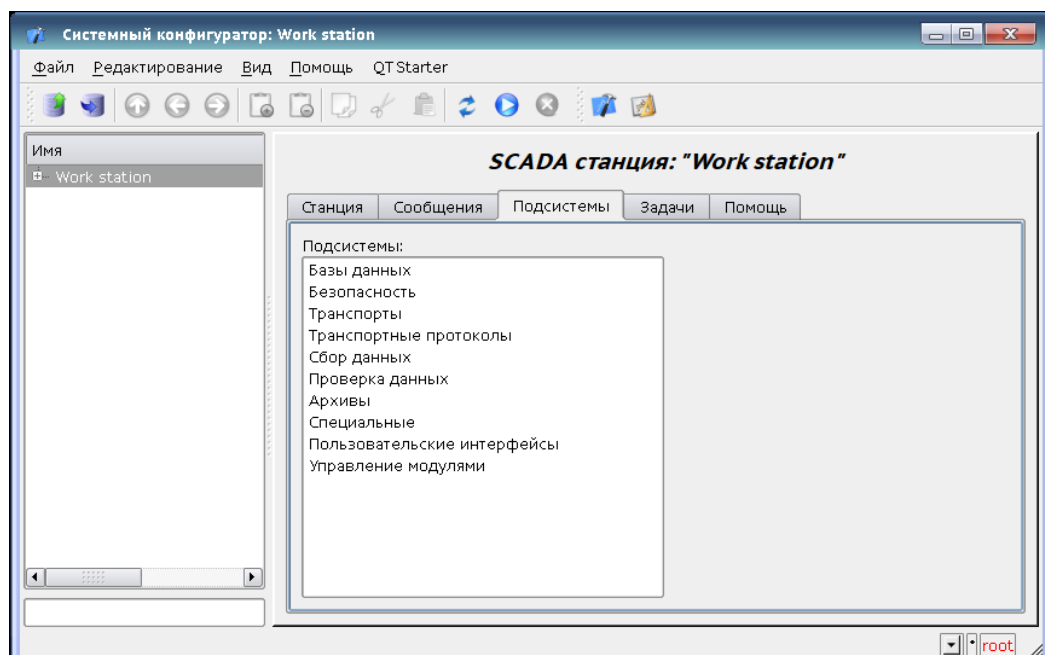


Рисунок 4

Вкладка "Задачи" содержит таблицу со списком задач открытых различными компонентами СКАДА. Из таблицы можно получить различную информацию о задачах, а также в колонке «УСТ.СРU» назначить процессоры для задач, на многопроцессорных системах.

Вид вкладки "Задачи" показан на рисунке 5.

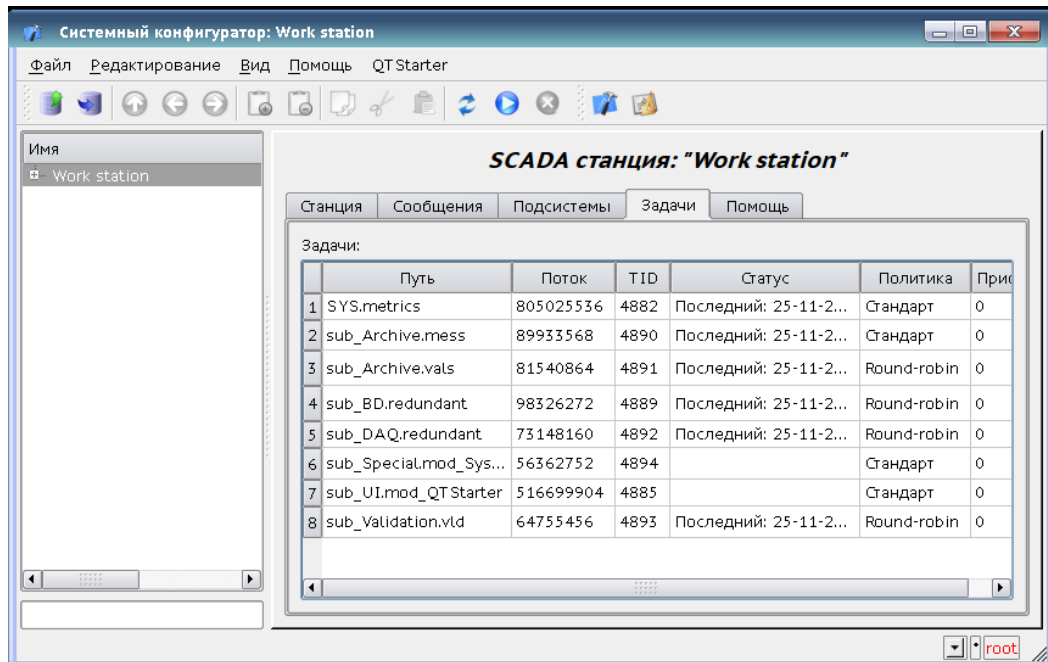


Рисунок 5

Вкладка "Помощь" содержит краткую помощь для данной страницы. Ее вид показан на рисунке 6.

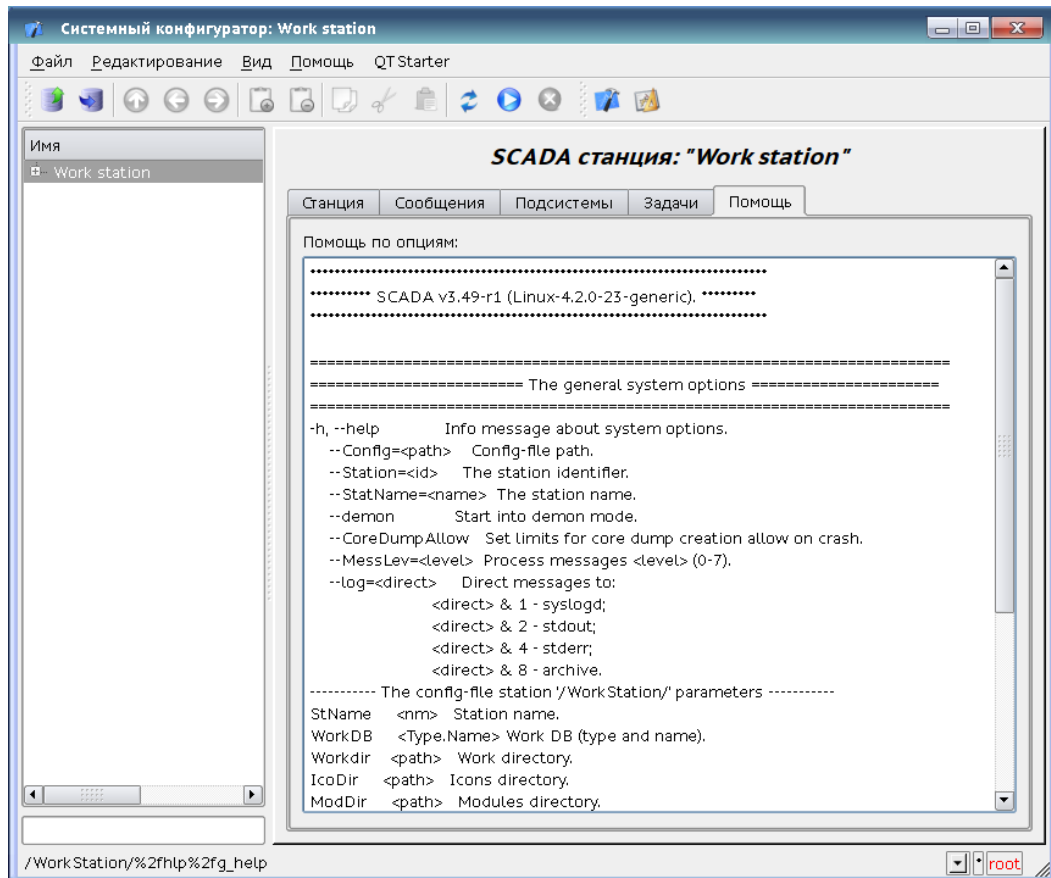


Рисунок 6

Кроме того, справочная информация о работе с системой, конфигурировании и настройкам модулей подсистемы доступны для чтения в пункте меню «Помощь» → «Справка» или нажатием клавиши «F1» .

Примечание. Для модификации полей корневой страницы станции могут потребоваться права привилегированного пользователя. Получить такие права можно, включив пользователя в группу суперпользователя "root", или, войдя на станцию от имени суперпользователя "root".

Поля идентификаторов всех объектов СКАДА недопустимы для прямого редактирования, поскольку являются ключом для хранения данных объектов в БД. Поменять идентификатор объекта можно с помощью команды переноса и последующей вставки объекта в конфигураторе.

3.3 Подсистема "БД"

3.3.1 Общие сведения

Для хранения данных системы используются базы данных (БД). В целях систематизации доступа и управления базами данных в системе СКАДА предусмотрена подсистема "Базы данных". Для обеспечения поддержки различных БД/СУБД подсистема выполнена модульной.

В роли модульных объектов, содержащихся в подсистеме, выступает тип БД/СУБД, т.е. модуль подсистемы «Базы данных» практически содержит реализацию доступа к определённому типу БД: PostgreSQL или SQLite.

Объект типа БД/СУБД, в свою очередь, содержит список объектов отдельных БД данного типа, а объект БД содержит список объектов таблиц, которые и содержат данные в табличной форме.

Практически все данные СКАДА хранятся в той или иной БД. Инструментарий системы позволяет легко переносить данные из одного типа БД в другой, и, как следствие, оптимально подбирать тип БД под конкретную область применения СКАДА. Перенос информации с одной БД в другую может быть выполнен двумя способами. Первый – это изменение адреса рабочей БД и сохранение всей системы на неё, второй – это прямое копирование информации между БД. Кроме копирования поддерживается и функция прямого редактирования содержимого таблиц БД.

Данные могут храниться также в конфигурационном файле системы. Реализован механизм полного отражения структуры БД на структуру конфигурационного файла. Т.е. стандартную конфигурацию можно размещать в конфигурационном файле. Суть такого механизма в том, что данные системы по умолчанию, например, при старте без БД можно описывать в конфигурационном файле. В дальнейшем эти данные могут переопределяться в БД. Кроме этого, для случаев невозможности запуска какой либо БД вообще, можно все данные хранить в конфигурационном файле.

Для доступа к базам данных используется механизм регистрации БД. Зарегистрированные в системе БД доступны всем подсистемам системы СКАДА и могут использоваться в их работе. Благодаря этому механизму можно обеспечить распределённость хранения данных. Например, различные библиотеки могут храниться и распространяться независимо, а подключение библиотеки будет заключаться в простой регистрации нужной БД.

3.3.2 Конфигурирование подсистемы «БД»

Для конфигурации подсистемы предусмотрена корневая страница подсистема "БД", содержащая вкладки "Резервирование", "Модули" и "Помощь". Вкладка "Резервирование" содержит поля для настройки резервирования БД. Ее вид показан на рисунке 7. Пример настройки резервирования БД описан в части 3 настоящего Руководства оператора. Вкладка "Модули" содержит список модулей подсистемы "БД", доступных на станции.

Вкладка "Помощь" содержит краткую помощь для данной страницы.

Для модификации полей страниц этой подсистемы могут потребоваться права привилегированного пользователя или включение пользователя в группу "БД".

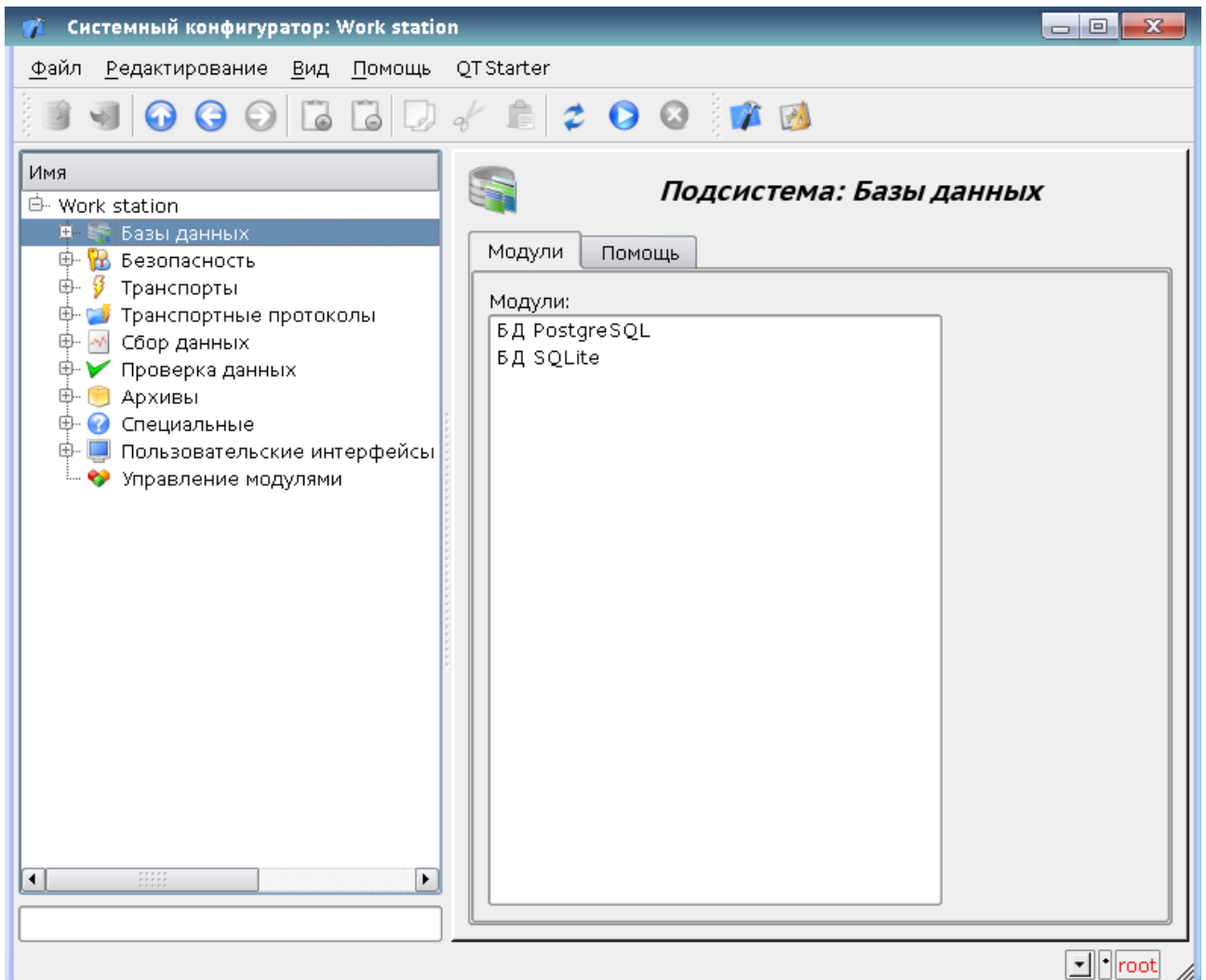


Рисунок 7

Каждый модуль подсистемы "БД" предоставляет конфигурационную страницу с вкладками "БД" и "Помощь". Вкладка "БД" содержит список БД, зарегистрированных в модуле, и флажок признака полного удаления БД (рисунок 8). Если этот флажок установлен, БД будет полностью удалена при её закрытии. Иначе БД будет просто закрыта.

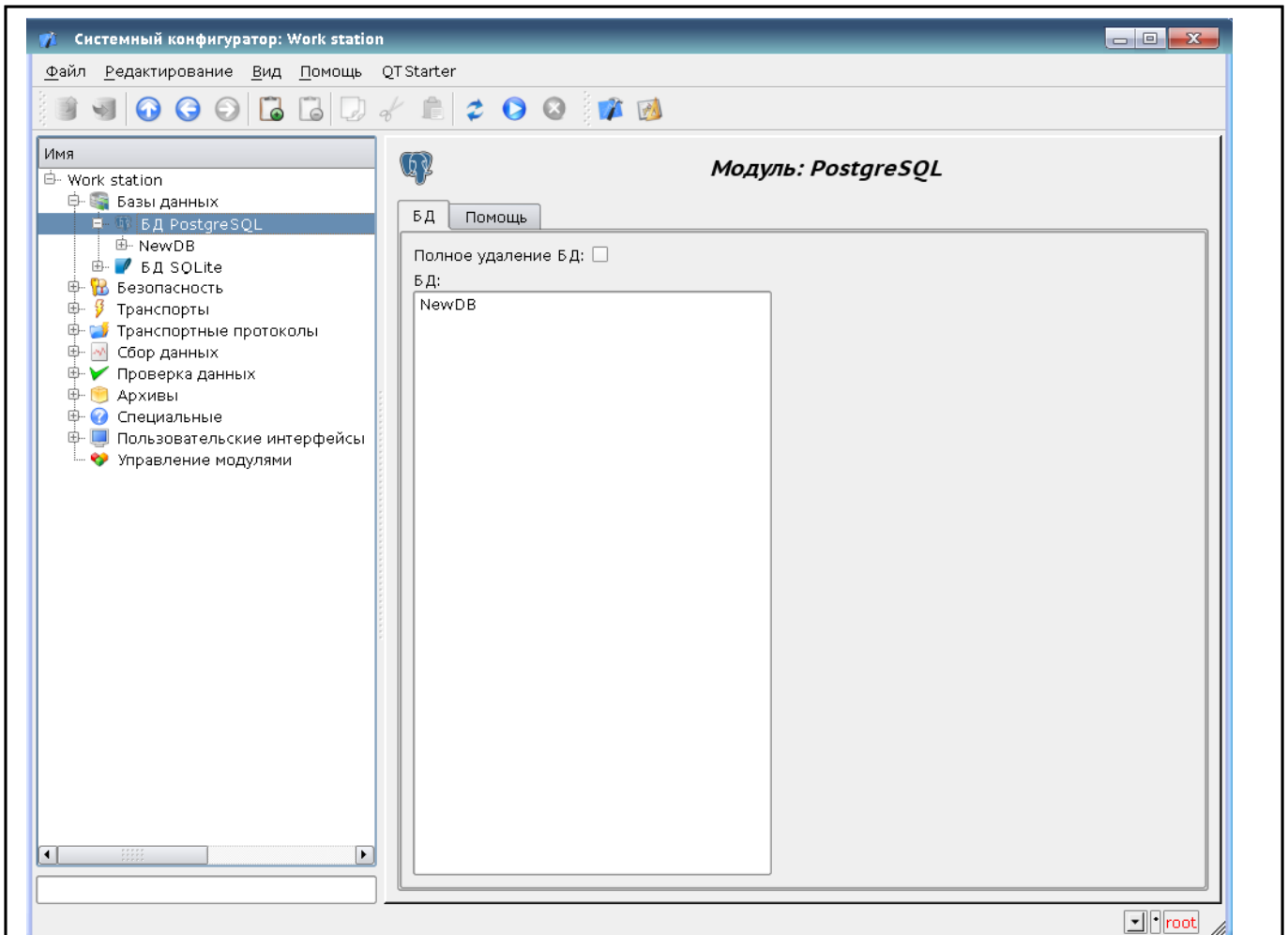


Рисунок 8

В контекстном меню списка БД пользователю предоставляется возможность добавления, удаления и перехода к нужной БД. Во вкладке "Помощь" содержится информация о модуле подсистемы "БД":

- *Модуль* – идентификатор модуля;
- *Имя* – имя модуля;
- *Тип* – тип модуля, идентификатор подсистемы, к которой модуль принадлежит;
- *Источник* – разделяемая библиотека - источник данного модуля;
- *Версия* – версия модуля;
- *Автор* – автор модуля;
- *Описание* – краткое описание модуля;
- *Лицензия* – лицензионное соглашение распространения модуля.

Каждая БД содержит собственную страницу конфигурации с вкладками "База данных" и "Таблицы". Кроме основных операций можно выполнять копирование содержимого БД стандартной функцией копирования объектов в конфигураторе. Операция копирования содержимого БД подразумевает копирование исходной БД в БД назначения, при этом содержимое БД назначения не очищается перед операцией копирования.

Копирование содержимого БД производится при условии включения обоих БД, иначе будет выполняться простое копирование объекта БД.

Вид вкладки "База данных" показан на рисунке 9.

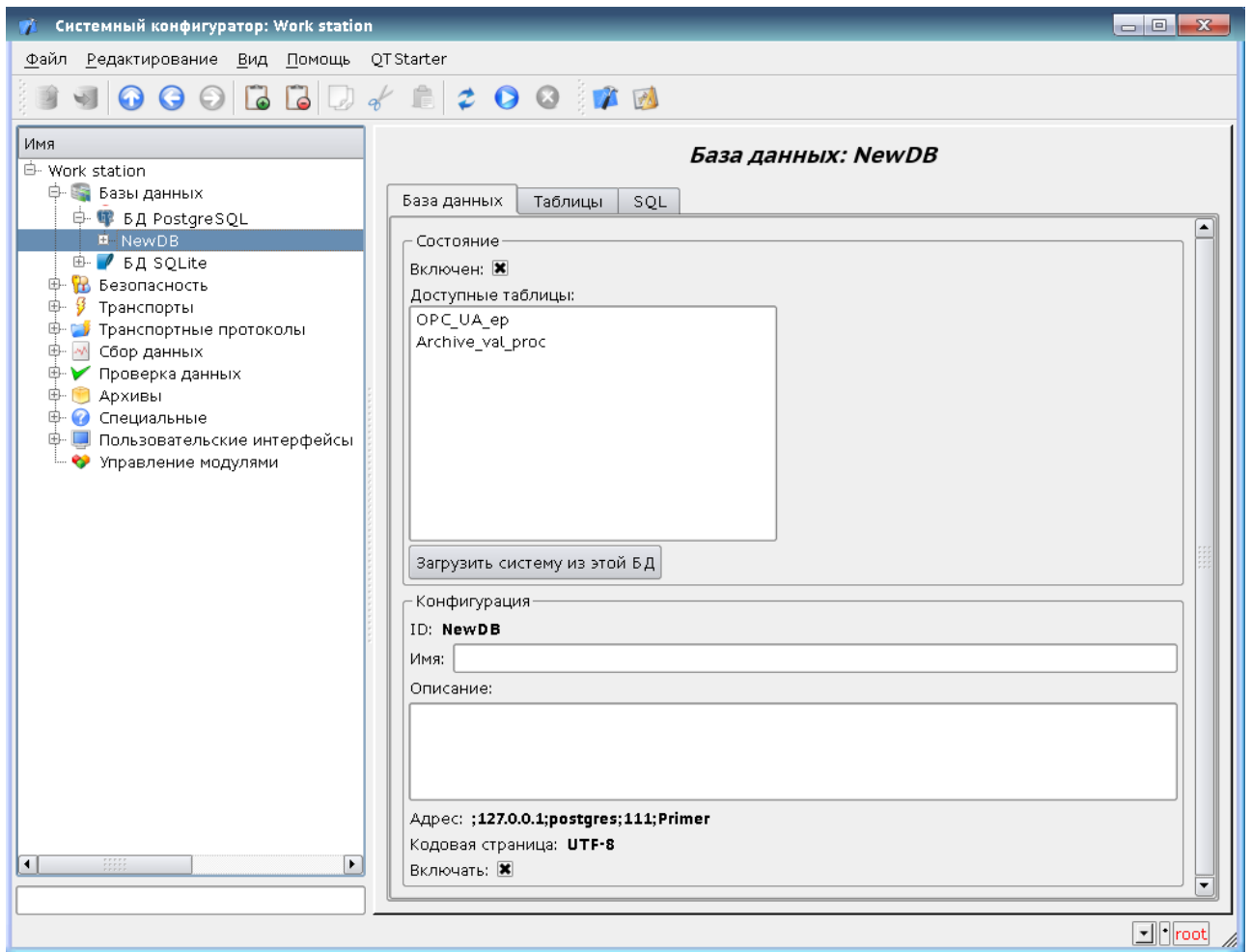


Рисунок 9

Вкладка "База данных" содержит основные настройки БД, заключенные в двух разделах.

Раздел "Состояние" - содержит свойства, характеризующие состояние БД:

- *Включен* - состояние БД "Включен";

- *Доступные таблицы* - перечень таблиц, которые содержит БД. Контекстным меню данного свойства предоставляется возможность физического удаления таблиц из БД;

- *Загрузить систему из БД* - команда для выполнения загрузки из данной БД. Может использоваться при переносе данных в БД между станциями. Например, можно сохранить участок одной станции в экспортную БД, физически перенести БД на другую станцию, подключить её в этой подсистеме и вызвать данную команду.

Раздел "Конфигурация" - непосредственно содержит поля конфигурации:

- *ID* - содержит информацию об идентификаторе БД;

- *Имя* - указывает имя БД;

- *Описание* - краткое описание БД и её назначения;

- *Адрес* - адрес БД в специфичном для типа БД (модуля) формате. Описание формата записи адреса БД, как правило, доступно во всплывающей подсказке этого поля. Форматы адресов для используемых БД приведены в таблице 2.

- *Кодовая страница* - указывает на кодовую страницу, в которой хранятся и предоставляются текстовые значения БД. Значение кодовой страницы БД в связке с внутренней кодировкой станции используется для прозрачного перекодирования текстовых сообщений при обмене между станцией и БД;

- *Включать* - указывает на состояние "Включен", в которое следует перевести БД при загрузке.

Таблица 2

Тип БД	Формат адреса
PostgreSQL	<p>[<host>;<hostaddr>;<user>;<pass>;<bd>;<port>;<connect_timeout>]. Где:</p> <p>host - Имя хоста для подключения. Если начинается с косой черты, оно указывает Unix-domain соединение вместо TCP/IP соединения, значение - это имя каталога, в котором хранится файл сокета.</p> <p>hostaddr - числовой IP адрес хоста для подключения, на котором работает сервер БД PostgreSQL;</p> <p>user - имя пользователя БД;</p> <p>pass - пароль пользователя для доступа к БД;</p> <p>bd - имя БД;</p> <p>port - порт, который слушает сервер БД (по умолчанию 5432);</p> <p>connect_timeout - таймаут соединения.</p> <p>В случае локального доступа к БД в пределах одного хоста строка адреса может выглядеть следующим образом: [;;user1;123456;SCADA;;10]</p> <p>В случае удалённого доступа к БД нужно использовать адрес хоста и порт сервера БД. Например: [server.nm.org;;user1;123456;SCADA;;10]</p>
SQLite	<p>[<FileDBPath>]. Где:</p> <p>FileDBPath - полный путь к файлу БД (./DATA/MainSt.db).</p> <p>Используйте пустой путь для создания временной базы данных на диске.</p> <p>Используйте ":memory:" для создания временной базы данных в памяти.</p>

Вкладка "Таблицы" содержит список открытых таблиц. Наличие открытых таблиц говорит о том, что программа сейчас работает с таблицами, или таблицы открыты пользователем для изучения их содержимого. После завершения работы с таблицами программа их закрывает и вкладка «Таблицы» становится пустой.

В контекстном меню перечня открытых таблиц можно открыть таблицу для просмотра и редактирования (команда "Перейти"), удалить выбранную таблицу (команда "Удалить").

Страница просмотра содержимого таблицы содержит только одну вкладку "Таблица". Вид ее показан на рисунке 10.

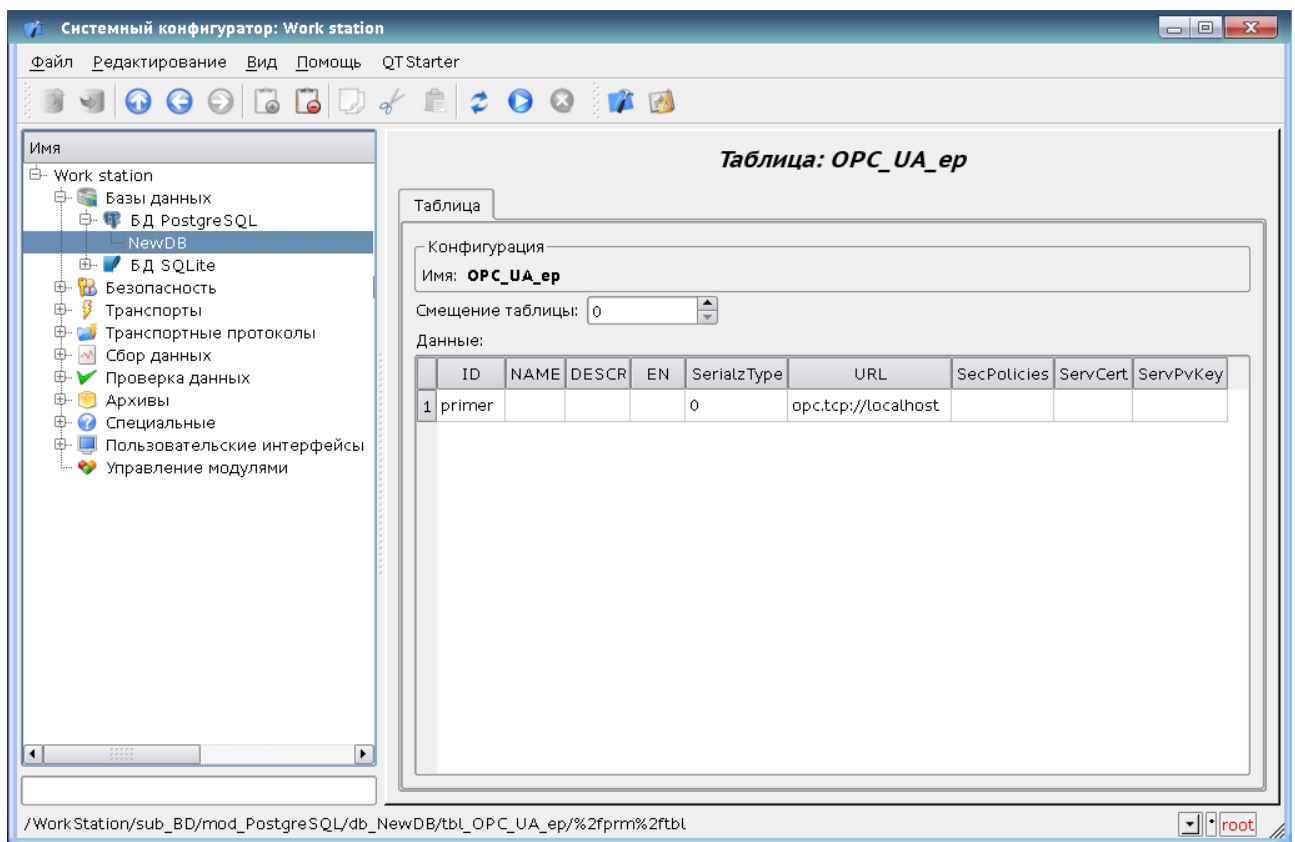


Рисунок 10

Вкладка "Таблица" содержит поле имени таблицы и таблицу с содержимым. Таблице содержимого предоставляются функции:

- редактирование содержимого ячеек таблицы;
- добавление записи (строки);
- удаление записи (строки).

Для редактирования ячейки необходимо осуществить на ней двойной щелчок мышью (рисунок 11). После этого можно ввести новое значение в ячейку (рисунок 12).

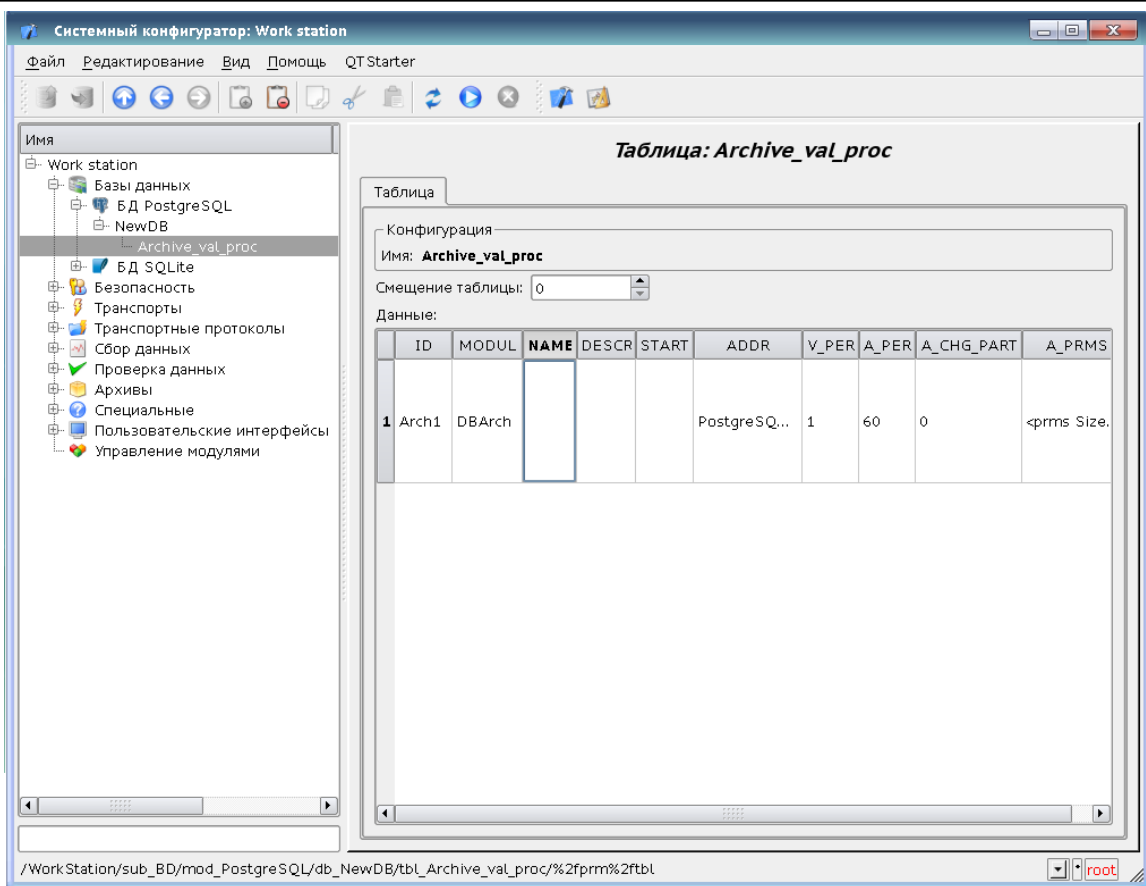


Рисунок 11

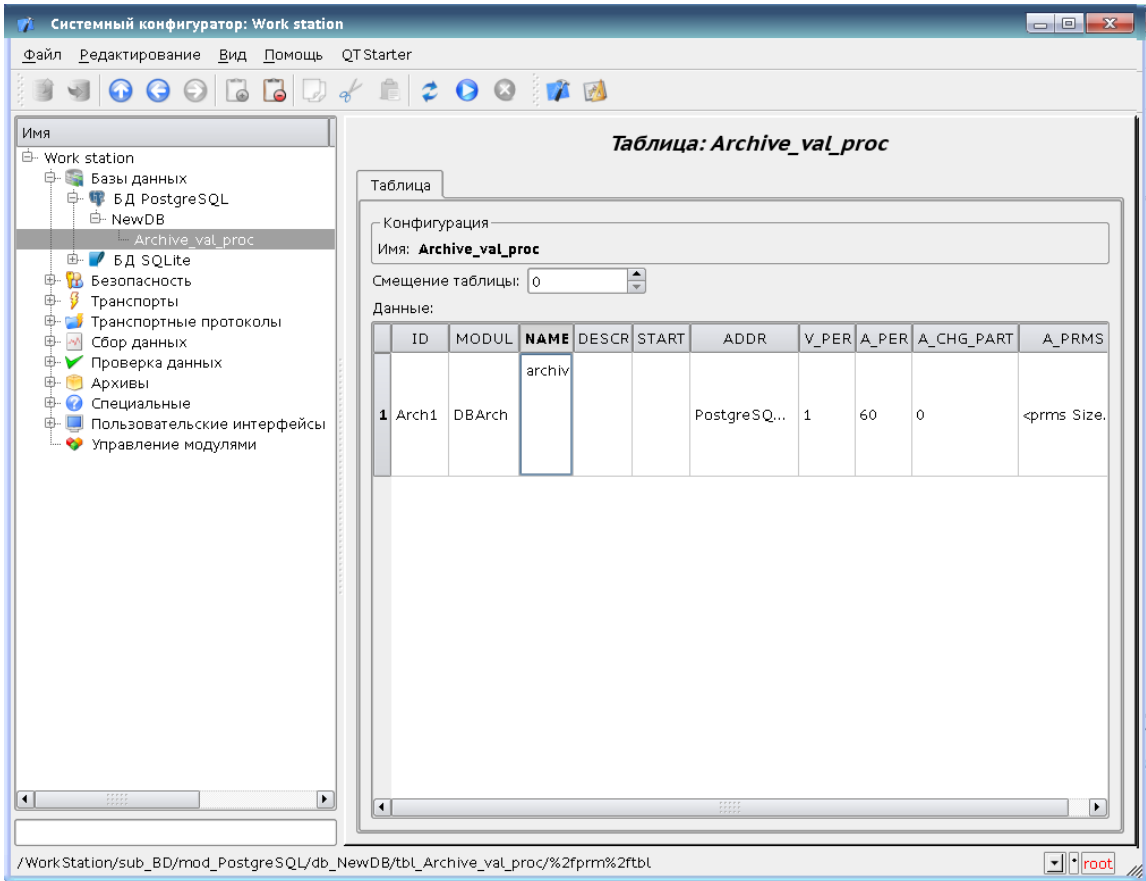


Рисунок 12

После завершения редактирования ячейки необходимо нажать клавишу «Enter» на клавиатуре, либо щелкнуть мышью по любой другой ячейке. В результате в ячейке сохранится введённое значение (рисунок 13). Если во время редактирования нажать клавишу «Esc» на клавиатуре, то вновь введенное значение не будет сохранено в ячейке.

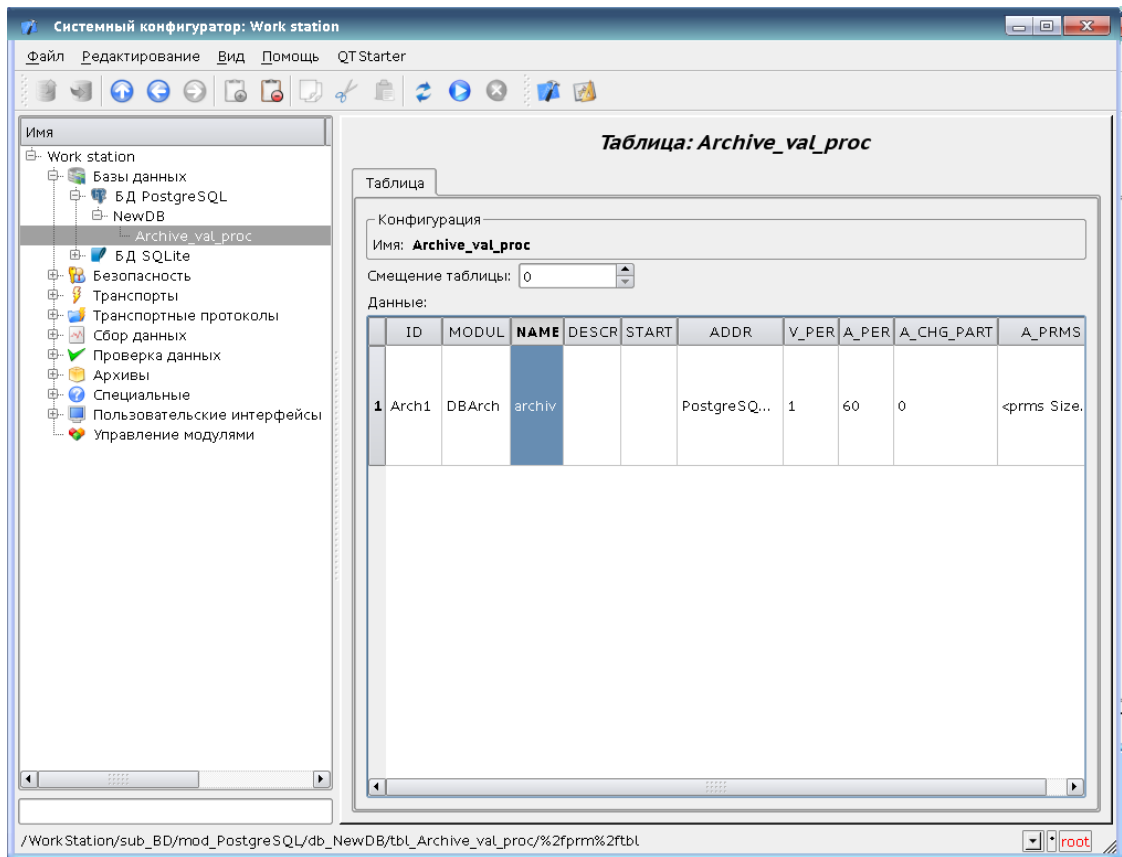


Рисунок 13

Для удаления записи (строки) нужно привести курсор мыши на нужную строку, нажать правую клавишу мыши и в появившемся списке выбрать пункт «Удалить запись» (рисунок 14).

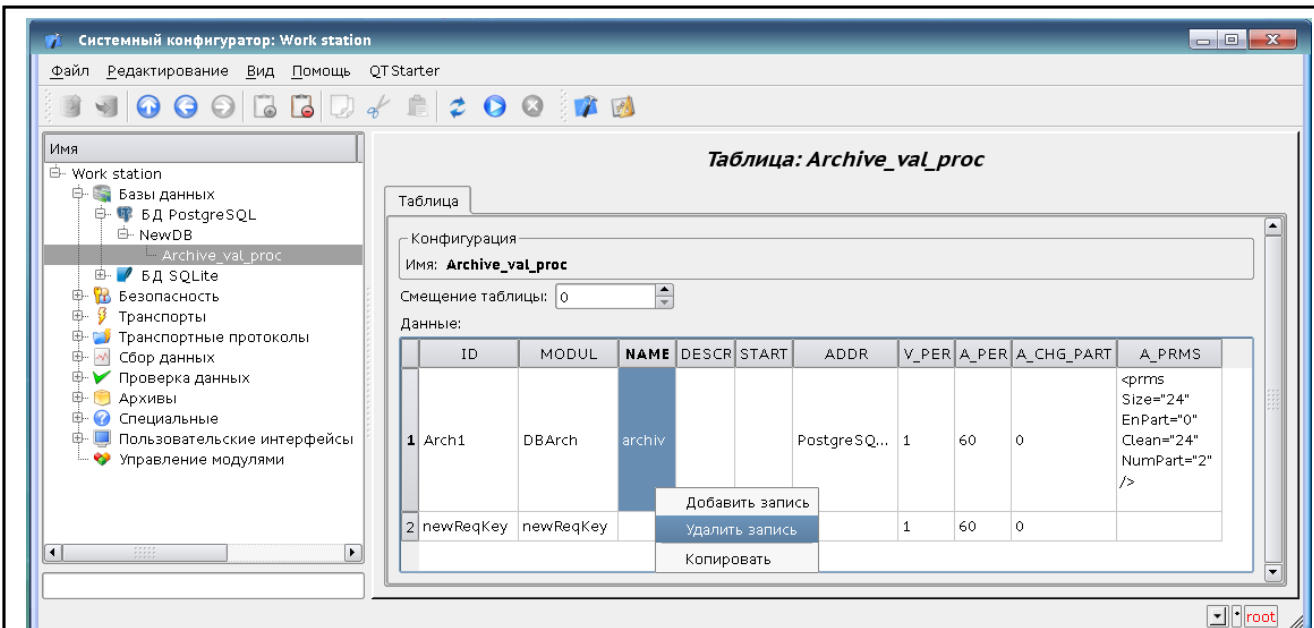


Рисунок 14

В результате данная строка будет удалена из таблицы, а остальные строки будут автоматически смещены.

Для добавления записи (строки) нужно щелкнуть правой клавишей мыши в любом месте таблицы и в появившемся списке выбрать пункт «Добавить запись».

В результате новая строка (запись) будет добавлена в конец таблицы (рисунок 15).

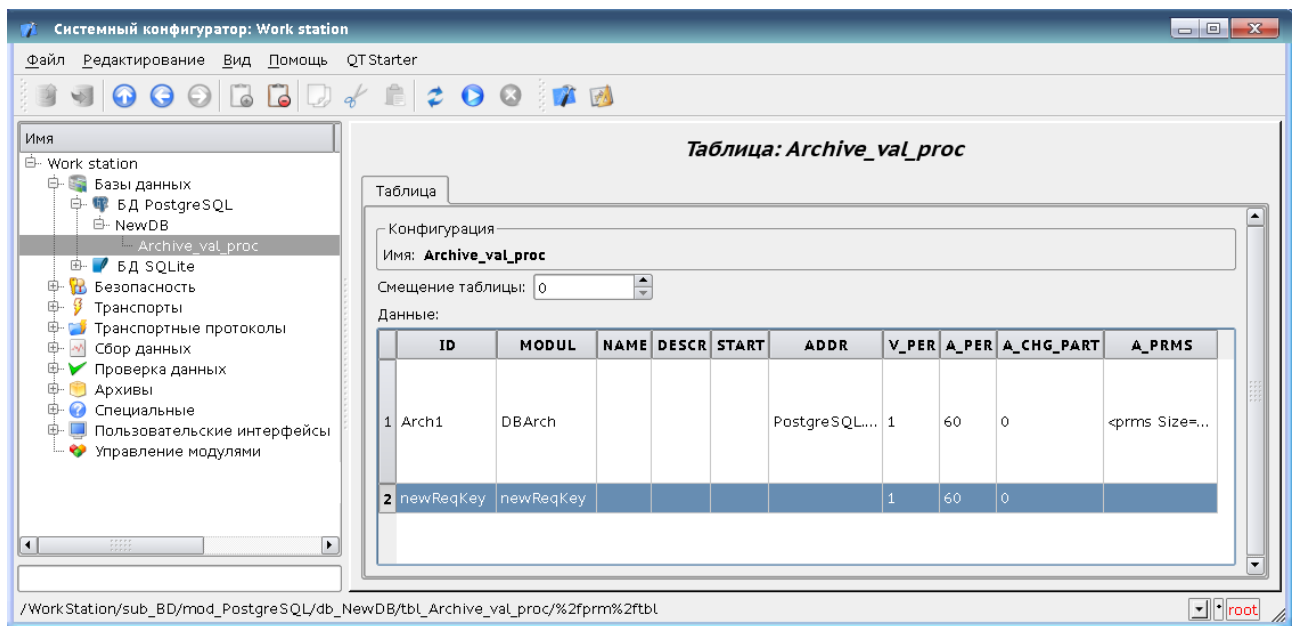


Рисунок 15

3.4 Подсистема "Безопасность"

3.4.1 Общие сведения

Для разграничения доступа в СКАДА предусмотрена подсистема "Безопасность". Основными функциями подсистемы "Безопасность" являются:

- хранение учётных записей пользователей и групп пользователей;
- аутентификация пользователей;
- проверка прав доступа пользователя к тому или иному ресурсу.

Подсистема "Безопасность" не является модульной.

3.4.2 Конфигурирование подсистемы «Безопасность»

Для конфигурации подсистемы предусмотрена корневая страница подсистемы "Безопасность", содержащая вкладки "Пользователи и группы пользователей" и "Помощь". Вид вкладки "Пользователи и группы пользователей" показан на рисунке 16.

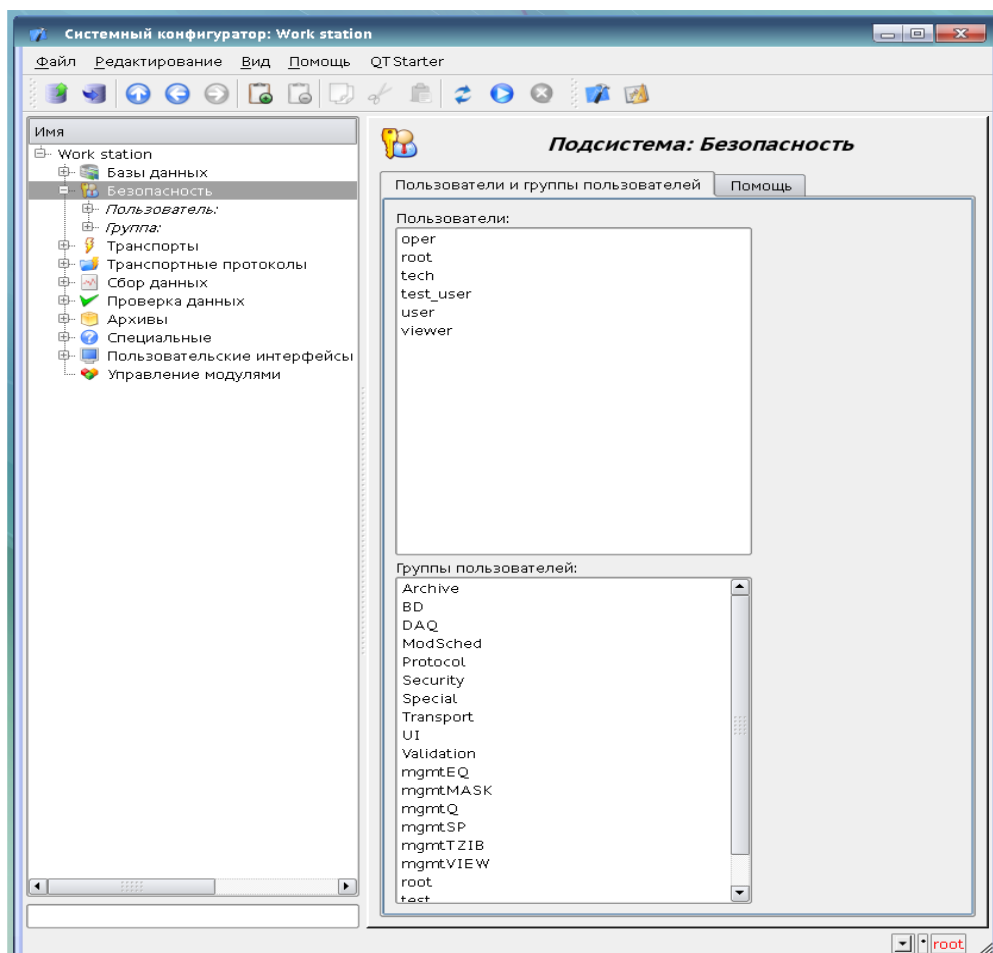


Рисунок 16

Вкладка "Пользователи и группы пользователей" содержит списки пользователей и групп пользователей. Пользователь в группе "Security" и с правами привилегированного пользователя может добавить, удалить пользователя или группу пользователей. Все остальные пользователи могут перейти к странице пользователя или группы пользователя.

Вкладка "Помощь" содержит краткую помощь для данной страницы.

Для конфигурации учетных данных пользователя предоставляется страница, содержащая только вкладку "Пользователь". Ее вид показан на рисунке 17.

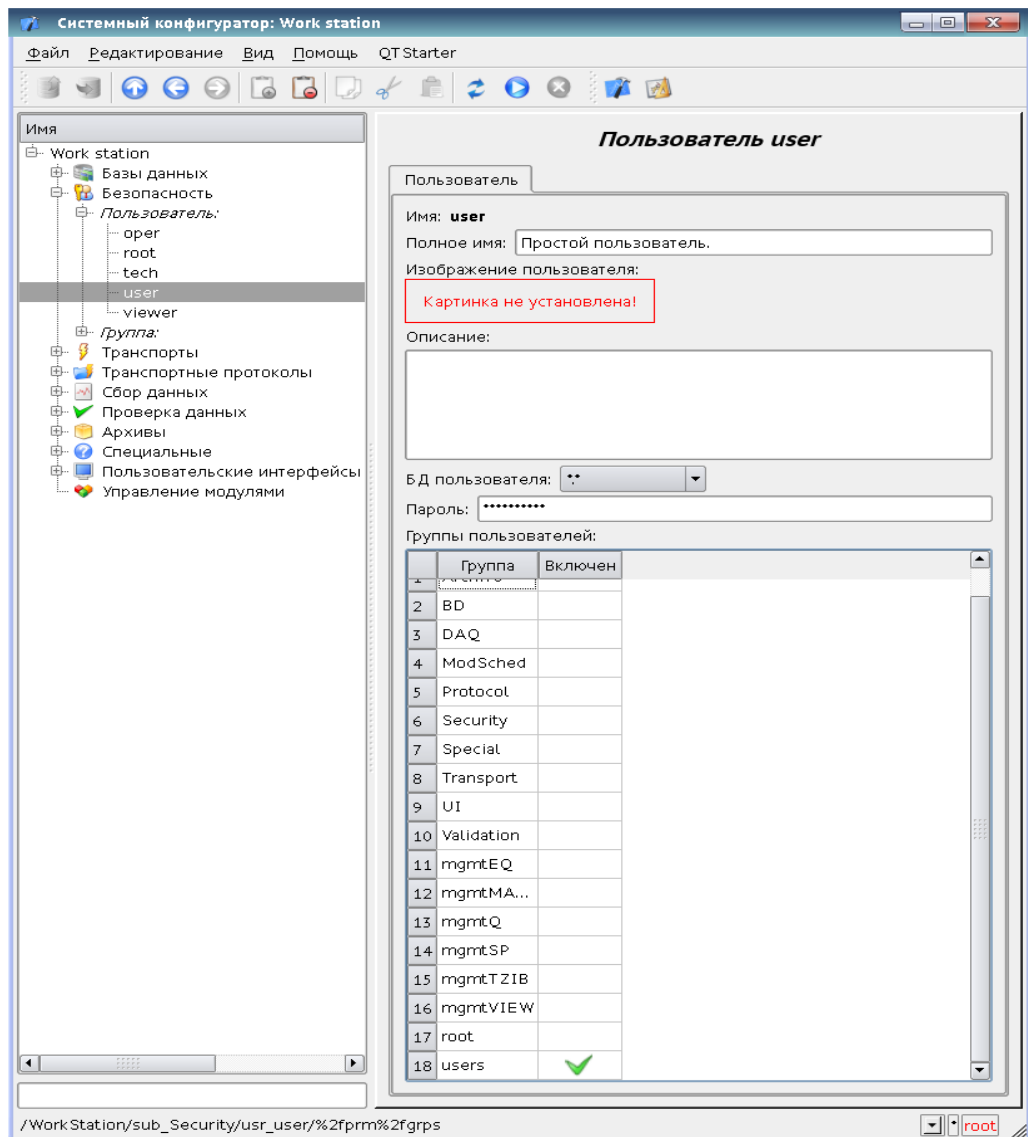


Рисунок 17

Вкладка содержит конфигурационные данные профиля пользователя, которые может изменять сам пользователь, пользователь в группе "Security" или привилегированный пользователь:

- *Имя* - информация об имени (идентификаторе) пользователя;
- *Полное имя* - указывает на полное имя пользователя;

- *Изображение пользователя* - указывает изображение пользователя. Изображение может быть загружено или выгружено.

- *БД пользователя* - адрес БД для хранения данных пользователя, выбирается из списка щелчком мыши;

- *Пароль* - поле для изменения пароля пользователя. Всегда отображает "*****";

- *Группы пользователей* - таблица с перечнем групп пользователей станции и признаком принадлежности пользователя к группам.

Чтобы добавить нового пользователя необходимо щелкнуть правой кнопкой мыши в области «Пользователи:» вкладки «Пользователи и группы пользователей». В появившемся меню выбрать пункт «Добавить» (рисунок 18).

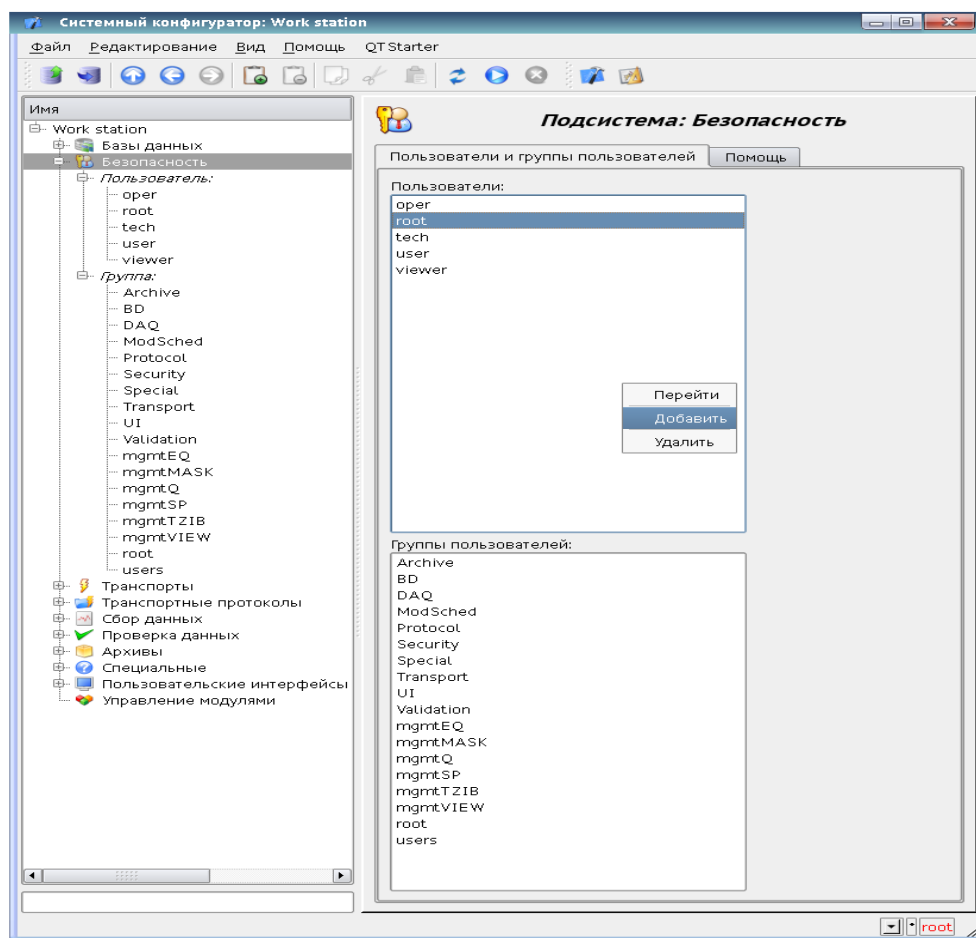


Рисунок 18

Появится окно «Добавление нового элемента», изображённое на рисунке 19. Здесь нужно ввести имя нового пользователя, например «test_user», и нажать кнопку «Ok».

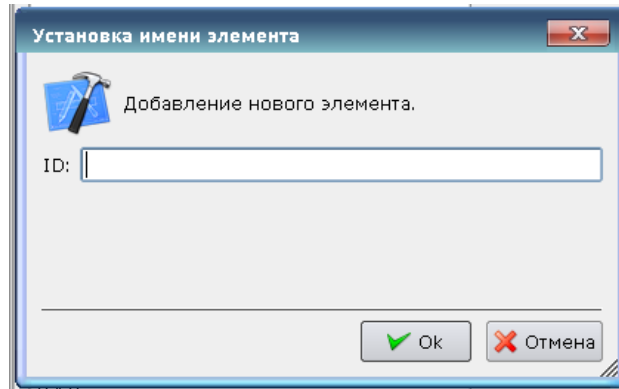


Рисунок 19

После этого новый пользователь «test_user» появится в списке пользователей, как показано на рисунке 20.

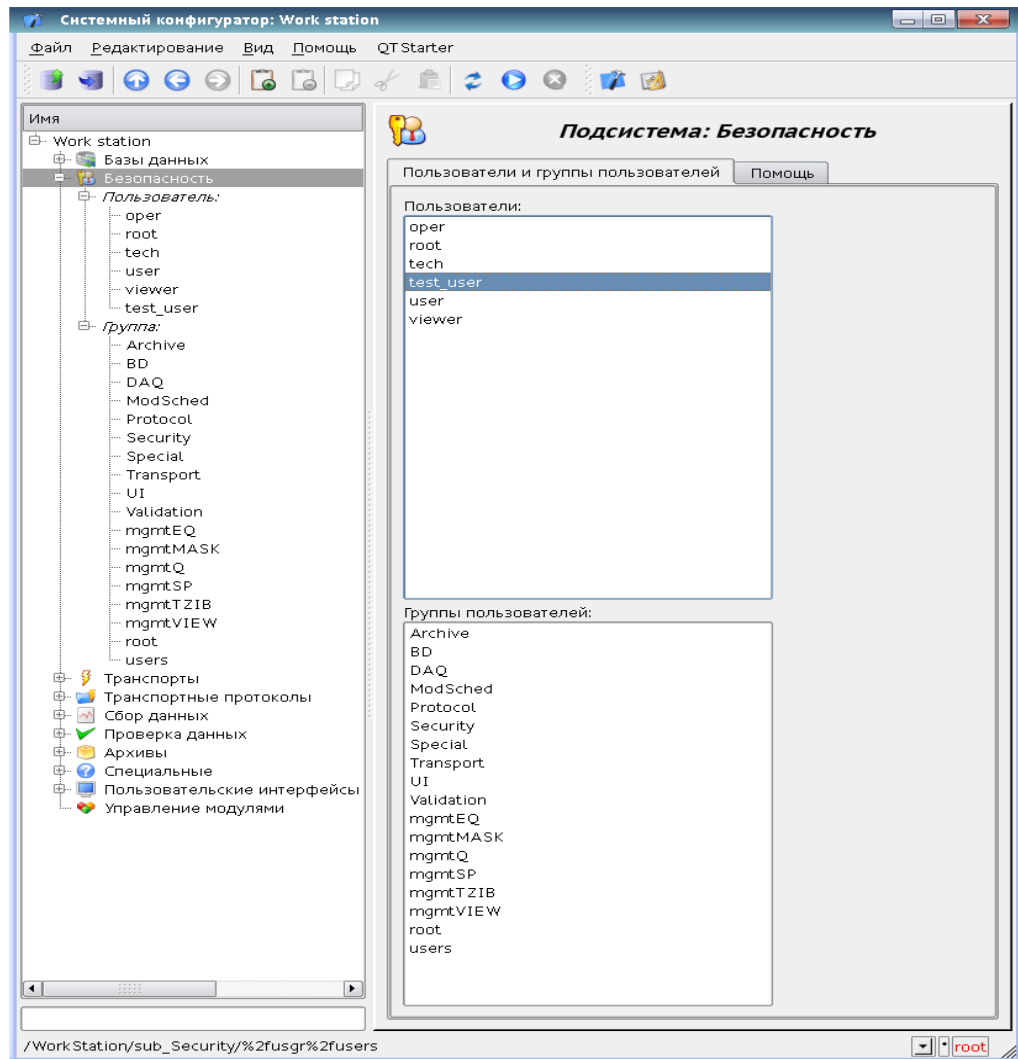


Рисунок 20

Для перехода к окну конфигурации учетных данных пользователя необходимо щелкнуть правой кнопкой мыши на новом пользователе «test_user» и в появившемся окне выбрать пункт «Перейти» (рисунок 21).

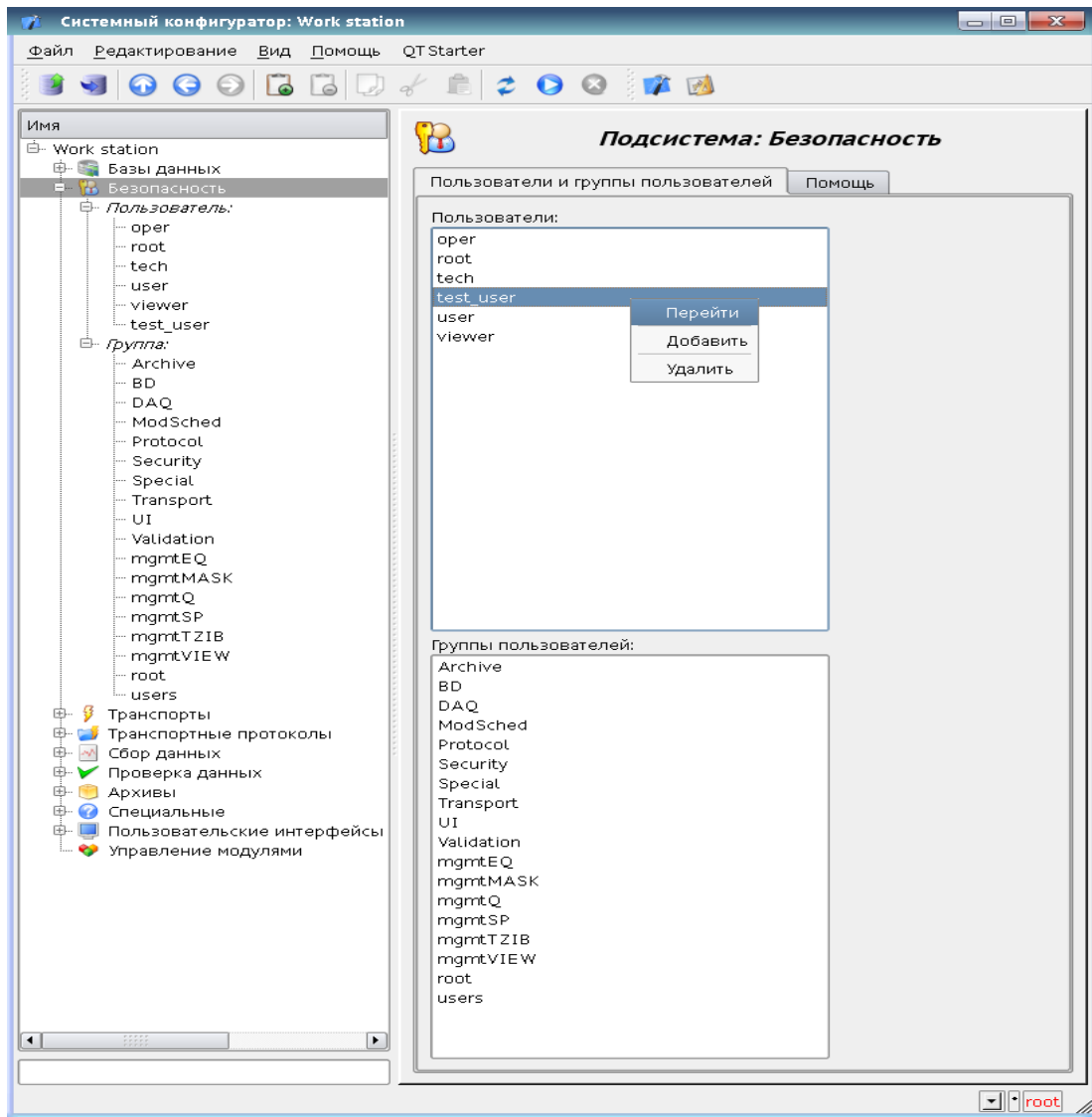



Рисунок 21

В появившемся окне производится настройка параметров нового пользователя. В поле «**Полное имя**» нужно ввести описание пользователя и нажать кнопку  напротив поля или клавишу «Enter» на клавиатуре.

Чтобы установить изображение пользователя нужно щелкнуть правой кнопкой мыши на поле «**Изображение пользователя:**» и нажать «Загрузить изображение» (рисунок 22).

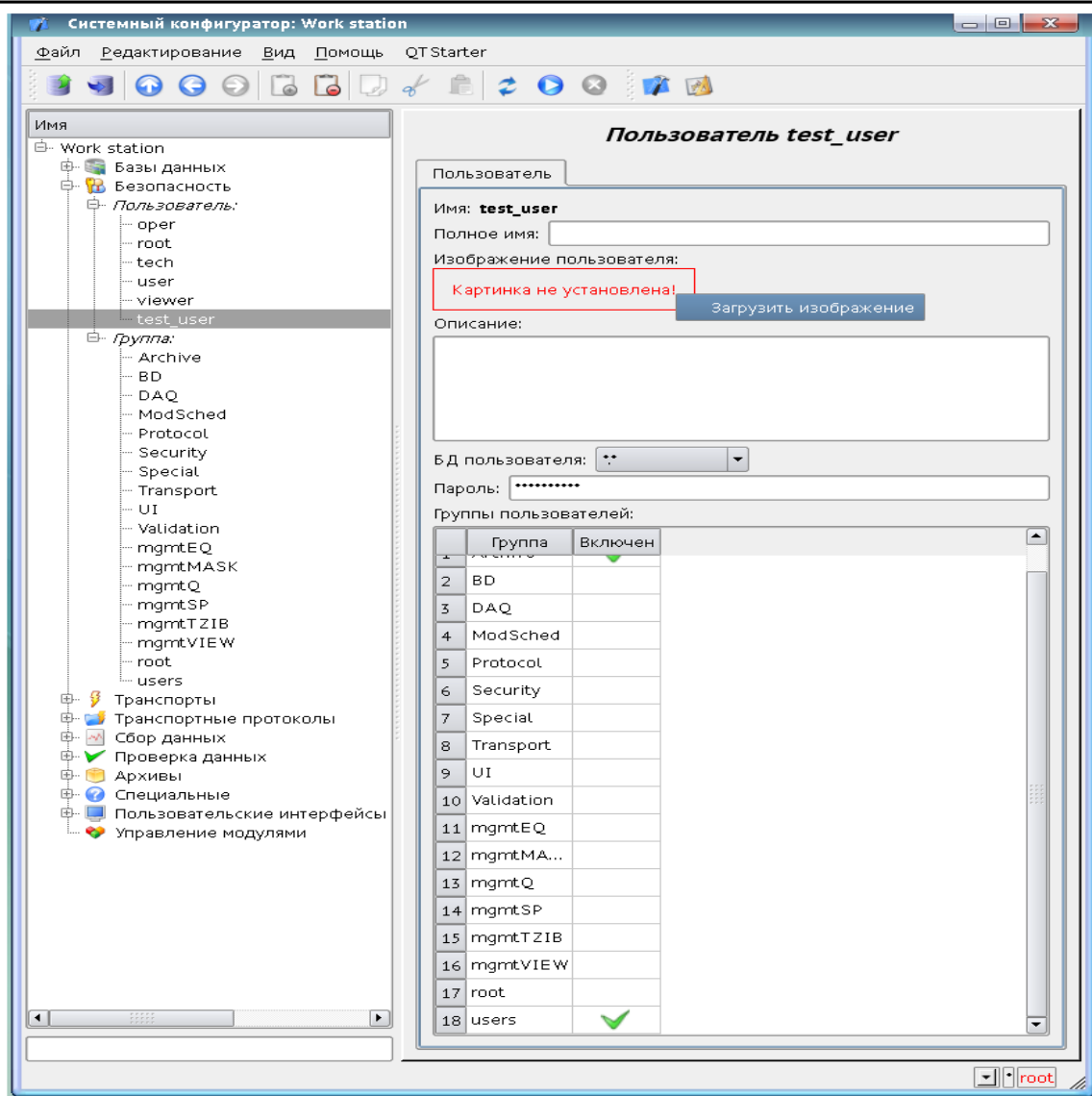


Рисунок 22

Для выбора БД, в которой будут храниться данные пользователя, необходимо щелкнуть левой кнопкой мыши по комбобоксу в поле «БД пользователя» и в появившемся списке выбрать нужную БД (рисунок 23).

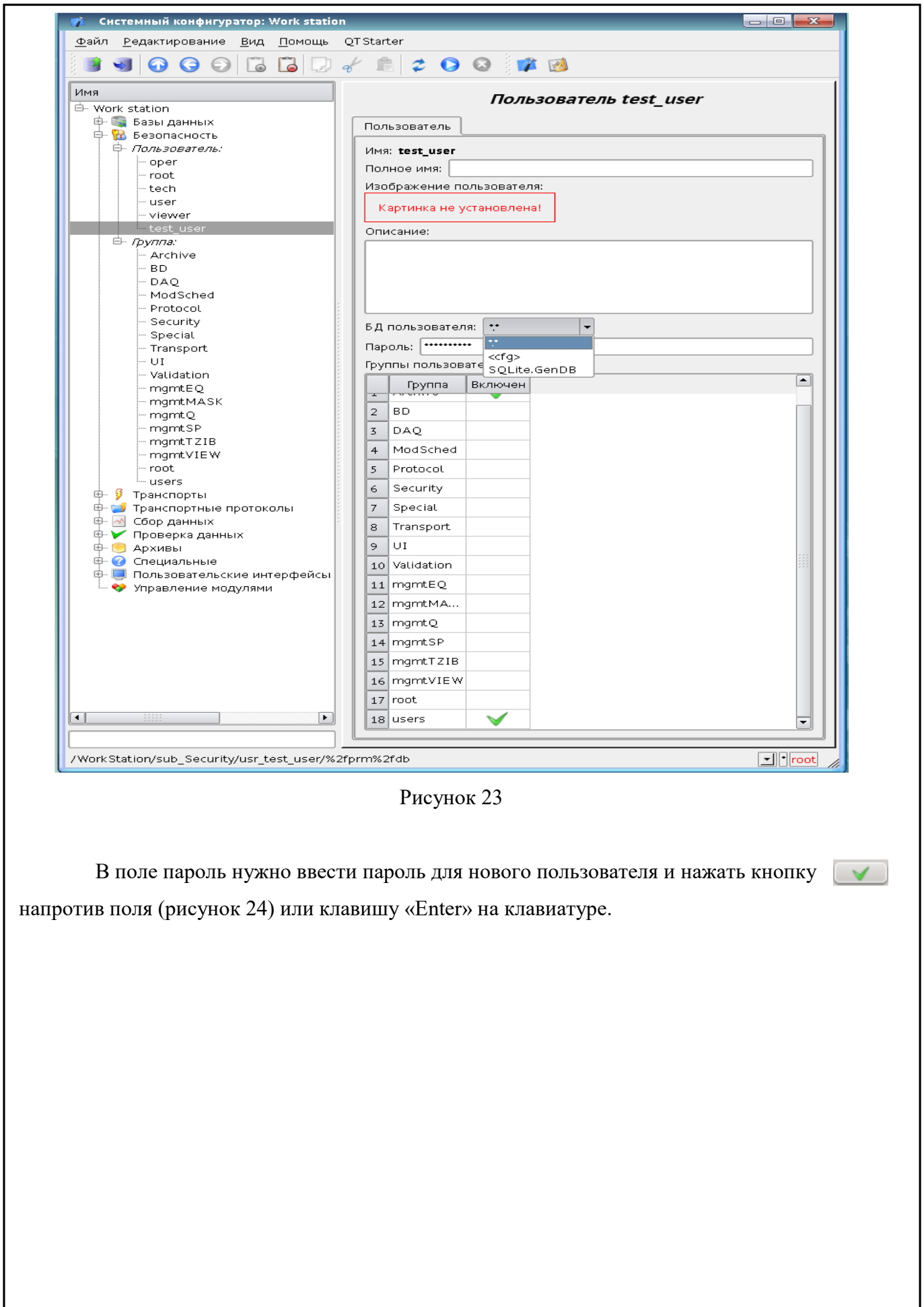



Рисунок 23

В поле пароль нужно ввести пароль для нового пользователя и нажать кнопку  напротив поля (рисунок 24) или клавишу «Enter» на клавиатуре.

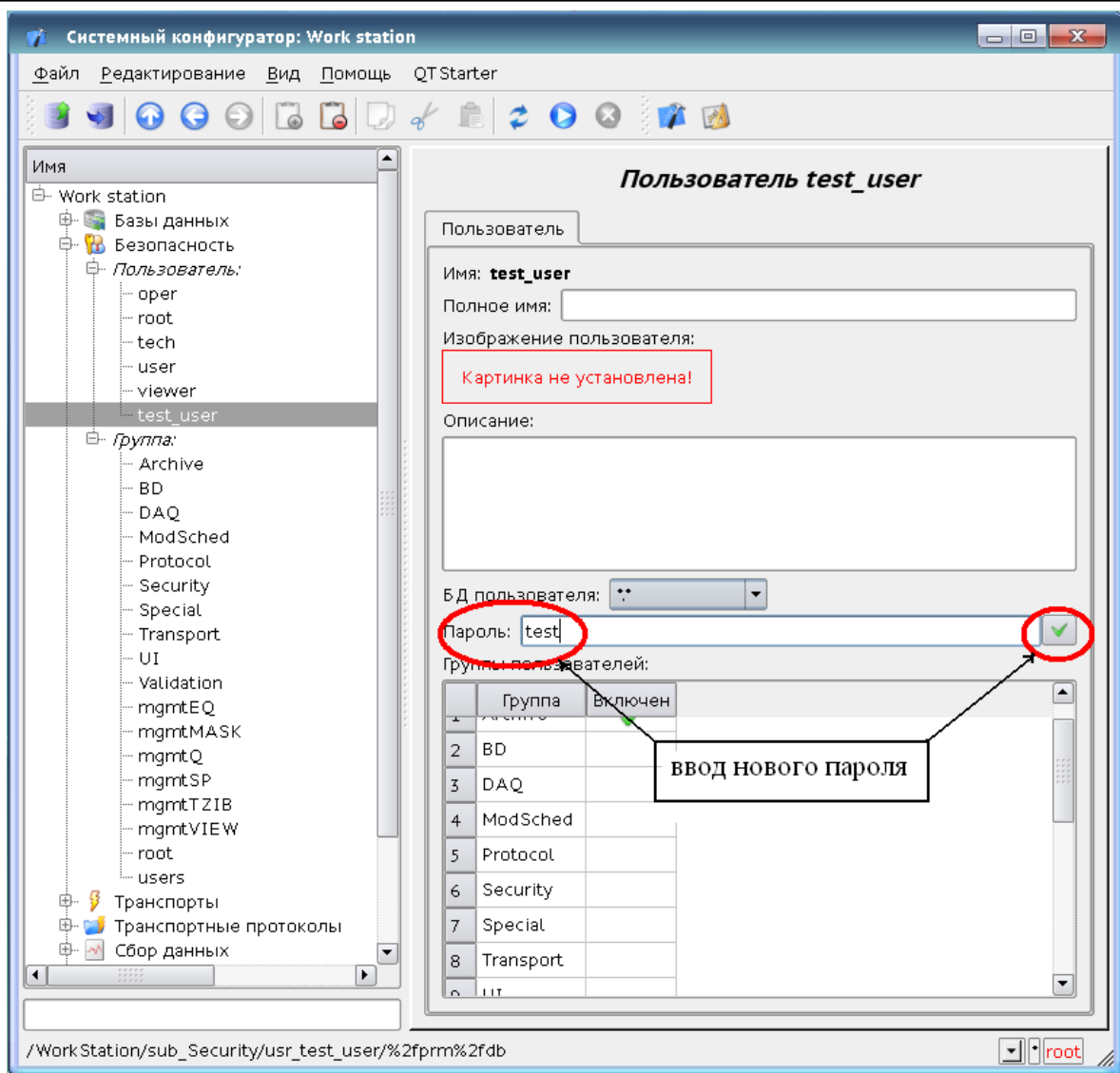


Рисунок 24

По умолчанию новый пользователь добавляется в группу «users». Чтобы добавить нового пользователя в другую группу используется поле «Группы пользователей». Например, для добавления в группу «Archive» необходимо произвести двойной щелчок левой кнопкой мыши на пустой ячейке «Включен» напротив «Archive». В ней появится комбобокс с надписью «False» (рисунок 25).

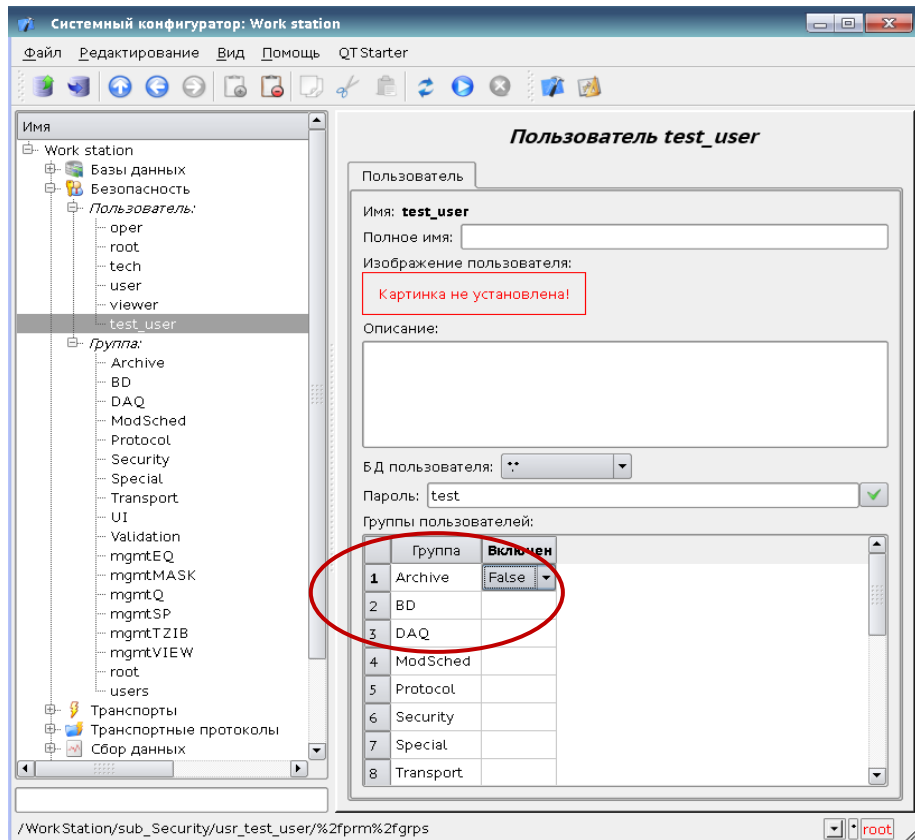


Рисунок 25

Далее нужно щелкнуть левой кнопкой мыши на этом комбобоксе и в появившемся списке выбрать «True» (рисунок 26).

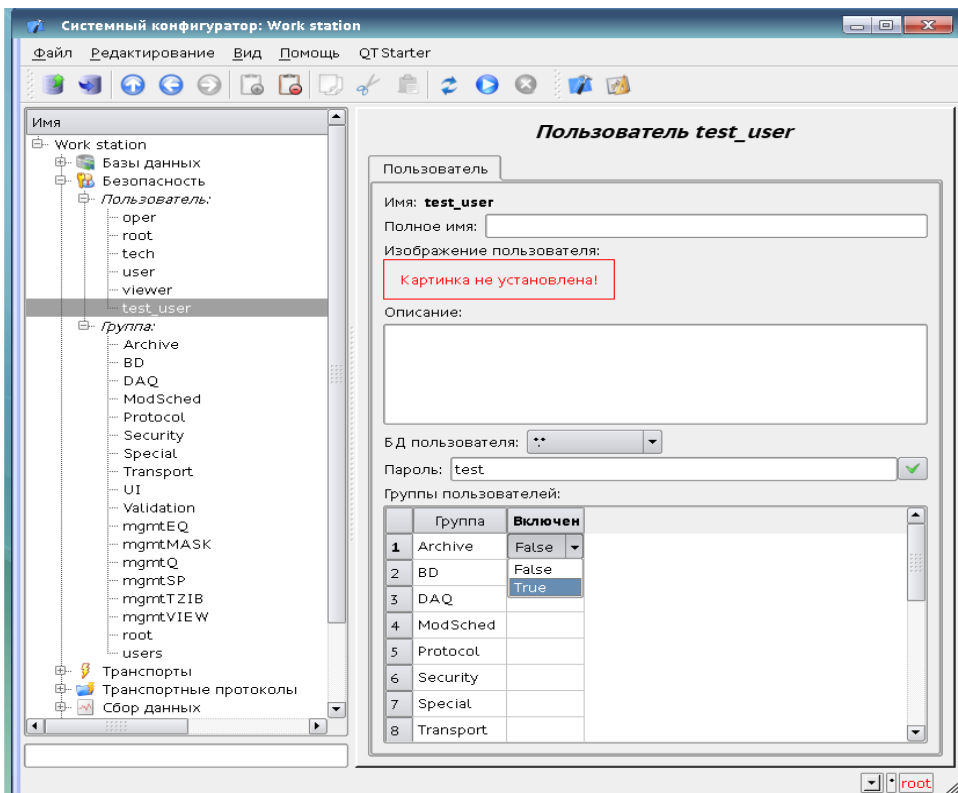


Рисунок 26

В результате пользователь «test_user» будет выбран в группу «Archive» (рисунок 27).

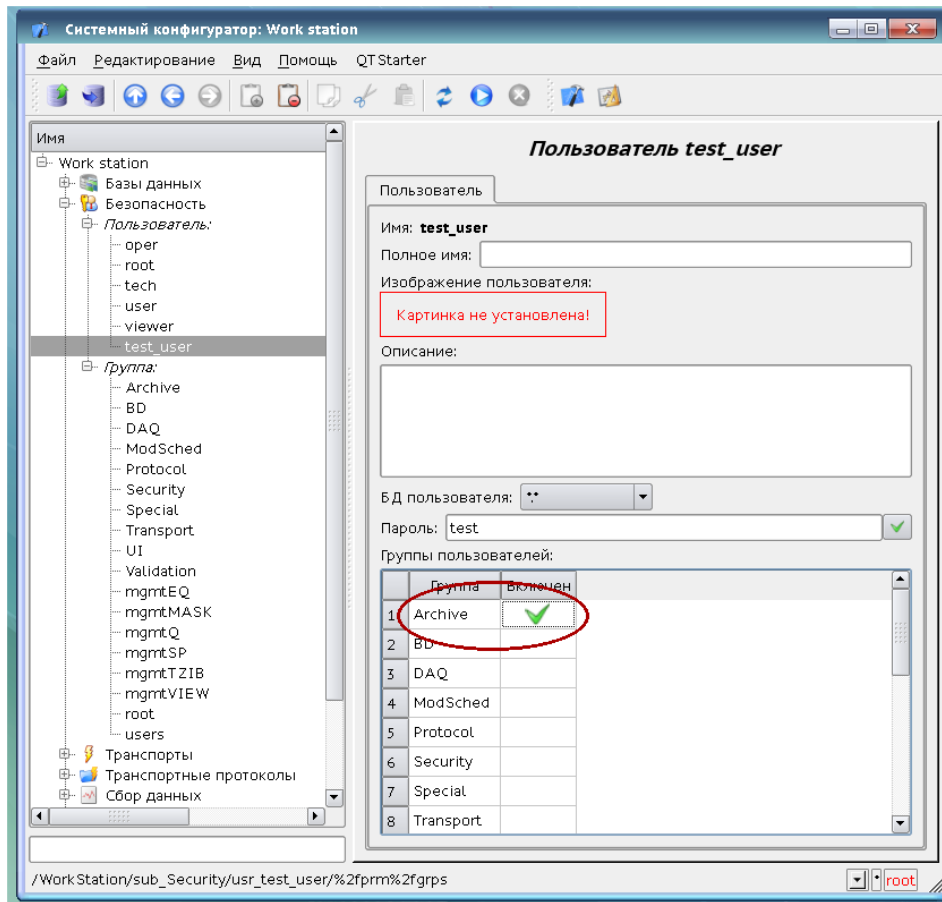


Рисунок 27

Для конфигурации группы пользователей предоставляется страница, содержащая вкладку "Группа". Ее вид показан на рисунке 28.

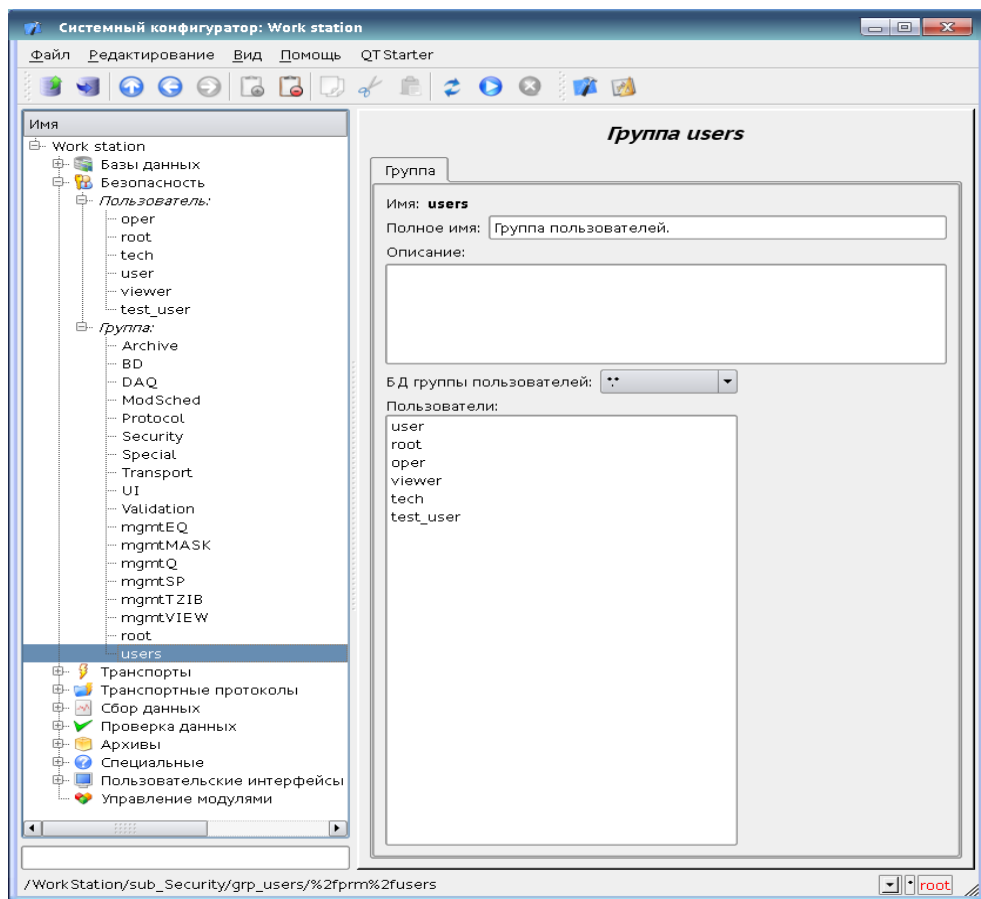


Рисунок 28

Вкладка содержит конфигурационные данные профиля группы пользователей, которые может изменять только привилегированный пользователь:

- *Имя* - информация об имени (идентификаторе) группы пользователей;
- *Полное имя* - указывает на полное имя группы пользователей;
- *БД группы пользователей* - адрес БД для хранения данных группы пользователей, выбирается из списка щелчком мыши;
- *Пользователи* - список пользователей, включенных в данную группу. С помощью контекстного меню списка можно добавить или удалить пользователя в группе.

Для добавления новой группы пользователей необходимо открыть вкладку «Пользователи и группы пользователей», щелкнуть правой кнопкой мыши на поле «Группы пользователей» и нажать «Добавить» (рисунок 29).

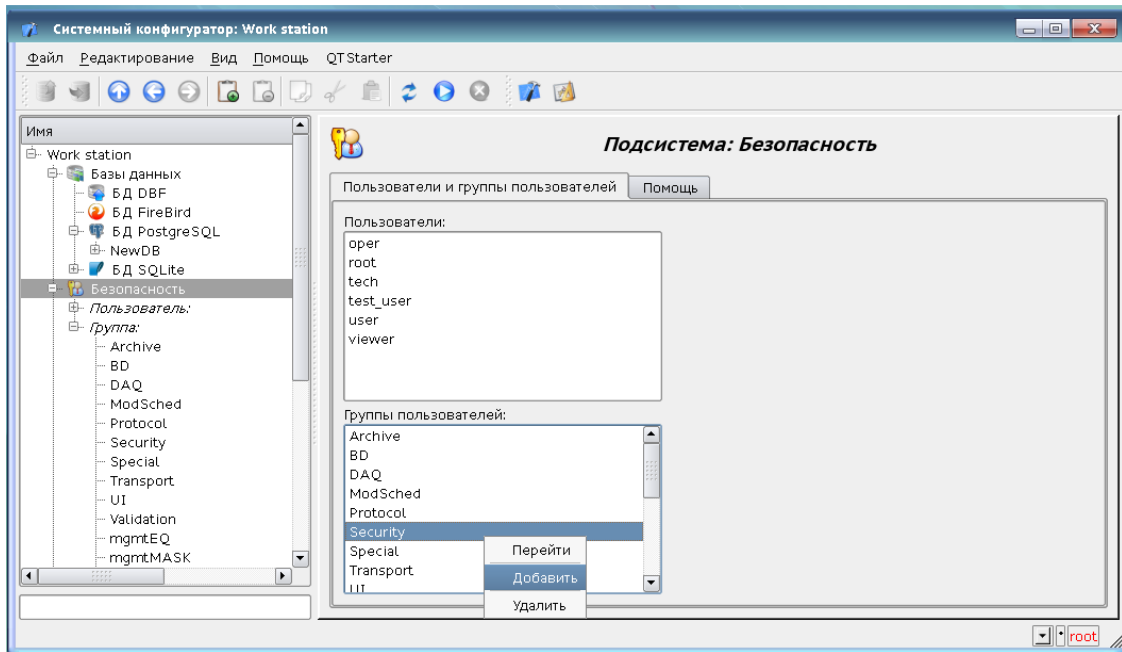


Рисунок 29

В результате будет отображено окно «Добавление нового элемента» (рисунок 30).

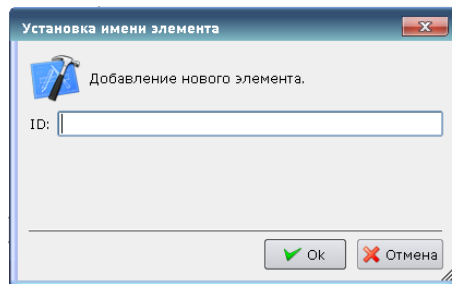


Рисунок 30

Здесь необходимо ввести имя новой группы, например «Test» (рисунок 31), а затем нажать кнопку «Ok», либо клавишу «Enter» на клавиатуре.

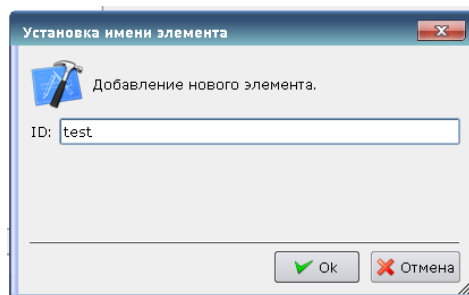


Рисунок 31

Теперь в списке групп пользователей будет отображаться имя новой созданной группы (рисунок 32).

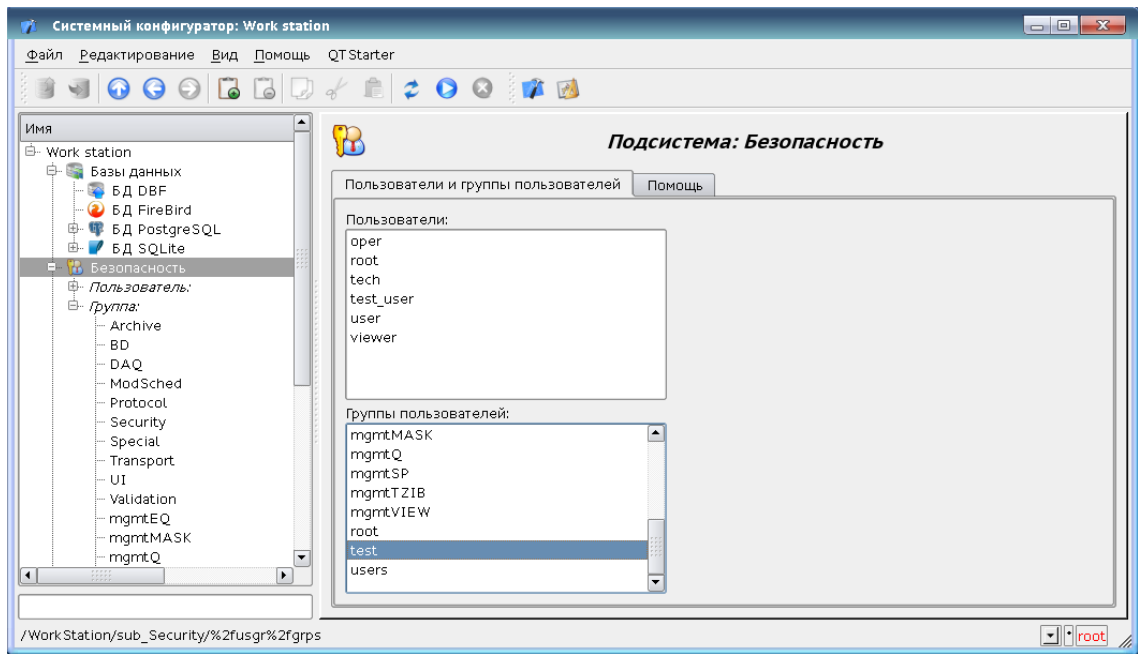


Рисунок 32

Для задания конфигурации новой группы нужно щелкнуть правой клавишей мыши по названию группы и в появившемся меню выбрать пункт «Перейти» (рисунок 33).

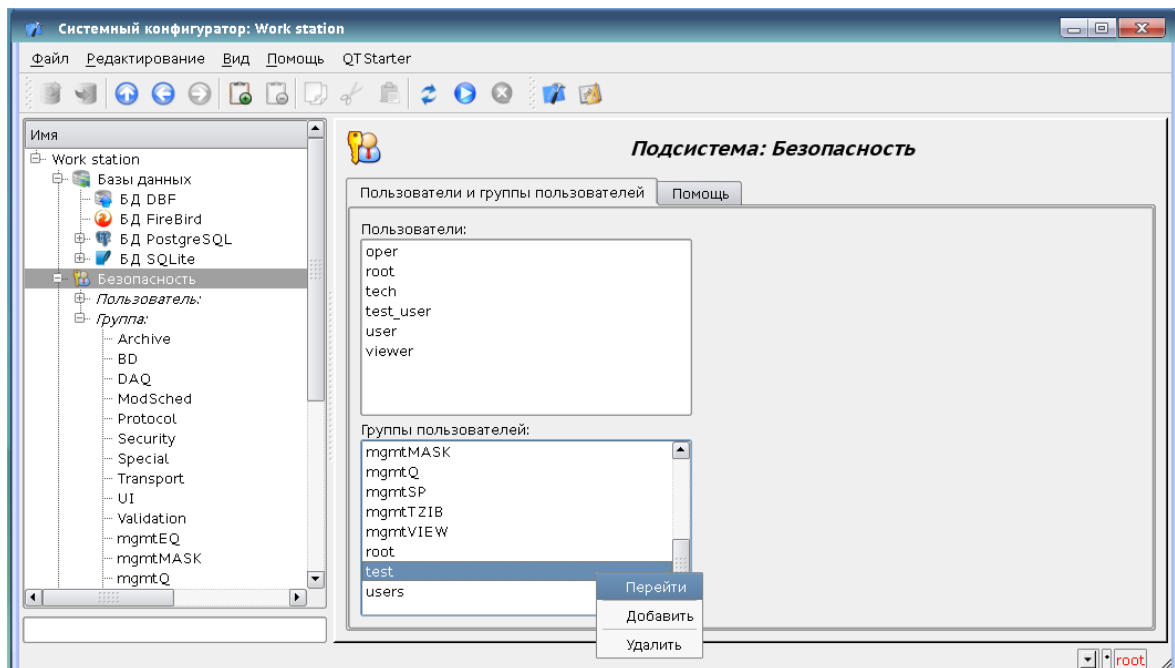



Рисунок 33

В поле «Полное имя» необходимо ввести описание новой группы, например «Группа тестовых пользователей» и нажать кнопку или  клавишу «Enter» на клавиатуре (рисунок 34).

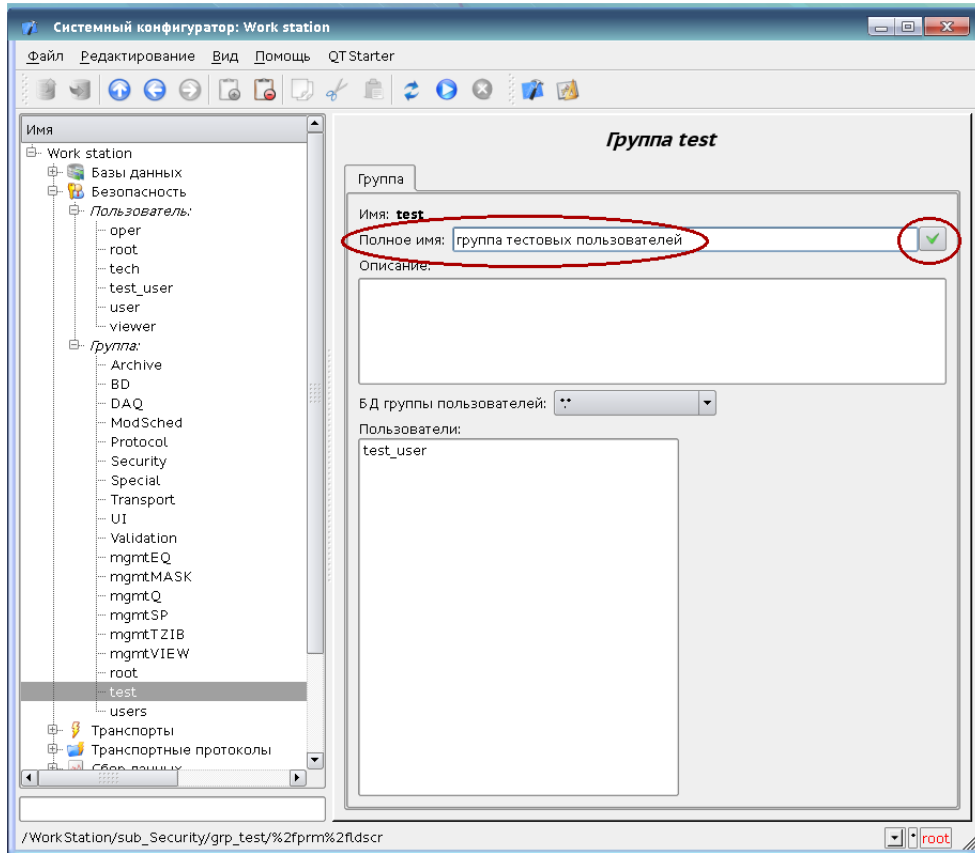



Рисунок 34

Для выбора БД, в которой будет храниться информация о новой группе, необходимо щелкнуть левой клавишей мыши по комбобоксу и в  появившемся списке выбрать нужную БД (рисунок 35).

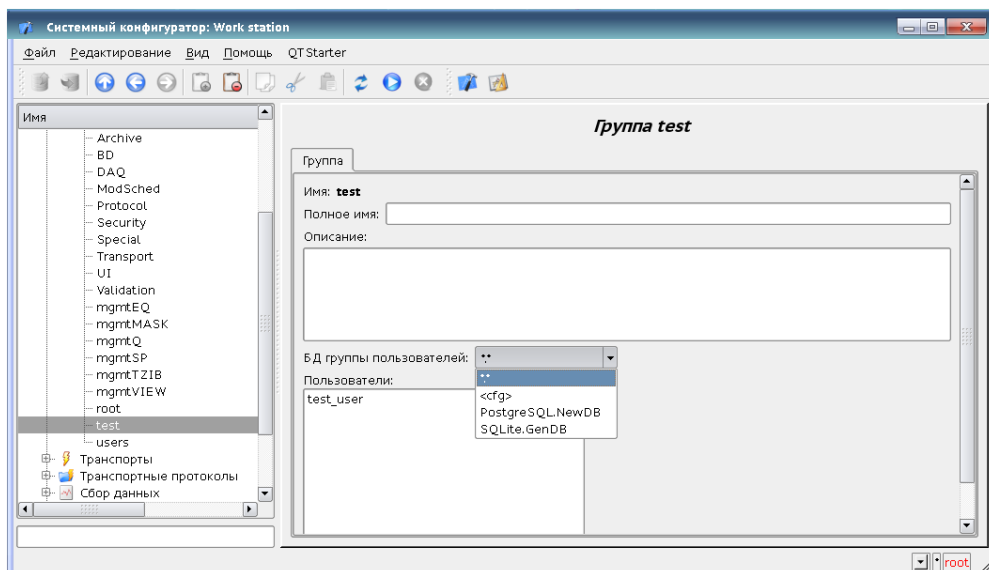


Рисунок 35

Для добавления пользователей в новую группу нужно щелкнуть правой клавишей мыши на поле «Пользователи» и нажать «Добавить» (рисунок 36).

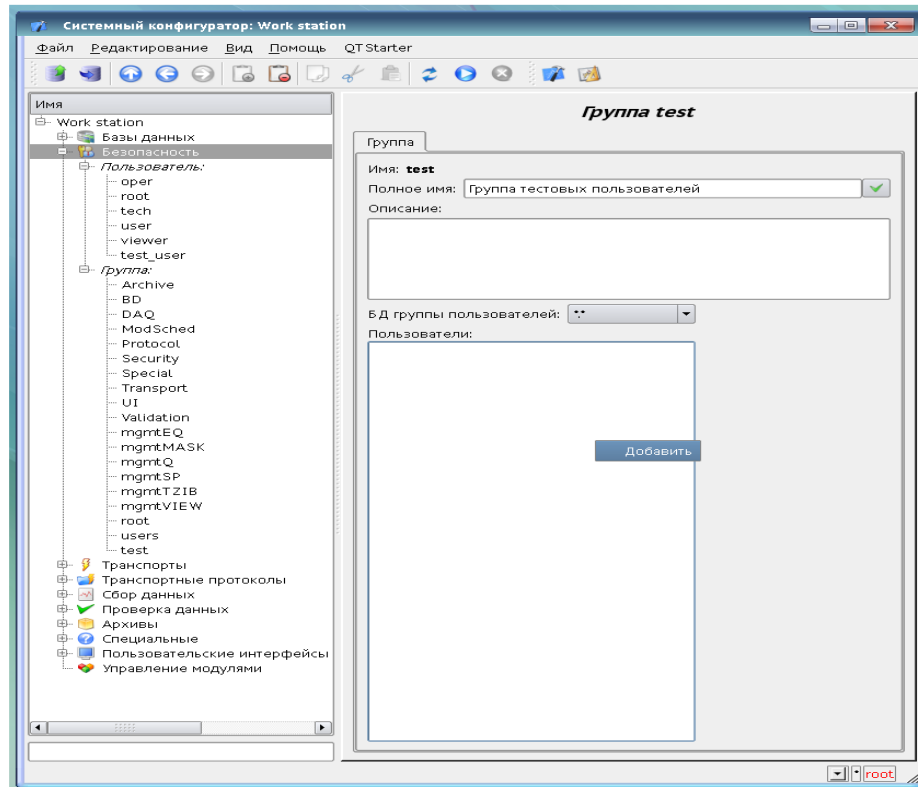


Рисунок 36

В появившемся окне «Добавление нового элемента» нужно ввести имя пользователя, например «test_user» (рисунок 37).

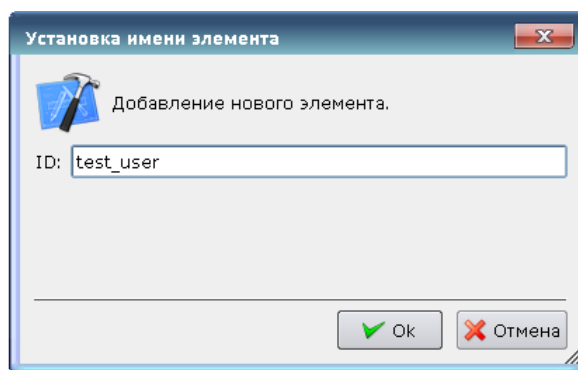


Рисунок 37

В результате пользователь «test_user» будет добавлен в группу «Test» (рисунок 38).

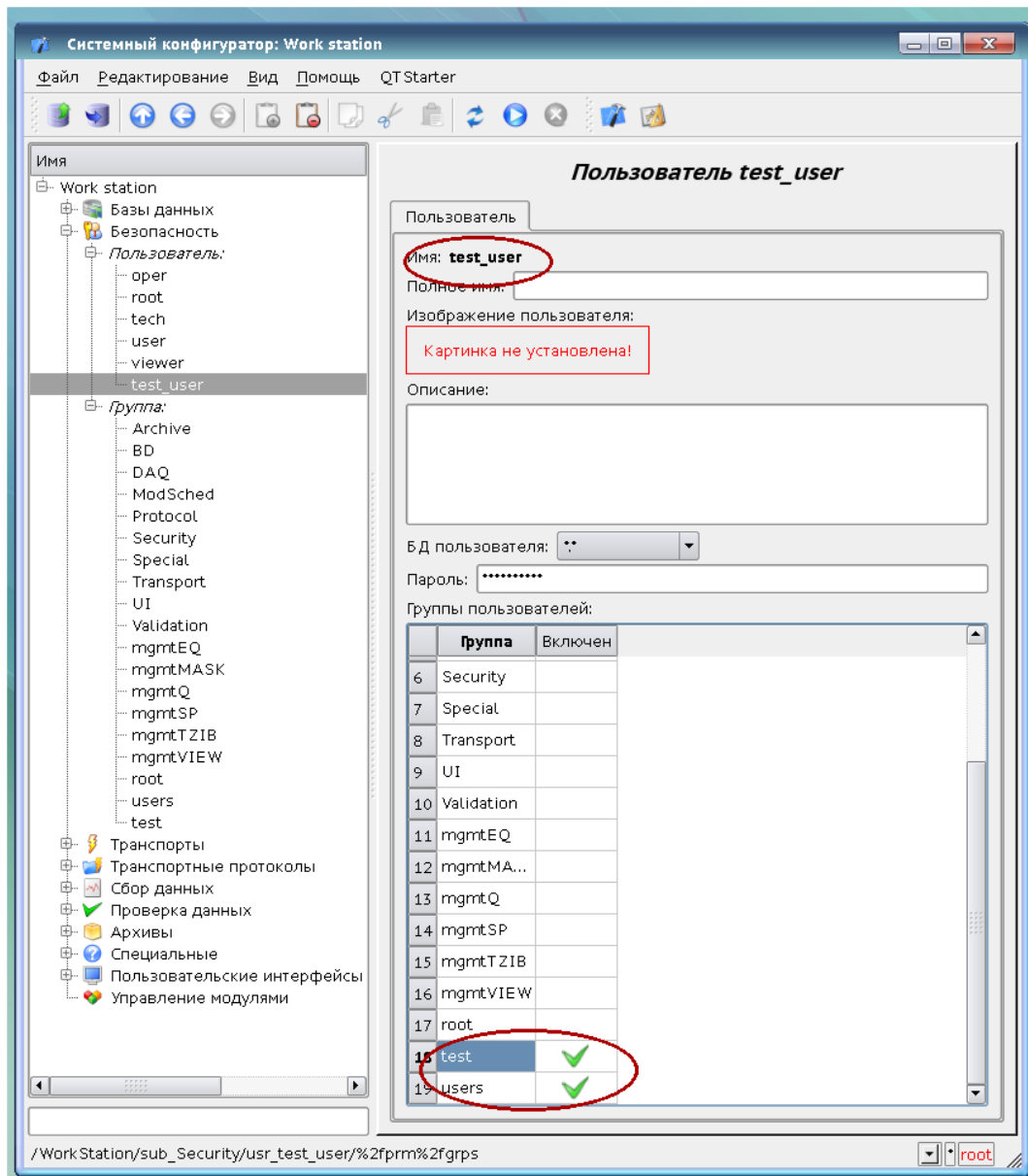


Рисунок 38

3.5 Подсистема "Транспорты"

3.5.1 Общие сведения

Подсистема «Транспорты» предназначена для обмена неструктурированными данными между СКАДА и внешними системами. В роли внешних систем могут выступать и удалённые СКАДА. Под неструктурированными данными понимается массив символов определённой длины. Модульным объектом, содержащимся в подсистеме «Транспорты», выступает тип транспорта. Тип транспорта определяет механизм передачи неструктурированных данных. Например, это могут быть:

- сокет (TCP/UDP/UNIX);
- каналы;
- разделяемая память.

Подсистема "Транспорты" включает поддержку входных и выходных транспортов. Входной транспорт предназначен для обслуживания внешних запросов и отправки ответов. Выходной транспорт, наоборот, предназначен для отправки сообщений и ожидания ответа.

В ПП «СКАДА А-СОФТ» реализованы следующие модули транспорта:

- последовательный интерфейс;
- сокет;
- SSL.

Тип транспорта *последовательный интерфейс* используется для обмена данными через последовательные интерфейсы типа RS232, RS485, GSM и др. В режиме модема модулем поддерживается смешанный режим работы. Смешанный режим подразумевает наличие входного транспорта, который ожидает внешних подключений, а также выходного транспорта на том же устройстве. Т.е. входной транспорт будет игнорировать все запросы при наличии установленного выходным транспортом соединения, в то же время выходной транспорт не будет осуществлять попыток установить соединение при наличии подключения к входному транспорту или соединения другого выходного транспорта, например, с другим номером телефона.

Модуль транспорта сокет предоставляет транспорт, основанный на сокетах. Поддерживаются входной и выходной транспорты, основанные на интернет сокетах: TCP, UDP и UNIX-сокет.

Сконфигурированный и запущенный входной транспорт открывает серверный сокет для ожидания соединения клиентов. В случае с UNIX сокетом создаётся файл UNIX сокета.

Сокеты TCP и UNIX являются многопоточными, т.е. при подключении клиента к сокетам данных типов создаётся клиентский сокет и новый поток, в котором производится обслуживание клиента. Серверный сокет в этот момент переходит к ожиданию запросов от нового клиента. Так достигается параллельное обслуживание клиентов. Каждый входной сокет обязательно связывается с одним из доступных транспортных протоколов, которому передаются входящие сообщения. В связке с транспортным протоколом поддерживается механизм объединения кусков разрозненных при передаче запросов.

Модуль транспорта SSL осуществляет поддержку транспортов, основанных на слое безопасных сокетов (SSL). В основе модуля лежит библиотека OpenSSL. Поддерживаются входящие и исходящие транспорты протоколов SSLv2, SSLv3 и TLSv1.

Сконфигурированный и запущенный входящий транспорт открывает серверный SSL-сокет для ожидания соединения клиентов. SSL-сокеты являются многопоточными, т.е. при подключении клиента создаётся клиентское SSL-соединение и новый поток, в котором производится обслуживание клиента. Серверный SSL-сокет, в этот момент, переходит к ожиданию запросов от нового клиента. Таким образом, достигается параллельное обслуживание клиентов.

Каждый входной транспорт обязательно связывается с одним из доступных транспортных протоколов, которому передаются входящие сообщения. В связке с транспортным протоколом поддерживается механизм объединения кусков раздробленных, при передаче, запросов.

Сконфигурированный и запущенный выходной транспорт открывает SSL соединение с указанным сервером. При разрыве соединения, выходной транспорт отключается. Для возобновления соединения транспорт нужно опять запустить.

Для полноценной работы модуля необходимы сертификаты и приватные ключи. В случае с входным SSL-транспортом (сервером) они обязательны. В случае с выходным SSL-транспортом их использование желательно.

Простейшей конфигурацией сертификата является самоподписной сертификат и приватный ключ. Ниже описана процедура их создания с помощью утилиты openssl:

```
# Генерация секретного ключа
$ openssl genrsa -out ./key.pem -des3 -rand /var/log/messages 2048
# Генерация самоподписанного сертификата
$ openssl req -x509 -new -key ./key.pem -out ./selfcert.pem -days 365
```

Далее содержимое файлов selfcert.pem и key.pem копируется в текстовое поле сертификата и ключа. Пароль приватного ключа устанавливается в соответствующем поле.

3.5.2 Конфигурирование подсистемы «Транспорты»

Для конфигурации подсистемы предусмотрена корневая страница подсистемы "Транспорты", содержащая вкладки "Подсистема", "Модули" и "Помощь".

Вид вкладки "Подсистема" показан на рисунке 39.

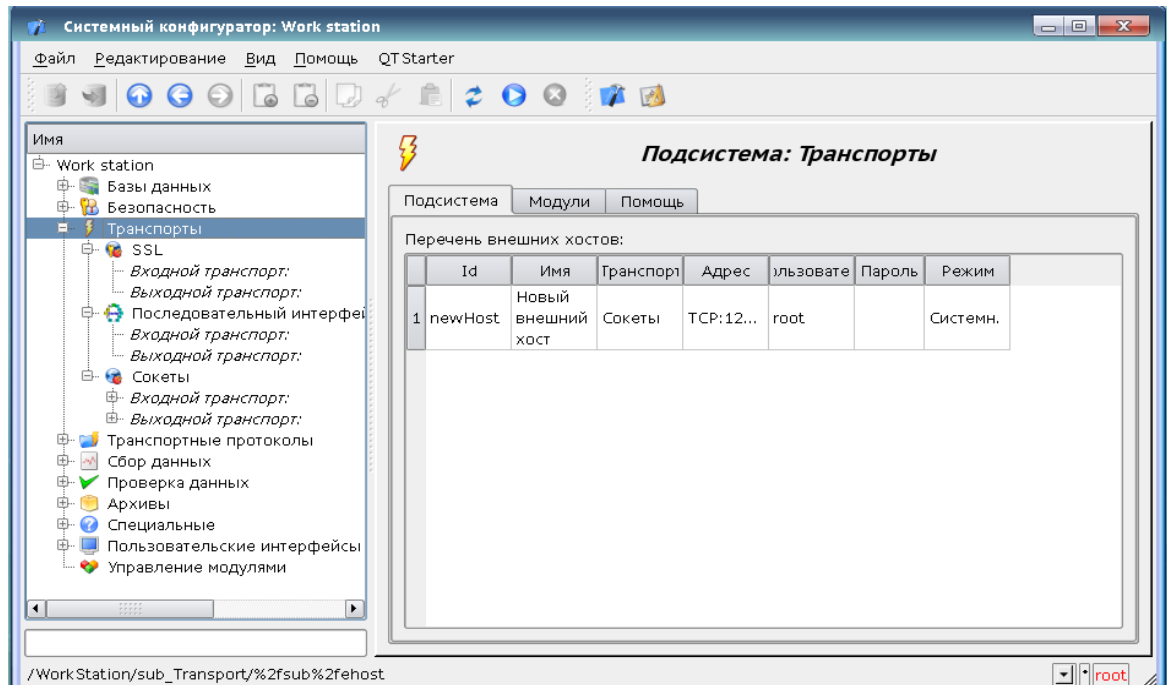


Рисунок 39

Вкладка "Подсистема" содержит таблицу конфигурации внешних для данной СКАДА станций. Внешние станции могут быть системными и пользовательскими, что выбирается соответствующим параметром. Системные внешние станции доступны только привилегированному пользователю и используются компонентами системного назначения, например, механизмом горизонтального резервирования и модулем DAQ.DAQGate.

Пользовательские внешние станции привязаны к пользователю, который их создавал, а это значит, что список пользовательских внешних станций индивидуален для каждого пользователя.

Пользовательские внешние станции используются компонентами графического интерфейса, например, UI.QTCfg и UI.Vision. В таблице внешних станций возможно добавление и удаление записей про станцию, а также их модификация. Каждая запись содержит поля:

- *Id* - идентификатор внешней станции;
- *Имя* - имя внешней станции;
- *Транспорт* - выбор из списка модуля подсистемы "Транспорты" для использования его в доступе к внешней станции;

- *Адрес* - адрес внешней станции в формате, специфичном для выбранного в предыдущем поле модуля подсистемы "Транспорты". Форматы адресов для различных типов транспортов приведены в таблице 3;

- *Пользователь* - имя/идентификатор пользователя удалённой станции, от имени которого выполнять подключение;

- *Пароль* - пароль пользователя удалённой станции;

- *Режим* – выбор из списка режимов:

- пользовательский – предоставляет дерево конфигурации удаленного хоста;
- системный – предоставляет возможность указывать подключение;
- пользовательский и системный – совмещает два этих режима.

Таблица 3

Тип транспорта	Формат адреса
Последовательный, входной	[dev]:[spd]:[format]:[fc]:[mdm]. Где: dev - адрес последовательного устройства (/dev/ttyS0); spd - скорость последовательного устройства из ряда: 300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200, 230400, 460800, 500000, 576000 или 921600; format - формат асинхронных данных "<размер><чётность><стоп>" (8N1, 7E1, 5O2, ...); fc - управление потоком: 'h' - аппаратное (CRTSCTS), 's' - программное (IXON IXOFF); mdm - режим модема, ожидание 'RING'.
Последовательный, выходной	[dev]:[spd]:[format]:[fc]:[modTel]. Где: dev - адрес последовательного устройства (/dev/ttyS0); spd - скорость последовательного устройства из ряда: 300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200, 230400, 460800, 500000, 576000 или 921600; format - формат асинхронных данных "<размер><чётность><стоп>" (8N1, 7E1, 5O2, ...); fc - управление потоком: 'h' - аппаратное (CRTSCTS), 's' - программное (IXON IXOFF); modTel - телефон модема, присутствие этого поля переключает транспорт на работу в режиме модема.

Тип транспорта	Формат адреса
Сокеты, входной	<p>TCP:[адрес]:[порт]:[режим] где:</p> <p>адрес – Адрес, на котором открывается сокет. Должен быть одним из адресов хоста. Если ничего не указано, то сокет будет доступен на всех интерфейсах хоста. Допускаются как символьное, так и IP представление адреса.</p> <p>порт – Сетевой порт, на котором открывается сокет. Возможно указание символьного имени порта (в соответствии с /etc/services).</p> <p>режим – режим работы входящего сокета (0 – разрывать соединение после сеанса приём-ответ; 1 – не разрывать).</p> <p>Пример: <TCP::10001:1> – TCP-сокет доступен на всех интерфейсах, открыт на порту 10001 и соединения не разрывает.</p> <p>UDP:[адрес]:[порт] где:</p> <p>адрес – тоже что в TCP;</p> <p>порт – тоже что в TCP.</p> <p>Пример: <UDP:localhost:10001> – UDP-сокет доступен только на интерфейсе “localhost” и открыт на порту 10001.</p> <p>UNIX:[имя]:[режим] где:</p> <p>имя – имя файла UNIX сокета;</p> <p>режим – тоже что в TCP.</p> <p>Пример: <UNIX:/tmp/scada:1> – UNIX-сокет доступен через файл /tmp/scada и соединения не разрывает.</p>
Сокеты, выходной	<p>TCP:[адрес]:[порт] UDP:[адрес]:[порт] где:</p> <p>адрес – Адрес, с которым выполняется соединение. Допускаются как символьное, так и IP представление адреса.</p> <p>порт – Сетевой порт, с которым выполняется соединение. Возможно указание символьного имени порта (в соответствии с /etc/services).</p> <p>Пример: <TCP:127.0.0.1:7634> – соединится с портом 7634 на хосте 127.0.0.1.</p> <p>UNIX:[имя] где:</p> <p>имя – имя файла UNIX сокета.</p> <p>Пример: <UNIX:/tmp/scada> – соединится с UNIX-сокетом через файл /tmp/scada.</p>

Тип транспорта	Формат адреса
SSL, входной	<p>[адрес]:[порт]:[режим]</p> <p>адрес – Адрес, на котором открывается SSL. Должен быть одним из адресов хоста. Если указано "*" то SSL будет доступен на всех интерфейсах хоста. Допускаются как символьное, так и IP представление адреса.</p> <p>порт – Сетевой порт, на котором открывается SSL. Возможно указание символьного имени порта (в соответствии с /etc/services).</p> <p>режим – SSL-режим и версия (SSLv2, SSLv3, SSLv23, TLSv1). По умолчанию и при ошибке используется SSLv23</p>
SSL, выходной	<p>[адрес]:[порт]:[режим]</p> <p>адрес – Адрес, с которым выполняется соединение. Допускаются как символьное так и IP представление адреса.</p> <p>порт – Сетевой порт, с которым выполняется соединение. Возможно указание символьного имени порта (в соответствии с /etc/services).</p> <p>режим – SSL-режим и версия (SSLv2, SSLv3, SSLv23, TLSv1). По умолчанию и при ошибке используется SSLv23</p>

Вкладка "Модули" содержит список модулей подсистемы "Транспорты" и идентична для всех модульных подсистем. Вкладка "Помощь" содержит краткую помощь для данной страницы.

Каждый модуль подсистемы "Транспорты" предоставляет конфигурационную страницу с вкладками "Транспорты" и "Помощь". Вид вкладки "Транспорты" показан на рисунке 40.

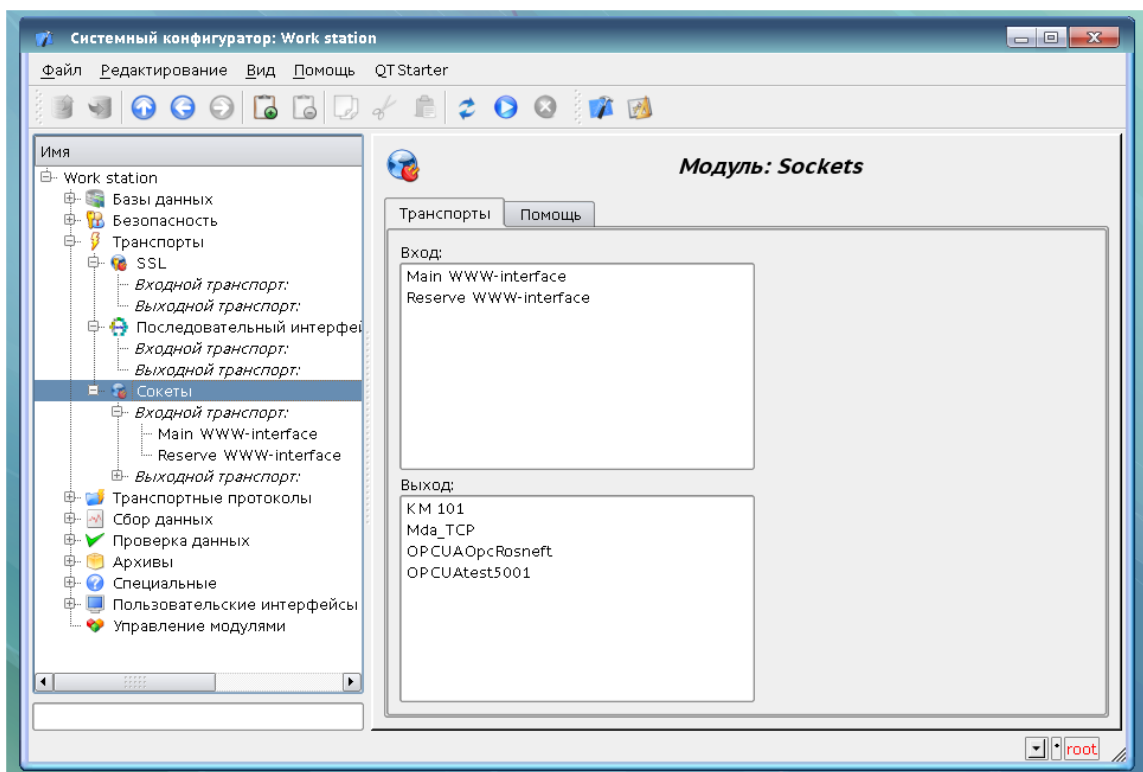


Рисунок 40

Вкладка "Транспорты" содержит список входных и выходных транспортов, зарегистрированных в модуле. В контекстном меню списков транспортов пользователю предоставляется возможность добавления, удаления и перехода к нужному транспорту. Во вкладке "Помощь" содержится информация о модуле подсистемы "Транспорты", состав которой идентичен для всех модулей.

Каждый транспорт содержит собственную страницу конфигурации с одной вкладкой "Транспорт". Эта вкладка содержит основные настройки транспорта. Вид страницы входного транспорта показан на рисунке 41.

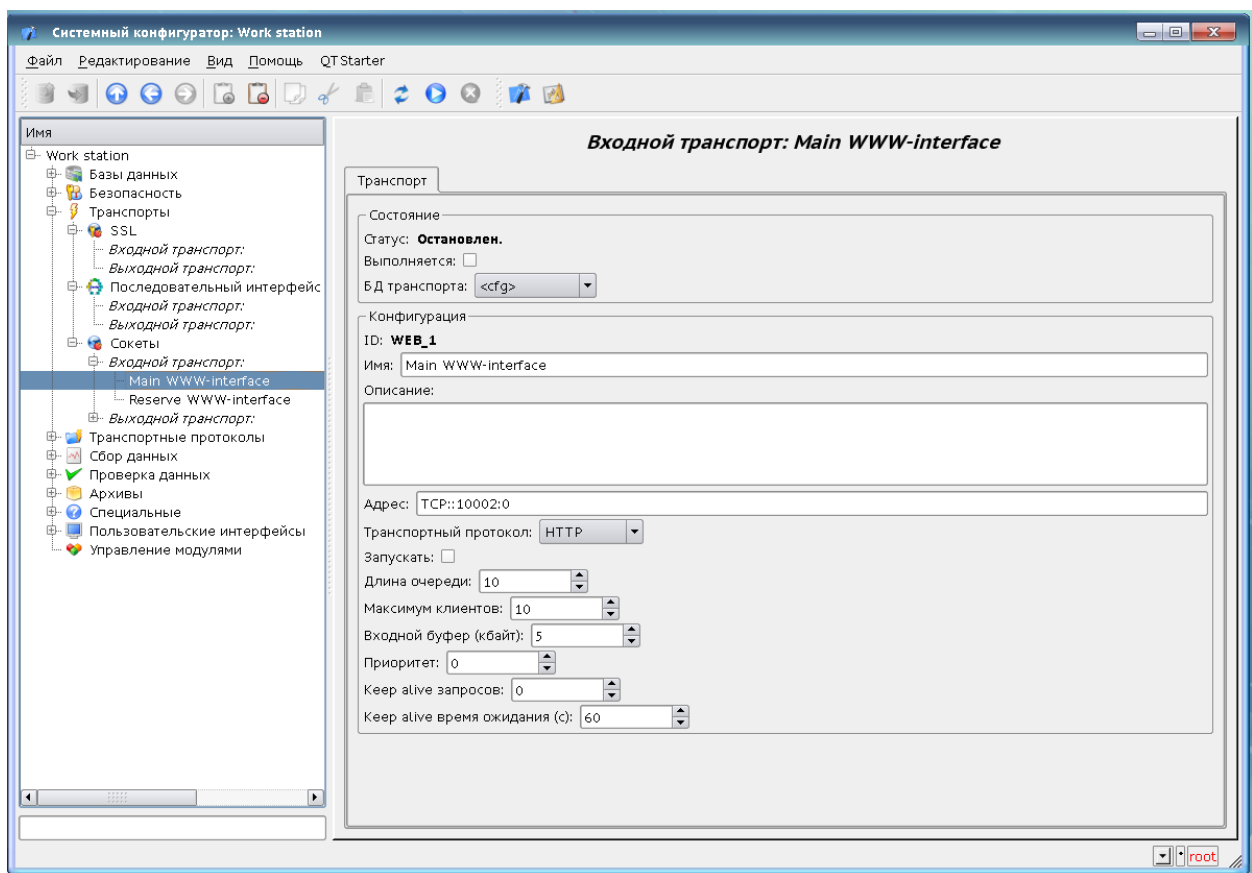


Рисунок 41

Входной транспорт содержит:

- Раздел "Состояние" - содержит свойства, характеризующие состояние транспорта:
- *Статус* - информация о текущем состоянии транспорта и статистика его работы;
- *Выполняется* - состояние транспорта "Выполняется";
- *БД транспорта* - адрес БД для хранения данных транспорта, выбирается из списка щелчком мыши.

- Раздел "Конфигурация" - непосредственно содержит поля конфигурации:

- *ID* - информация об идентификаторе транспорта;
- *Имя* - указывает имя транспорта;
- *Описание* - краткое описание транспорта и его назначения;
- *Адрес* - адрес транспорта в специфичном для типа транспорта (модуля) формате.

Описание формата записи адреса транспорта, как правило, доступно во всплывающей подсказке этого поля (см. таблицу 3);

- *Транспортный протокол* - указывает на модуль транспортного протокола (подсистема "Транспортные протоколы"), который должен работать в связке с данным входным транспортом. Т.е. полученные неструктурированные данные этот модуль будет направлять на структуризацию и обработку указанному модулю транспортного протокола.

- *Запускать* - указывает на состояние "Выполняется", в которое следует перевести транспорт при загрузке;

- *Длина очереди* - размер очереди сокетов;

- *Максимум клиентов* - максимальное количество обслуживаемых клиентов;

- *Входной буфер* - размер входного буфера;

- *Приоритет* – уровень приоритета (0 – стандартный пользовательский приоритет);

- *Keep alive запросов* – закрытие подключения после указанного количества запросов (если задан 0, то никогда не закрывать);

- *Keep alive время ожидания* (с) - закрытие подключения после отсутствия запросов в течении указанного времени (если указан ноль, то никогда не закрывать).

Вид страницы выходного транспорта показан на рисунке 42.

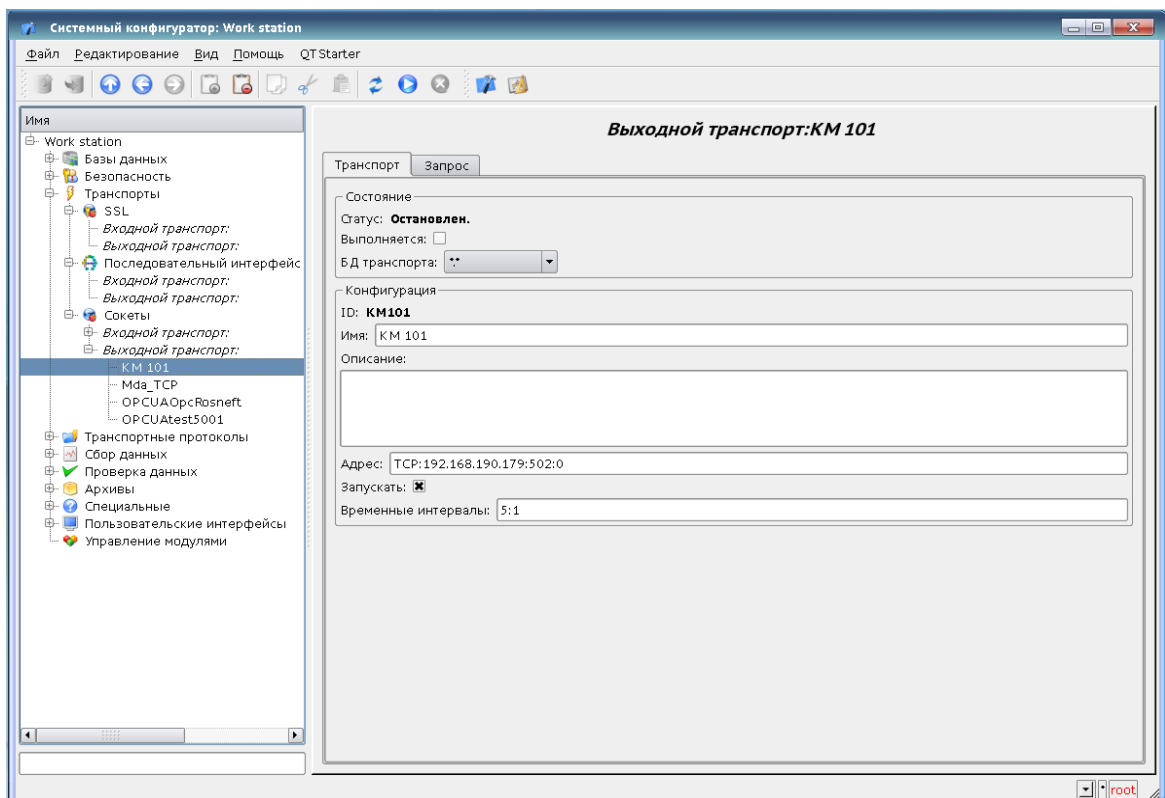


Рисунок 42

Выходной транспорт содержит:

- Раздел "Состояние" - содержит свойства, характеризующие состояние транспорта:

- *Статус* - информация о текущем состоянии транспорта и статистика его работы;

- *Выполняется* - состояние транспорта "Выполняется";

- *БД транспорта* - адрес БД для хранения данных транспорта, выбирается из списка щелчком мыши.

- Раздел "Конфигурация" - непосредственно содержит поля конфигурации:

- *ID* - информация об идентификаторе транспорта;

- *Имя* - указывает имя транспорта;
- *Описание* - краткое описание транспорта и его назначения;
- *Адрес* - адрес транспорта в специфичном для типа транспорта (модуля) формате.

Описание формата записи адреса транспорта, как правило, доступно во всплывающей подсказке этого поля (см. таблицу 3);

- *Запускать* - указывает на состояние "Выполняется", в которое следует перевести транспорт при загрузке;

- *Временные интервалы* - временные интервалы интерфейса в формате строки: "[conn]:[symbol]". Где:

- conn - время ожидания соединения, т.е. ответа от удалённого устройства.
- symbol - время символа в миллисекундах. Используется для контроля факта окончания фрейма.

Выходной транспорт, в дополнение, предоставляет вкладку формирования пользовательского запроса через данный транспорт. Вид вкладки показан на рисунке 43.

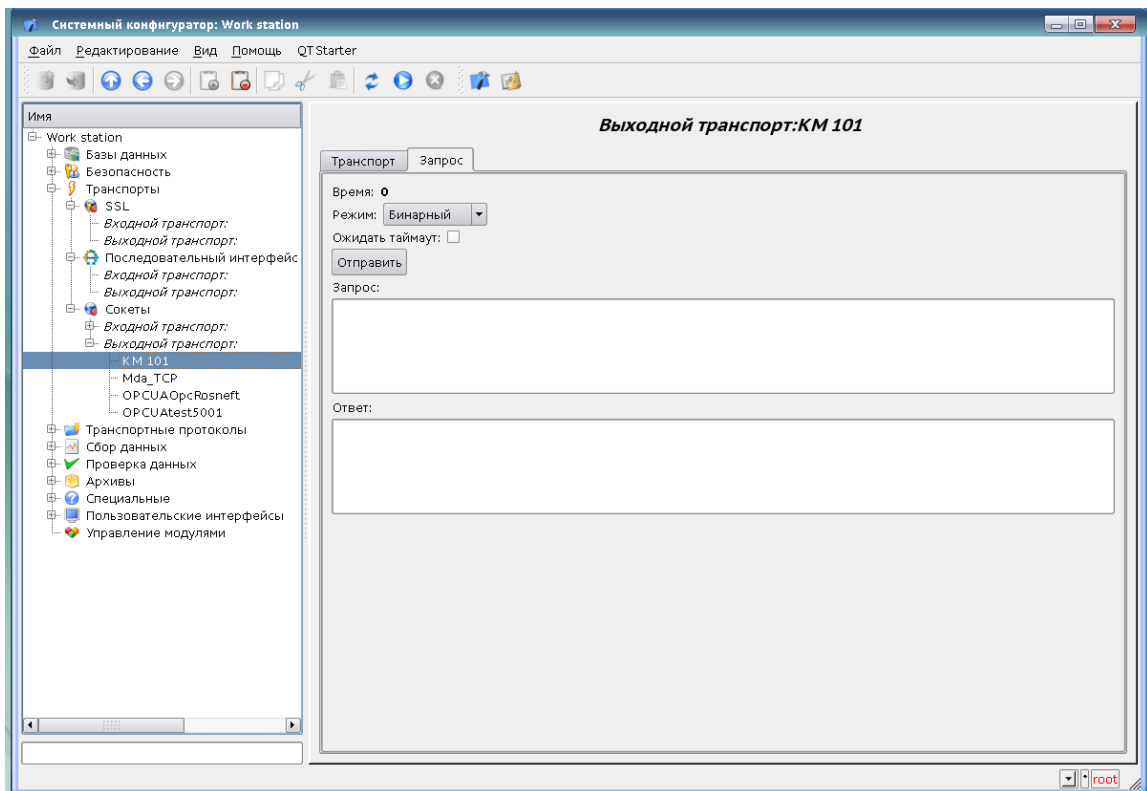


Рисунок 43

Вкладка предназначена для наладки связи, а также для отладки протоколов и содержит:

- *Время (мс)* - информация о времени, затраченном на запрос и получение ответа;
- *Режим* - указывает режим данных, из списка "Текст" и "Бинарный", в котором будет формироваться запрос и предоставляться ответ. В бинарном режиме данные

записываются парами чисел в шестнадцатеричном исчислении, т.е. байтами, разделёнными пробелами;

- *Ожидать таймаут* – признак ожидания таймаута при получении ответа. При установке этого флага будет осуществляться ожидание всех кусков ответа, вплоть до отсутствия данных в течение таймаута дождания ответа (например, протокол HTTP).

- *Отправить* - команда отправить запрос;

- *Запрос* - содержит запрос в выбранном режиме представления данных;

- *Ответ* - предоставляет ответ в выбранном режиме представления данных.

3.6 Подсистема "Транспортные протоколы"

3.6.1 Общие сведения

Подсистема "Транспортные протоколы" предназначена для структуризации данных, полученных от подсистемы "Транспорты". По сути, подсистема "Транспортные протоколы" является продолжением подсистемы "Транспорты" и выполняет функции проверки структуры и целостности полученных данных. Так, для указания протокола, в связке с которым должен работать транспорт, предусмотрено специальное конфигурационное поле. Модульным объектом, содержащимся в подсистеме "Протоколы", является сам протокол. Например, транспортными протоколами могут быть:

- OPC UA;
- собственный протокол СКАДА;
- ModBUS;
- HTTP;
- пользовательский протокол.

Полную цепочку связи можно описать следующим образом:

- сообщение передаётся в транспорт;
- транспорт передаёт сообщение связанному с ним протоколу путём создания нового объекта протокола;
- протокол проверяет целостность данных;
- если пришли все данные, то сообщить транспорту о прекращении ожидания данных и передать ему ответ иначе сообщить, что нужно ожидать ещё;
- транспорт, получив подтверждение, отсылает ответ и удаляет объект протокола;
- если подтверждения нет, то транспорт продолжает ожидание данных, и в случае их поступления передаёт их сохранённому объекту протокола.

Поддерживаются протоколы и для исходящих транспортов. Исходящий протокол берёт на себя функцию общения с транспортом и реализацию особенностей протокола. Внутренняя сторона доступа к протоколу реализуется потоковым образом с собственной структурой для каждого протокольного модуля. Такой механизм позволяет выполнять прозрачный доступ к внешней системе, посредством транспорта, просто указывая имя протокола, с помощью которого обслуживать передачу.

3.6.2 Модуль «OPC UA»

OPC UA - это платформи-независимый стандарт, с помощью которого системы и устройства различного типа могут взаимодействовать путём отправки сообщений между клиентом и сервером через различные типы сетей. Протокол поддерживает безопасное взаимодействие путём валидации клиентов и серверов, а также противодействия атакам. OPC UA определяет понятие Сервисы, которые сервера могут предоставлять, а также сервисы, которые сервер поддерживает для клиента. Информация передаётся в виде определённых OPC UA типов данных.

OPC UA предоставляет совмещение интегрированного адресного пространства и сервисной модели. Это позволяет серверу интегрировать данные, нарушения (Alarms) и события (Events), историю в этом адресном пространстве, а также предоставлять доступ к ним посредством интегрированных сервисов. Сервисы также предоставляют интегрированную модель безопасности.

OPC UA позволяет серверам предоставлять для клиентов определения типов, для доступа к объектам из адресного пространства. OPC UA допускает предоставление данных в различных форматах, включая бинарные структуры и XML-документы. Через адресное пространство клиенты могут запросить у сервера метаданные, которые описывают формат данных.

OPC UA добавляет поддержку множественной связности между узлами вместо простого ограничения иерархичностью.

В ПП «СКАДА А-СОФТ» модуль OPC UA содержит код реализации протокольной части OPC UA как для клиентского, так и для серверного сервисов. Для построения OPC UA сервера достаточно создать входной транспорт, обычно это TCP-транспорт модуля Sockets, и выбрать в нём модуль данного протокола (рисунок 45), а также сконфигурировать хотя бы один конечный узел модуля протокола.

Входной транспорт создается в соответствии с пунктом 3.5.2:

- в модуле «Транспорты» добавить входной транспорт «Сокет»;
- в качестве транспортного протокола выбрать OPC UA;
- указать адрес сервера и порт соединения;
- сохранить сделанные изменения в БД.

На рисунке 44 представлено окно настройки входного транспорта «opcServer».

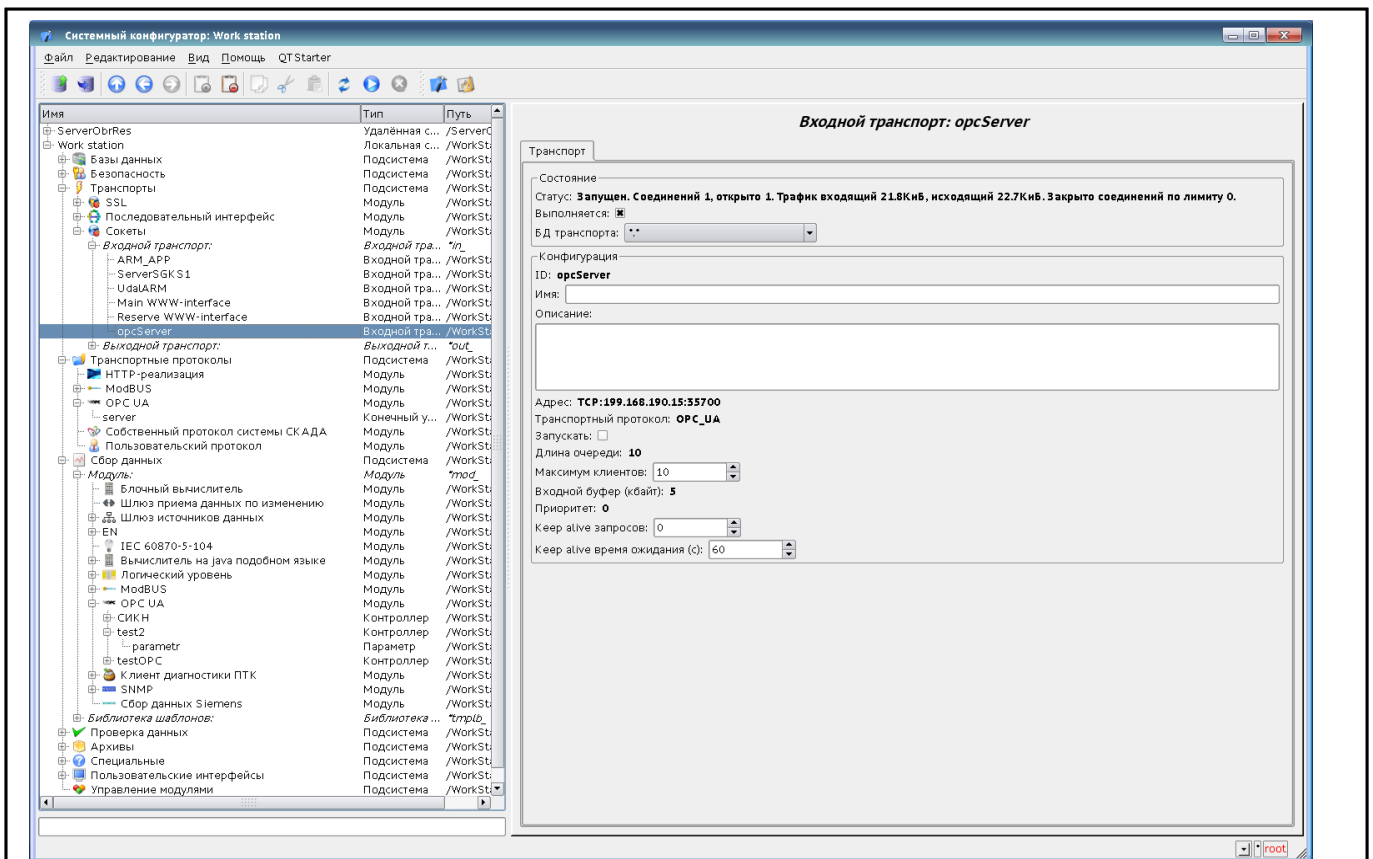


Рисунок 44

Входящие запросы к модулю-протоколу обрабатываются модулем в соответствии со сконфигурированными конечными узлами OPC UA.

Конечный узел протокола OPC UA - это фактически объект сервера OPC UA. В ПП «СКАДА А-СОФТ» реализованы только локальные конечные узлы. Локальный конечный узел предназначен для предоставления ресурсов СКАДА-станции по протоколу OPC UA.

Общая конфигурация конечного узла осуществляется на главной вкладке страницы «Транспортные протоколы» параметрами:

- состояние узла, а именно: статус, "Включен" и имя БД, содержащей конфигурацию;
- идентификатор, имя и описание узла;
- состояние, в которое переводить узел при загрузке: "Включен";
- тип кодирования протокола ("Бинарный");
- URL конечной точки – адрес сервера и порт соединения;
- сертификат сервера в формате PEM - поле обязательное для заполнения;
- приватный ключ в формате PEM - поле обязательное для заполнения;
- политики безопасности сервера – выбирается значение None(нет).

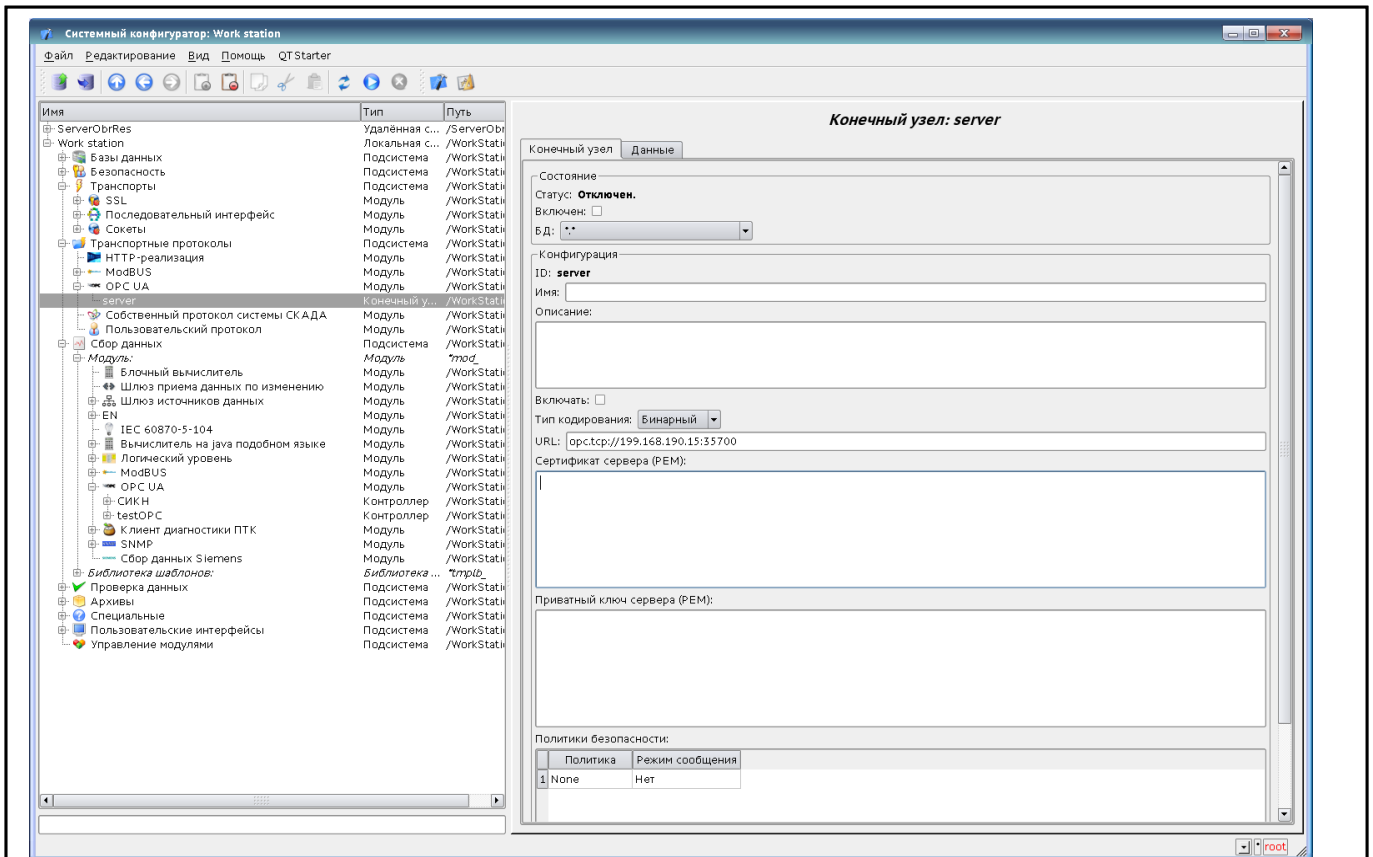


Рисунок 45

Настройка клиента OPC UA осуществляется в одноименном модуле подсистемы «Сбор данных» и описана в части 3 настоящего руководства оператора.

3.6.3 Модуль «HTTP»

Модуль транспортного протокола HTTP предназначен для реализации поддержки сетевого протокола HTTP (Hypertext Transfer Protocol) в системе СКАДА.

Протокол HTTP используется для передачи содержимого WWW. Так, через HTTP передаются следующие типы документов: html, xhtml, png, java и другие. Добавление поддержки HTTP в ПП «СКАДА А-СОФТ» в комплексе с транспортом Sockets позволяет реализовывать различные пользовательские функции на основе WWW интерфейса. Модуль HTTP реализует два основных метода протокола HTTP: GET и POST. Модуль HTTP обеспечивает контроль целостности HTTP-запросов и, совместно с транспортом Sockets, позволяет “собирать” целостные запросы из их фрагментов, а также обеспечивать сохранение соединения живым (Keep-Alive).

Для гибкого подключения пользовательских интерфейсов к данному модулю используются модули подсистемы “Пользовательские интерфейсы” с дополнительным информационным полем “SubType”, имеющим значение “WWW”.

В запросах к Web ресурсам используется URL (Universal Resource Locator), который передаётся как основной параметр через HTTP. Первый элемент запрашиваемого URL используется для идентификации модуля UI. Например URL: `http://localhost:10002/WebCfg` означает обращение к модулю "WebCfg" на хосте `http://localhost:10002`. В случае ошибочного указания идентификатора модуля или при обращении вообще без идентификатора "HTTP" модуль генерирует диалог информации о входе и с предоставлением выбора одного из доступных пользовательских интерфейсов.

3.6.4 Модуль «ModBUS»

ModBUS — коммуникационный протокол, основан на клиент-серверной архитектуре для использования в контроллерах с программируемой логикой (PLC). Широко применяется для организации связи промышленного электронного оборудования. Используется для передачи данных через последовательные линии связи RS-485, RS-422, RS-232, а также сети TCP/IP.

Существуют три режима протокола: ModBUS/RTU, ModBUS/ASCII и ModBUS/TCP. Первые два используют последовательные линии связи (в основном RS-485, реже RS-422/RS-232), последний используется для передачи данных сети TCP/IP.

Модуль «ModBUS» подсистемы «Сбор данных» предоставляет возможность собирать информацию у различных устройств по протоколу ModBUS во всех режимах, а также модулем реализуются функции горизонтального резервирования, а именно совместной работы с удалённой станцией этого же уровня. В то же время модуль протокола позволяет сформировать и выдать данные по протоколу ModBUS в различных режимах и через интерфейсы, поддерживаемые модулями подсистемы "Транспорты".

Протокол ModBUS/RTU предполагает одно ведущее (запрашивающее) устройство в линии (master), которое может передавать команды одному или нескольким ведомым устройствам (slave), обращаясь к ним по уникальному в линии адресу. Синтаксис команд протокола позволяет адресовать 247 устройств на одной линии связи стандарта RS-485 (реже RS-422 или RS-232). В случае с режимом TCP адресация исключена из протокола, поскольку выполняется на уровне TCP/IP стека.

Инициатива проведения обмена всегда исходит от ведущего устройства. Ведомые устройства прослушивают линию связи. Мастер подаёт запрос (посылка, последовательность байт) в линию и переходит в состояние прослушивания линии связи. Ведомое устройство отвечает на запрос, пришедший в его адрес.

Окончание ответной посылки определяется в зависимости от режима. В режиме RTU окончание посылки определяется по временному интервалу между окончанием приёма

предыдущего байта и началом приёма следующего, время символа. Если этот интервал превысил время, необходимое для приёма полтора байта на заданной скорости передачи то приём фрейма ответа считается завершённым. В режиме ASCII критерием окончания посылки является символ '\r', а в режиме TCP — ожидаемый размер посылки, информация о котором присутствует в заголовке пакета.

Все операции с данными привязаны к нулю, каждый вид данных (регистр, бит, регистр входа или бита входа) начинаются с адреса 0000 и заканчиваются 65535.

В протоколе ModBUS можно выделить несколько подмножеств команд:

- стандартные – коды 1 - 21;
- резерв для расширенных функций – коды 22-64;
- пользовательские – коды 65-119;
- резерв для внутренних нужд – коды 120-255.

Модулем сбора данных используются команды 0x03 и 0x06(0x10) для чтения и записи регистров, 0x01 и 0x05(0x0F) для чтения и записи битов, 0x02 и 0x04 для чтения бита и регистра входа соответственно.

Модуль протокола обрабатывает запросы командами 0x03 и 0x06(0x10) для чтения и записи регистров, 0x01 и 0x05(0x0F) для чтения и записи битов.

Входная часть обслуживания запросов к модулю протокола осуществляет проверку и обработку запросов посредством объектов узлов. Узел протокола эквивалентен физическому узлу устройства сети ModBUS. Узел протокола может работать в трёх режимах:

- "Данные" — режим отражения данных системы СКАДА на массивы регистров и битов ModBUS с передачей их по запросу клиентского узла или мастера;
- "Шлюз узла" — режим перенаправления (шлюзования) запросов к узлу в другой сети ModBUS через данный узел;
- "Шлюз сети" — режим перенаправления запросов к любому узлу в другую сеть ModBUS, фактически выполняя интеграцию нескольких сетей ModBUS в одну.

Поскольку узлов протокола может быть создано множество, то получается, что на одном интерфейсе, т.е. в одной сети, одна СКАДА-станция может прозрачно представлять несколько узлов сети ModBUS с различными данными.

Часть 3 настоящего руководства оператора содержит подробное описание конфигурирования связи по протоколу ModBUS.

3.6.5 *Модуль «Собственный протокол системы СКАДА» (SelfSystem)*

Модуль транспортного протокола SelfSystem предназначен для отражения интерфейса управления СКАДА-системы в сеть с целью предоставления возможности внешним системам взаимодействовать с системой СКАДА, а также для взаимодействия станций, построенных на основе СКАДА между собой. Модуль используется для обеспечения удалённого конфигурирования одной СКАДА станции из другой через сеть посредством модуля конфигурирования QTCfg.

3.6.6 *Модуль «Пользовательский протокол» (UserProtocol)*

Модуль транспортного протокола UserProtocol предназначен для предоставления пользователю возможности создания реализаций различных протоколов собственными силами на одном из внутренних языков СКАДА, обычно JavaLikeCalc, и, не прибегая, к низкоуровневому программированию СКАДА.

Основная цель модуля - упростить задачу подключения к системе СКАДА устройств источников данных, которые имеют незначительное распространение и/или предоставляют доступ к собственным данным по специфическому протоколу, обычно достаточно простому для реализации на внутреннем языке СКАДА. Для реализации этого предоставляется механизм формирования протокола исходящего запроса.

Кроме механизма протокола исходящего запроса предоставляется механизм протокола входящего запроса, который позволяет СКАДА обслуживать запросы на получение данных по специфическим протоколам, которые достаточно просто могут быть реализованы на внутреннем языке СКАДА.

Модуль предоставляет возможность создания реализаций множества различных протоколов в объекте "Пользовательский протокол" (рисунок 46).

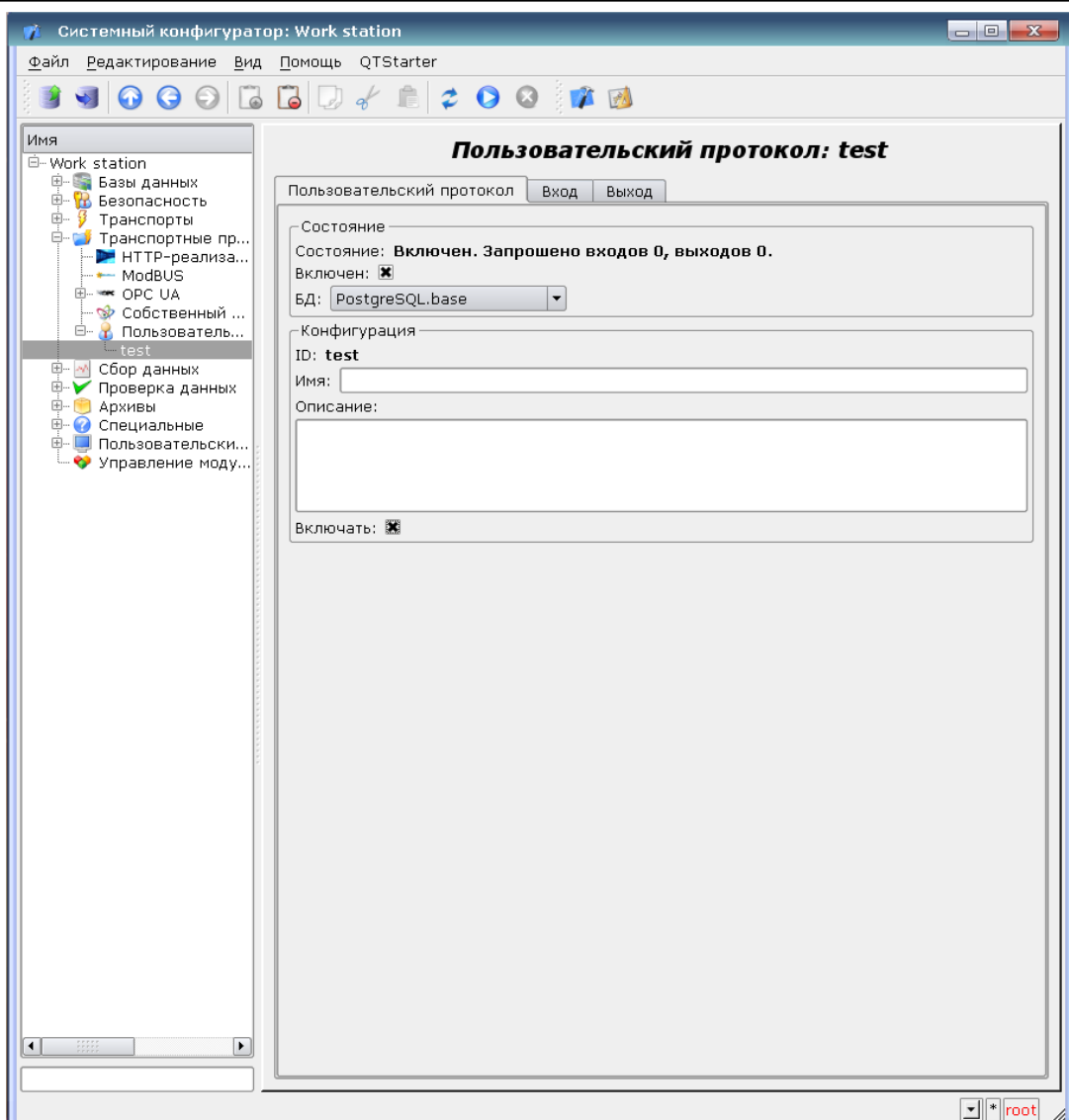


Рисунок 46

Главная вкладка содержит основные настройки пользовательского протокола:

Раздел "Состояние" - содержит свойства, характеризующие состояние протокола:

- *Состояние* - текущее состояние протокола.
- *Включен* - состояние протокола "Включен".
- *БД* - БД в которой хранится конфигурация.

Раздел "Конфигурация" - непосредственно содержит поля конфигурации:

- *ID* - информация об идентификаторе протокола.
- *Имя* - указывает имя протокола.
- *Описание* - краткое описание протокола и его назначения.
- *Включать* - указывает на состояние "Включен", в которое переводить протокол при загрузке.

3.6.7 Конфигурирование подсистемы «Транспортные протоколы»

Для конфигурации подсистемы предусмотрена корневая страница подсистемы "Транспортные протоколы", содержащая вкладки "Модули" и "Помощь".

Вкладка "Модули" (рисунок 47) содержит список модулей подсистемы "Транспортные протоколы".

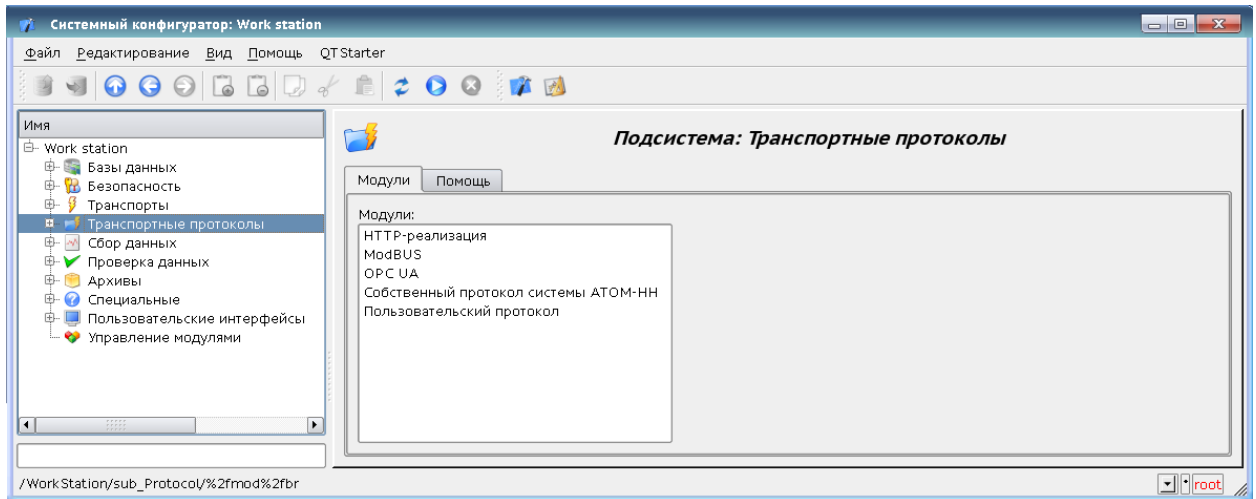


Рисунок 47

Вкладка "Помощь" содержит краткую помощь для данной страницы.

Каждый модуль подсистемы "Транспортные протоколы" предоставляет конфигурационную страницу с индивидуальными настройками в соответствии с выбранным протоколом, а также содержит вкладку "Помощь" с информацией о модуле подсистемы (рисунок 48).

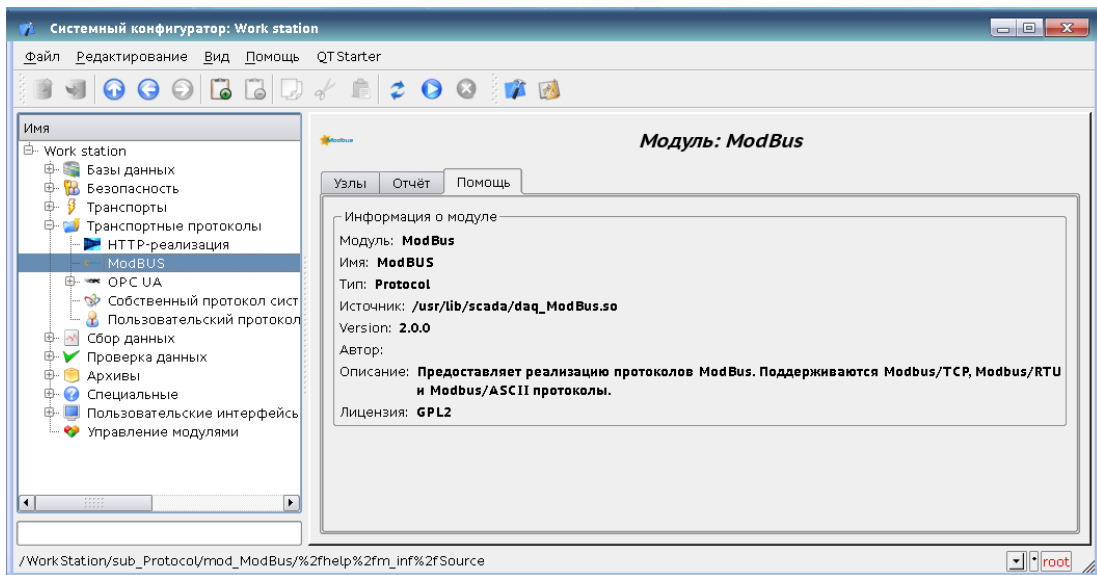


Рисунок 48

3.7 Подсистема "Сбор данных"

3.7.1 Общие сведения

Подсистема «Сбор данных» предназначена для обеспечения поддержки источников динамических данных, будь то PLC-контроллеры, платы УСО, виртуальные источники и т.д., а также различных механизмов взаимодействия в СКАДА. В функции этой подсистемы входит предоставление полученных данных в структурированном виде и обеспечение управления этими данными, например, модификация данных.

Подсистема является модульной. В качестве модуля подсистемы выступает драйвер для сопряжения с источником данных отдельного типа. Каждый модуль может содержать конфигурацию нескольких устройств этого типа в виде объектов "Контроллер" СКАДА.

В настоящее время поддерживаются следующие типы источников данных:

- сбор данных операционной системы (ОС);
- блочный вычислитель;
- вычислитель на Java-подобном языке;
- шлюз приема данных по изменению от удаленных СКАДА станций в локальную;
- шлюз источников данных подсистемы "Сбор данных" от одной СКАДА станции к другой;
- протокол EN;
- протокол HART;
- протокол ModBUS;
- протокол OPC UA;
- DCON клиент - реализация клиентского сервиса протокола DCON, поддерживается протокол I-7000 DCON;
- IEC 60870-5-104 клиент;
- IEC 61850 клиент;
- клиент диагностики ПТК;
- TANGO;
- PCI;
- сбор данных сетевых устройств посредством протокола SNMP;
- источник данных логического уровня СКАДА.

Каждый тип источника выполнен в виде отдельного модуля, который может быть подключен/отключен. Описание особенностей источников данных приведено в частях 3 и 4 настоящего руководства оператора.

В терминах системы СКАДА предоставляются следующие объекты для обслуживания механизма сбора данных:

- атрибут — объект отражения данных сигнала, включает текущее значение с типом сигнала и историю изменения значений;
- параметр — объект группы атрибутов (сигналов) со структурой, соответствующей особенностям отдельно взятого источника данных;
- контроллер — объект отдельного устройства данных.

Отдельно взятый контроллер может содержать параметры определённых модулем типов. Например, параметры аналогового типа: основной информацией, которую они предоставляют, является значение целого или вещественного типа. Структурно параметр представляет собой список атрибутов, которые и содержат данные. Атрибуты могут быть четырёх базовых типов: символьная строка (текст), целое, вещественное и логический тип.

Источник динамических данных может быть удалённым, т.е. быть подключен на удалённой СКАДА-системе.

3.7.2 *Конфигурирование подсистемы «Сбор данных»*

Для конфигурации подсистемы предусмотрена корневая страница подсистемы "Сбор данных", содержащая вкладки "Резервирование", "Задачи событий по изменению", "Библиотеки шаблонов", "Модули" и "Помощь". Вкладка "Задачи событий по изменению" в настоящее время не используется.

Для получения доступа на модификацию объектов этой подсистемы необходимы права пользователя в группе "DAQ" или права привилегированного пользователя.

На рисунке 49 представлен вид вкладки "Резервирование" подсистемы "Сбор данных".

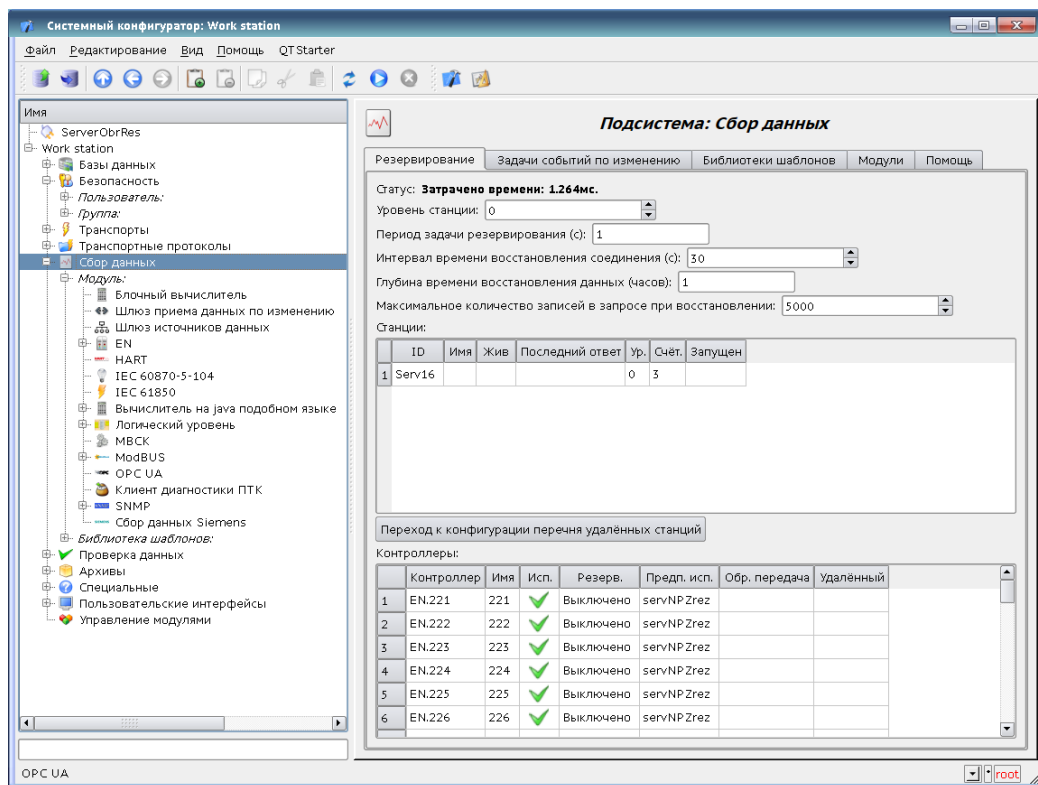


Рисунок 49

Вкладка "Резервирование" содержит конфигурацию резервирования источников данных подсистемы "Сбор данных" станции в составе настроек:

- *Статус* - содержит информацию о работе схемы резервирования, обычно это время, затраченное на исполнения одного цикла задачи обслуживания резерва;
- *Уровень станции* - указывает уровень данной станции в схеме резервирования (0-255);
- *Период задачи резервирования* - указывает периодичность исполнения задачи резервирования в секундах (1-255);
- *Интервал времени восстановления соединения* - указывает интервал времени, через который возможно осуществить попытку восстановления соединения с потерянной резервной станцией в секундах (0-255);
- *Глубина времени восстановления данных* - указывает на максимальную глубину архивных данных для восстановления из архива удалённой станции при запуске, в часах (0-12);
- *Станции* - содержит таблицу с информацией о резервных станциях. Станции можно добавлять и удалять посредством контекстного меню. Идентификатор добавленных станций нужно выбрать из списка доступных системных станций СКАДА. Таблица предоставляет следующую информацию о станции:

- *ID* - идентификатор системной станции СКАДА, должен быть изменён после добавления путём выбора из перечня доступных идентификаторов;
- *Имя* - имя системной станции СКАДА;
- *Жив* - признак наличия связи с резервной станцией;
- *Last time* – время последнего соединения с резервной станцией;
- *Уровень* - уровень удалённой станции в схеме резервирования;
- *Счётчик* - счётчик запросов к резервной станции или времени ожидания в случае отсутствия связи;
- *Запущен* - список доступных контроллеров с признаком (+) локального исполнения на удалённой станции.
- *Переход к конфигурации перечня удалённых станций* - команда для перехода на страницу конфигурации удалённых СКАДА станций, в подсистеме "Транспорты".

- *Контроллеры* - содержит таблицу с перечнем контроллеров, доступных для резервирования, и текущее их состояние:

- *Контроллер* - полный идентификатор контроллера;
- *Имя* - имя контроллера;
- *Запущен* - признак исполнения контроллера локальной станцией;
- *Резервирование* - режим резервирования контроллера, может быть выбран из перечня: "Выключен", "Асимметричное" и "Архивное";

- *Предпочтение исполнения* - конфигурация предпочтительного исполнения на указанной станции, может иметь следующие значения:

- <Высокий уровень> - исполнение на станции с наивысшим уровнем,
- <Низкий уровень> - исполнение на станции с самым низким уровнем,
- <Оптимально> - выбор для исполнения наименее нагруженной станции.

- *Удалённый* - признак, указывающий на исполнение контроллера удалённой станцией и перевод локальной в режим синхронизации данных из удалённой станции.

Вкладка "Библиотеки шаблонов" содержит список библиотек шаблонов для параметров этой подсистемы. Вид вкладки показан на рисунке 50.

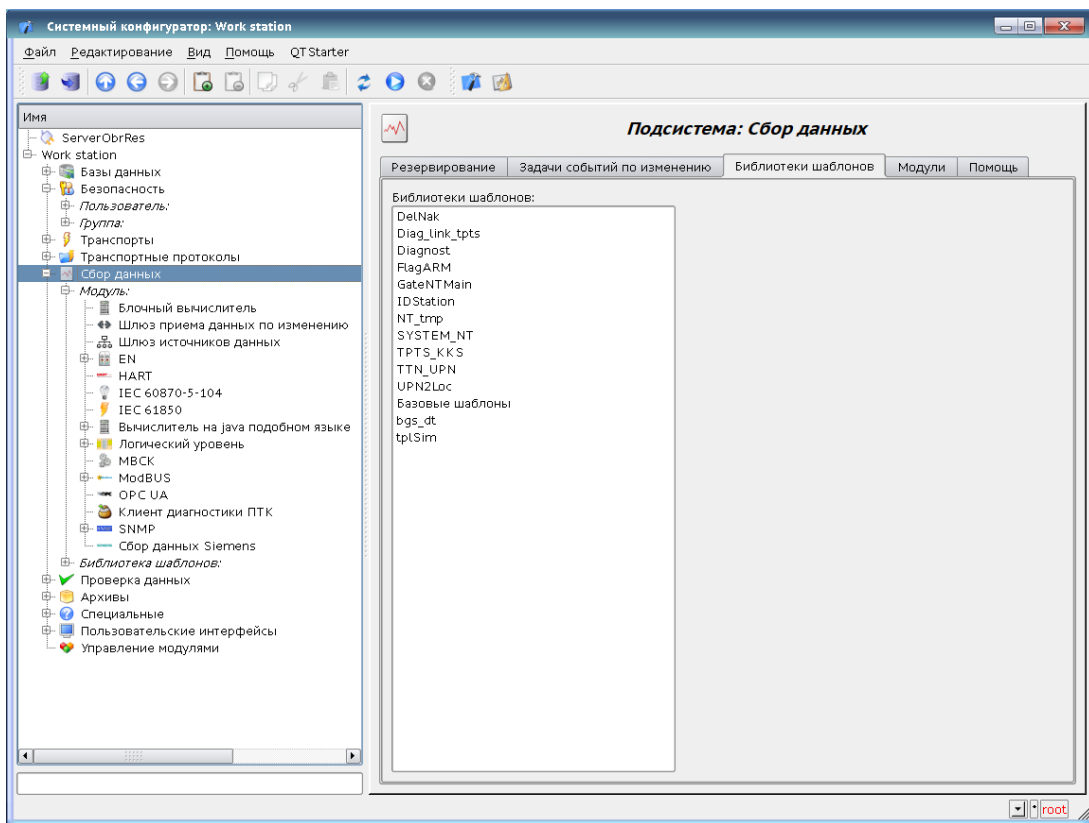


Рисунок 50

В контекстном меню списка библиотек шаблонов пользователю предоставляется возможность добавления, удаления и перехода к нужной библиотеке.

Вкладка "Модули" содержит список модулей подсистемы "Сбор данных" и идентична для всех модульных подсистем.

Вкладка "Помощь" содержит краткую помощь для данной страницы.

Каждая библиотека шаблонов подсистемы "Сбор данных" предоставляет конфигурационную страницу с вкладками "Библиотека" и "Шаблоны параметров". Вкладка "Библиотека" (рисунок 51) содержит основные настройки библиотеки в составе:

- Раздел "Состояние" - содержит свойства, характеризующие состояние библиотеки:
- *Доступен* - состояние библиотеки "Доступен";
- *БД библиотеки* - адрес БД для хранения данных библиотеки и шаблонов, выбирается из списка щелчком мыши.
- Раздел "Конфигурация" - непосредственно содержит поля конфигурации:
- *ID* - информация об идентификаторе библиотеки;
- *Имя* - указывает имя библиотеки;
- *Описание* - краткое описание библиотеки и её назначения.

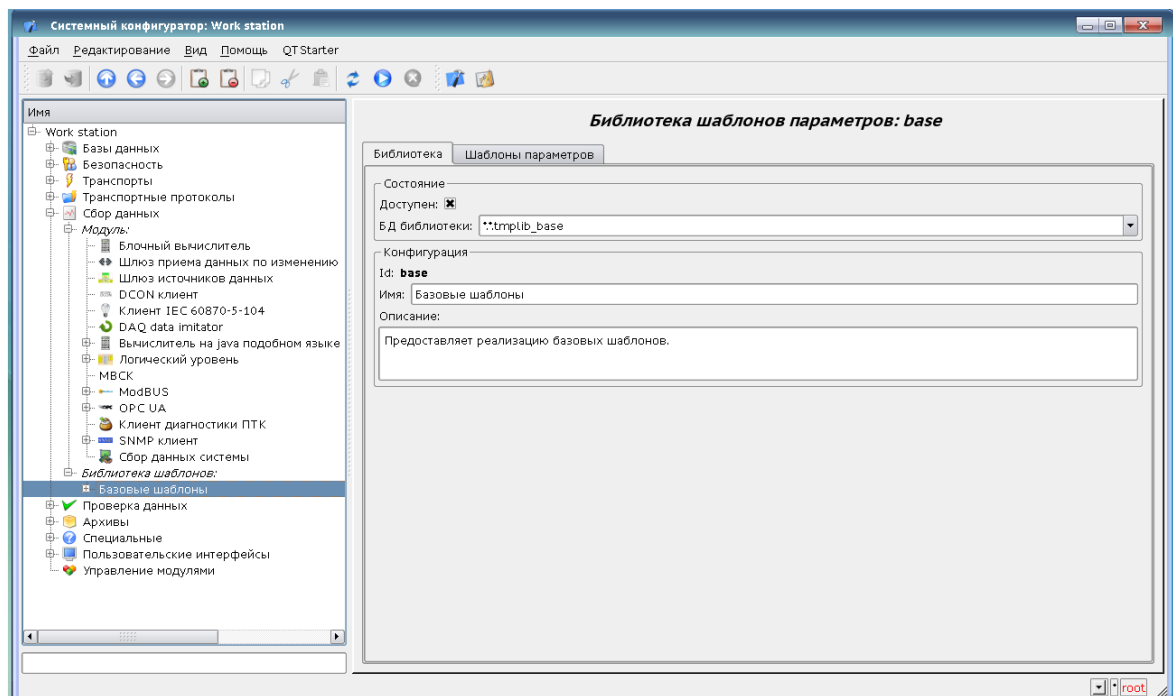


Рисунок 51

Вкладка "Шаблоны параметров" (рисунок 52) содержит список шаблонов в библиотеке. В контекстном меню списка пользователю предоставляется возможность добавления, удаления и перехода к нужному шаблону.

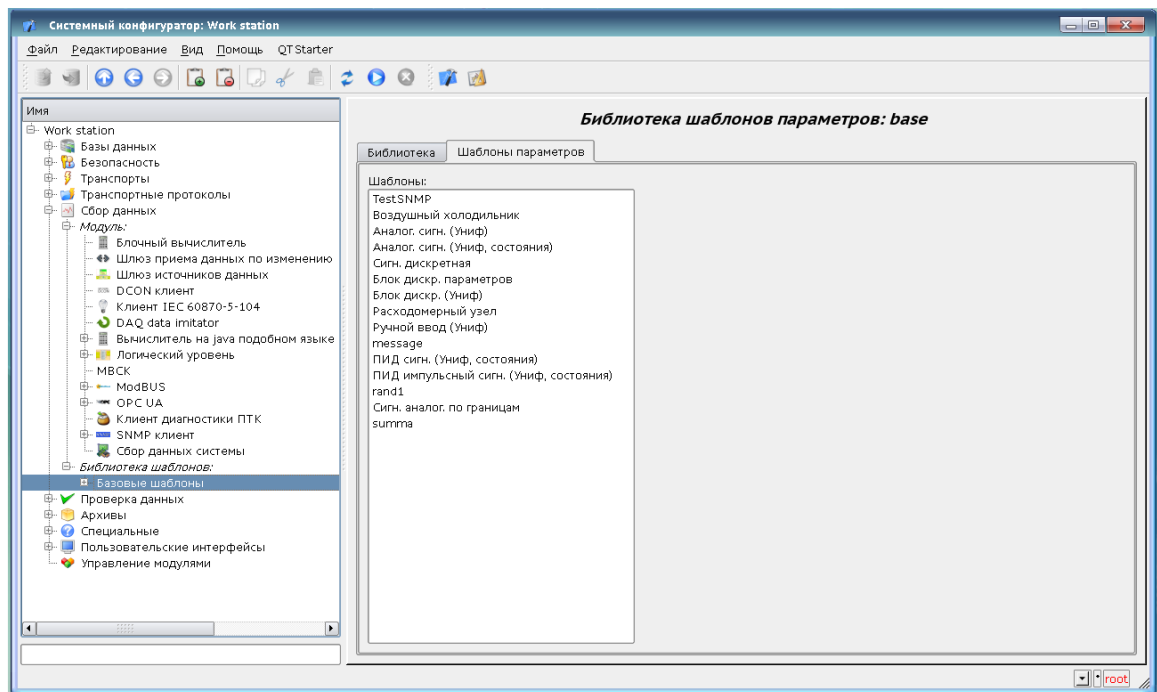


Рисунок 52

Каждый шаблон библиотеки шаблонов предоставляет конфигурационную страницу с вкладками "Шаблон" и "Ю". Вид вкладки "Шаблон" показан на рисунке 53.

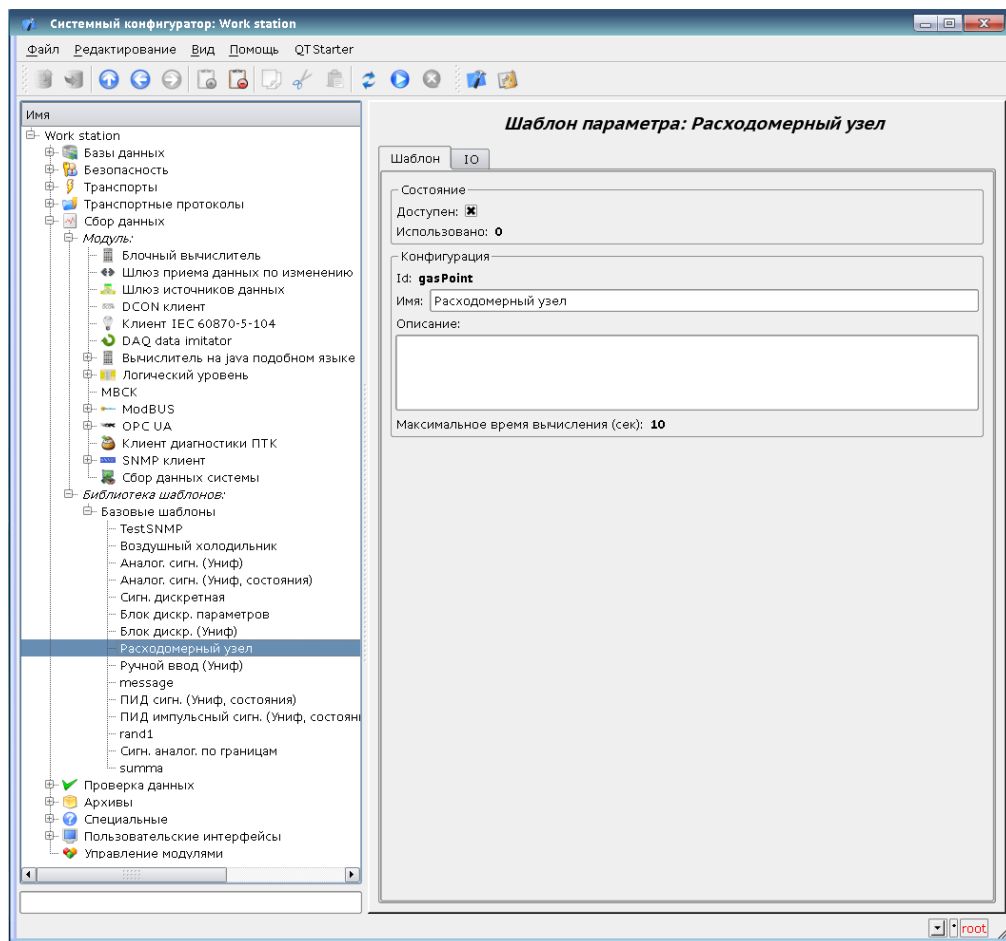


Рисунок 53

Вкладка "Шаблон" содержит основные настройки шаблона в составе:

- Раздел "Состояние" - содержит свойства, характеризующие состояние шаблона:
 - *Доступен* - состояние шаблона "Доступен";
 - *Использовано* - указывает, сколько раз шаблон использован;
- Раздел "Конфигурация" - непосредственно содержит поля конфигурации:
 - *ID* - информация об идентификаторе шаблона;
 - *Имя* - указывает имя шаблона;
 - *Описание* - краткое описание шаблона и его назначения.

Вид вкладки "IO" показан на рисунке 54.

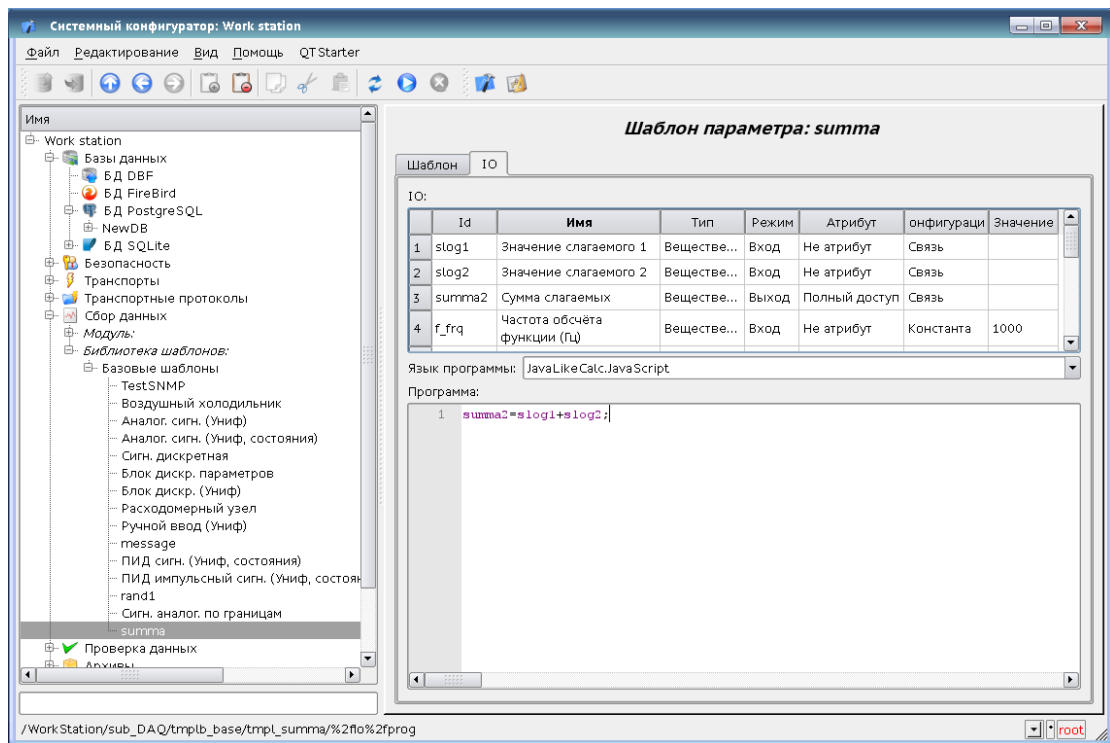


Рисунок 54

Вкладка "IO" содержит конфигурацию атрибутов (IO) шаблонов и программу шаблона на встроенном языке DAQ.JavaLikeCalc.JavaScript. Описание языка приведено в части 5 настоящего руководства.

В таблице атрибутов шаблона пользователь может посредством контекстного меню добавить, вставить, удалить, передвинуть вверх или вниз запись атрибута, а также отредактировать поля атрибута:

- *Id* - идентификатор атрибута;
- *Имя* - имя атрибута;
- *Тип* - выбор типа значения атрибута из списка: "Вещественный", "Целый", "Логический", "Строка".
- *Режим* - выбор режима атрибута из списка: "Вход", "Выход";
- *Атрибут* - режим атрибута параметра, реализованного на основе шаблона из списка: "Не атрибут", "Только чтение", "Полный доступ". Для атрибутов шаблона, у которых это поле установлено, будет создаваться соответствующий атрибут у параметра контроллера этой подсистемы;
- *Конфигурация* - режим конфигурации атрибута во вкладке конфигурации шаблона у параметра контроллера этой подсистемы из списка: "Константа", "Публичная константа", "Связь". В режимах "Публичная константа" и "Связь" во вкладке конфигурации шаблона будут добавлены эти атрибуты для установки константы или указания внешней связи параметра.

- *Значение* - значение атрибута по умолчанию или шаблон ссылки для доступа по ссылке. Формат шаблона ссылки зависит от компонента, который его использует. Обычно для модуля DAQ.LogicLev шаблон ссылки записывается в виде: **{Параметр}{атрибут}**. Поле **{Параметр}** - указывает имя параметра как контейнера атрибутов. Атрибуты с одинаковым значением **{Параметр}** будут группироваться и позволят назначаться только указанием контейнера атрибутов, а отдельные атрибуты будут связаны с атрибутами контейнера в соответствии с полем **{атрибут}**.

Каждый модуль подсистемы "Сбор данных" предоставляет конфигурационную страницу с вкладками "Контроллеры" и "Помощь". Вид вкладки "Контроллеры" показан на рисунке 55.

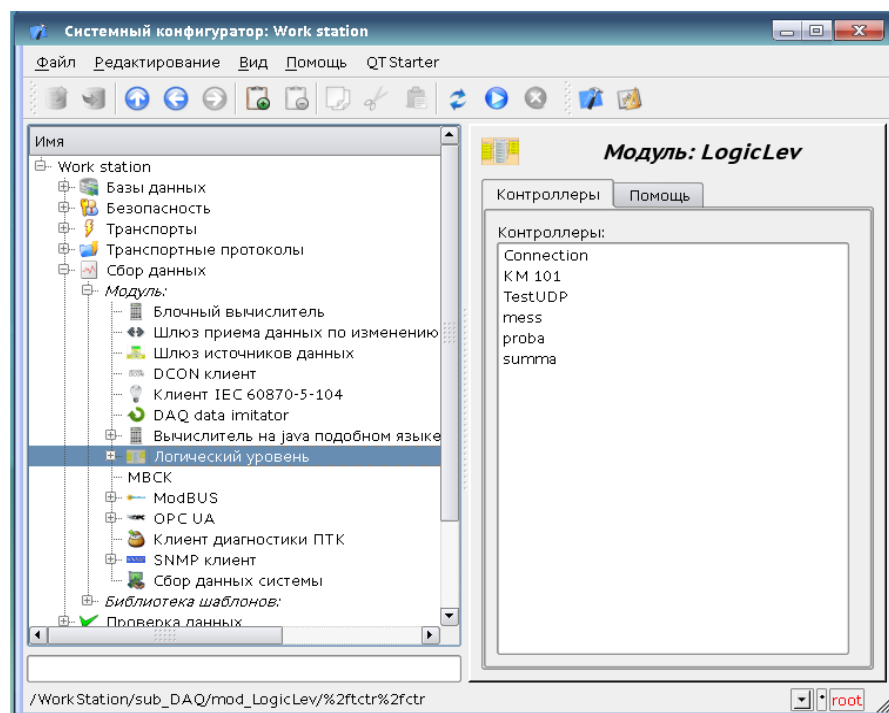


Рисунок 55

Вкладка "Контроллеры" содержит список контроллеров, зарегистрированных в модуле. В контекстном меню списка пользователю предоставляется возможность добавления, удаления и перехода к нужному контроллеру. Во вкладке "Помощь" содержится информация о модуле подсистемы "Сбор данных", состав которой идентичен для всех модулей.

Каждый контроллер содержит собственную страницу конфигурации с вкладками "Контроллер" и "Параметры".

Вид вкладки "Контроллер" показан на рисунке 56.

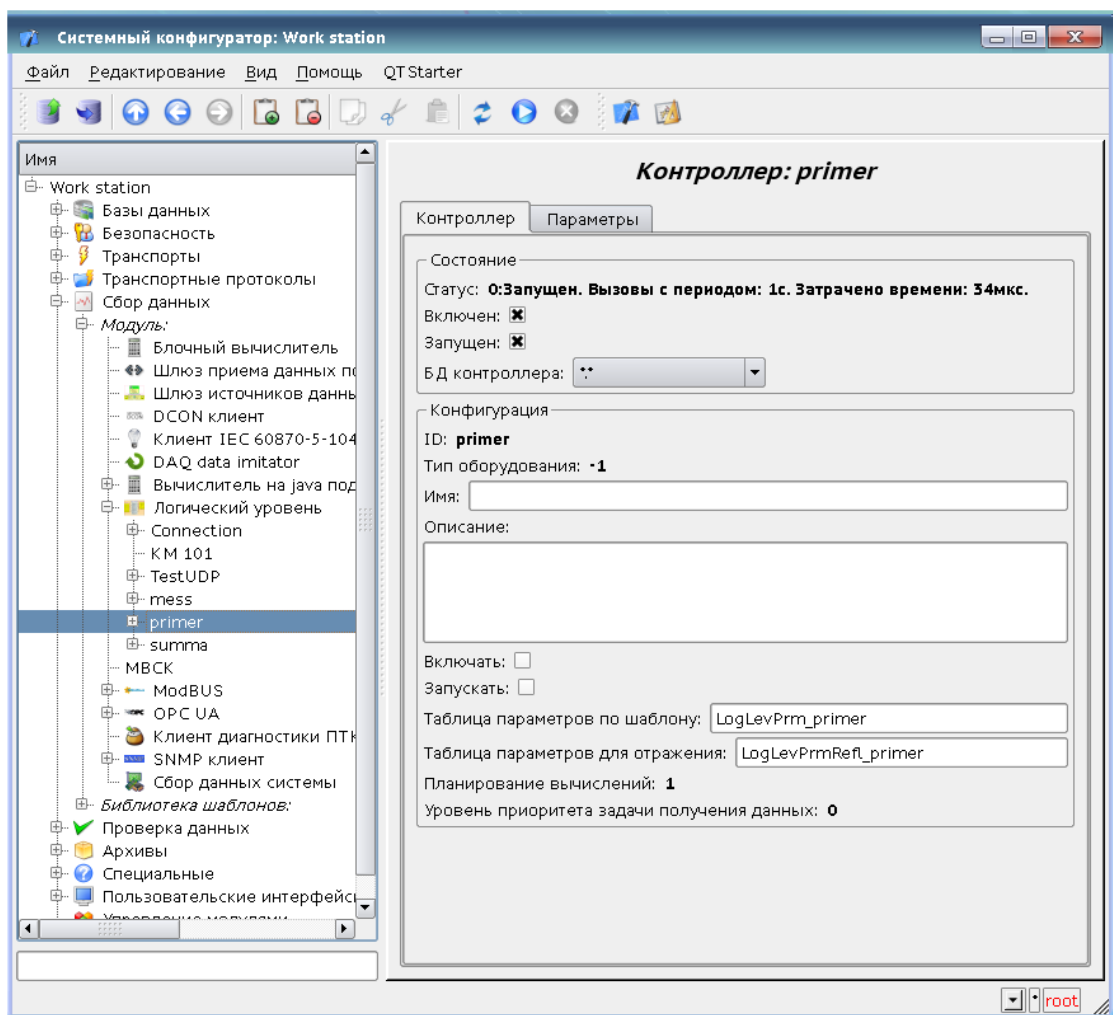


Рисунок 56

Вкладка "Контроллер" содержит основные настройки. Состав этих настроек может несколько отличаться от одного модуля этой подсистемы к другому, о чём можно узнать в собственной документации модулей. В качестве примера рассмотрим настройки контроллера модуля «Логический уровень» DAQ.LogicLev:

- Раздел "Состояние" - содержит свойства, характеризующие состояние контроллера:
 - *Статус* - указывает статус контроллера и время его вычисления;
 - *Включен* - состояние контроллера "Включен". Включенный контроллер предоставляет возможность создания параметров и их конфигурации;
 - *Запущен* - состояние контроллера "Запущен". Исполняющийся контроллер выполняет физический сбор данных и/или включает механизмы доступа к этим данным;
 - *БД контроллера* - адрес БД для хранения данных контроллера и его параметров, выбирается из списка щелчком мыши.
- Раздел "Конфигурация" - непосредственно содержит поля конфигурации:
 - *ID* - информация об идентификаторе контроллера;

- *Имя* - указывает имя контроллера;
- *Описание* - краткое описание контроллера и его назначения;
- *Включать* - указывает на состояние "Включать", в которое следует перевести контроллер при загрузке;
- *Запускать* - указывает на состояние "Запущен", в которое следует перевести контроллер при загрузке;
- *Таблица параметров по шаблону* - имя таблицы, в которой сохранять параметры (имеются в виду объекты параметров сбора данных) контроллера;
- *Планирование вычислений (с)* - периодичность выполнения задачи;
- *Уровень приоритета задачи получения данных* - устанавливает приоритетность задачи сбора данных этого контроллера. Используется при планировании задач операционной системой. В случае исполнения станции от имени суперпользователя "root" это поле включает планирование задачи контроллера в режиме реального времени и с указанным приоритетом.

Вкладка "Параметры" содержит список параметров в контроллере, а также информацию об общем количестве и количестве включенных параметров. В контекстном меню списка пользователю предоставляется возможность добавления, удаления и перехода к нужному параметру (рисунок 57).

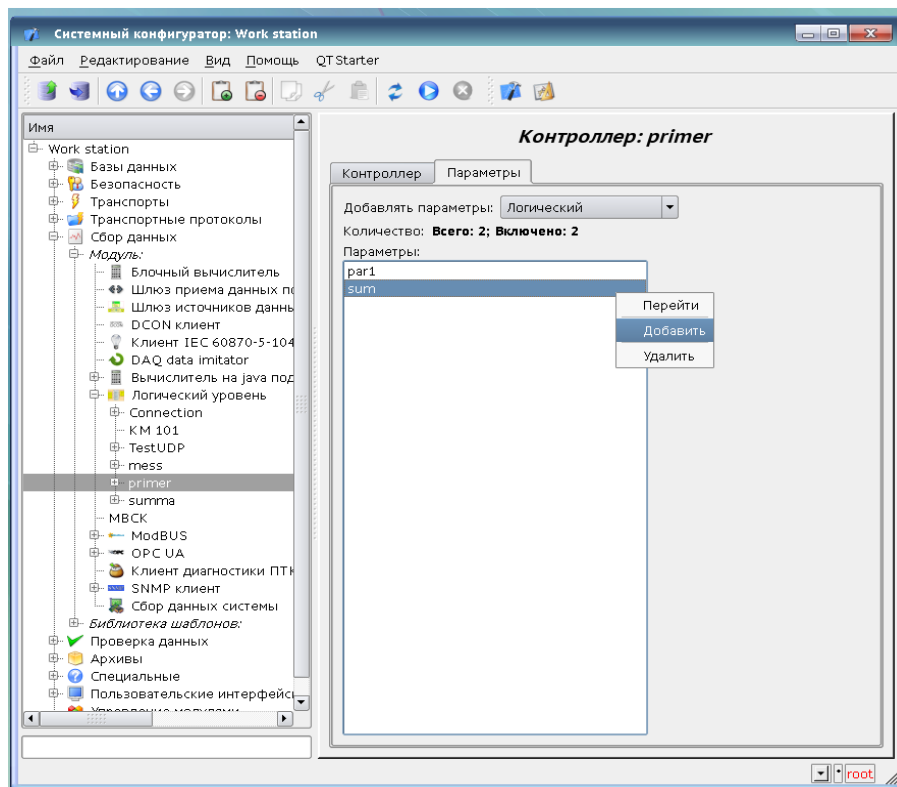


Рисунок 57

Параметры контроллеров подсистемы "Сбор данных" предоставляют конфигурационную страницу с вкладками "Параметр", "Атрибуты", "Архивация" и "Конфигурация шаблона". Вкладка "Конфигурация шаблона" не является стандартной, а присутствует только в модулях подсистемы "Сбор данных", которые реализуют механизмы работы по шаблону в контексте источника данных, ими обслуживаемого.

Вкладка "Параметр" (рисунок 58) содержит основные настройки в составе:

- Раздел "Состояние" - содержит свойства, характеризующие состояние параметра:
- *Тип* - информация о типе параметра;
- *Включен* - состояние параметра "Включен". Включенный параметр используется контроллером для сбора данных.
- Раздел "Конфигурация" - непосредственно содержит поля конфигурации:
- *ID* - содержит информацию об идентификаторе параметра;
- *Имя* - указывает имя параметра;
- *Описание* - краткое описание параметра и его назначения;
- *Включать* - указывает на состояние "Включать", в которое переводить параметр при загрузке;

- *Шаблон параметра* – выбор из выпадающего списка шаблонов, настроенных в «Библиотеке шаблонов».

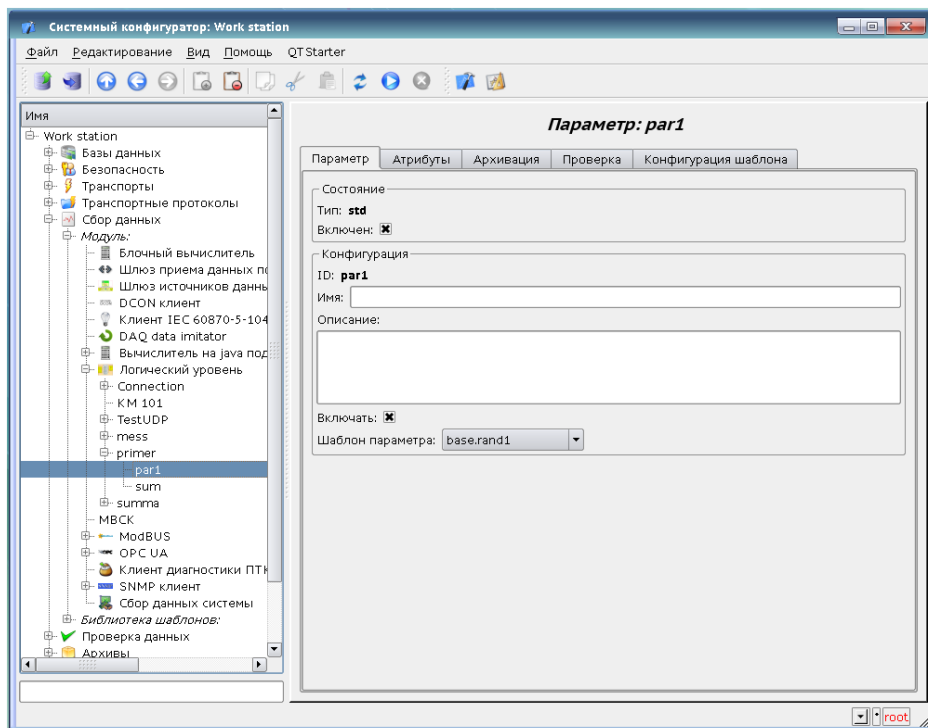


Рисунок 58

Вкладка "Атрибуты" (рисунок 59) содержит атрибуты параметра и их значения в соответствии с конфигурацией используемого шаблона и выполнением его программы.

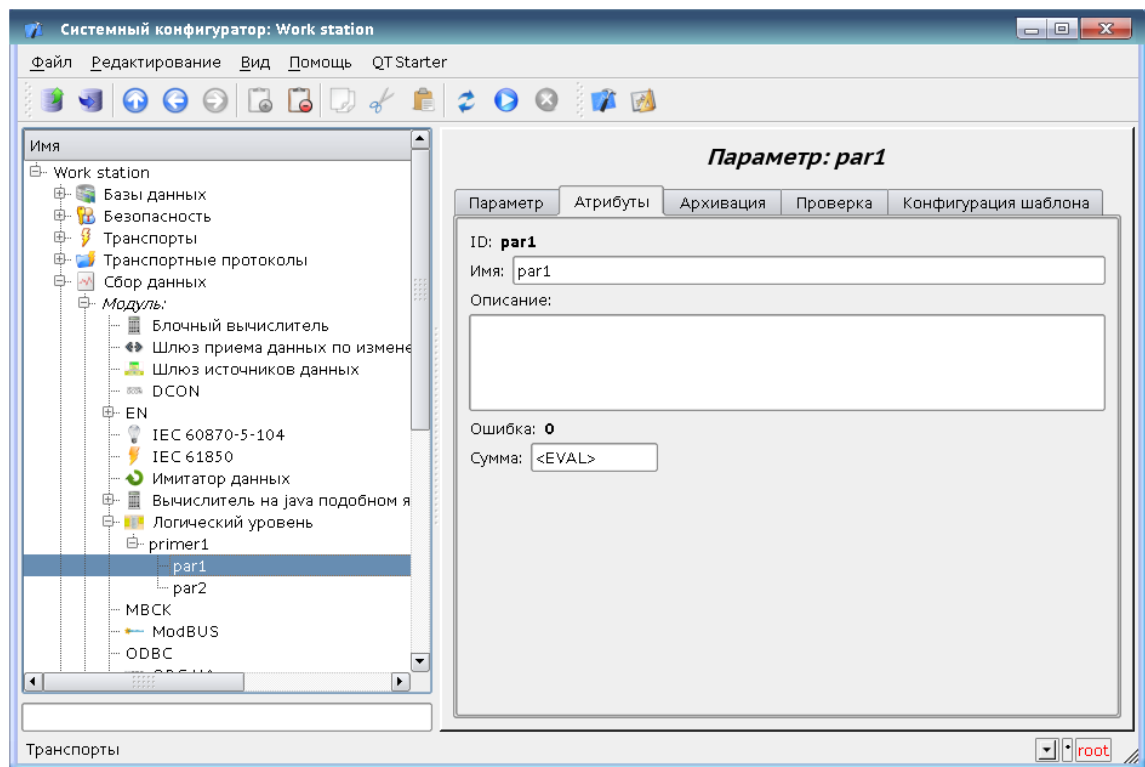


Рисунок 59

Вкладка "Архивация" (рисунок 60) содержит таблицу с атрибутами параметра в строках, и архиваторами в колонках. Пользователь имеет возможность установить архивацию нужного атрибута требуемым архиватором, просто изменив ячейку на пересечении. Например, чтобы добавить архивацию атрибута «NAME» архиватором «DBArch.Arch1» необходимо произвести двойной щелчок мышью в ячейке на пересечении соответствующей колонки и строки (рисунок 61). При этом архиватор «DBArch.Arch1» предварительно должен быть создан в модуле подсистемы «Архивы» (порядок действий в 3.9.4). Затем щелкнуть мышью на появившемся списке и выбрать «True» (рисунок 62). В результате для атрибута «NAME» будет установлена необходимая архивация (рисунок 63).

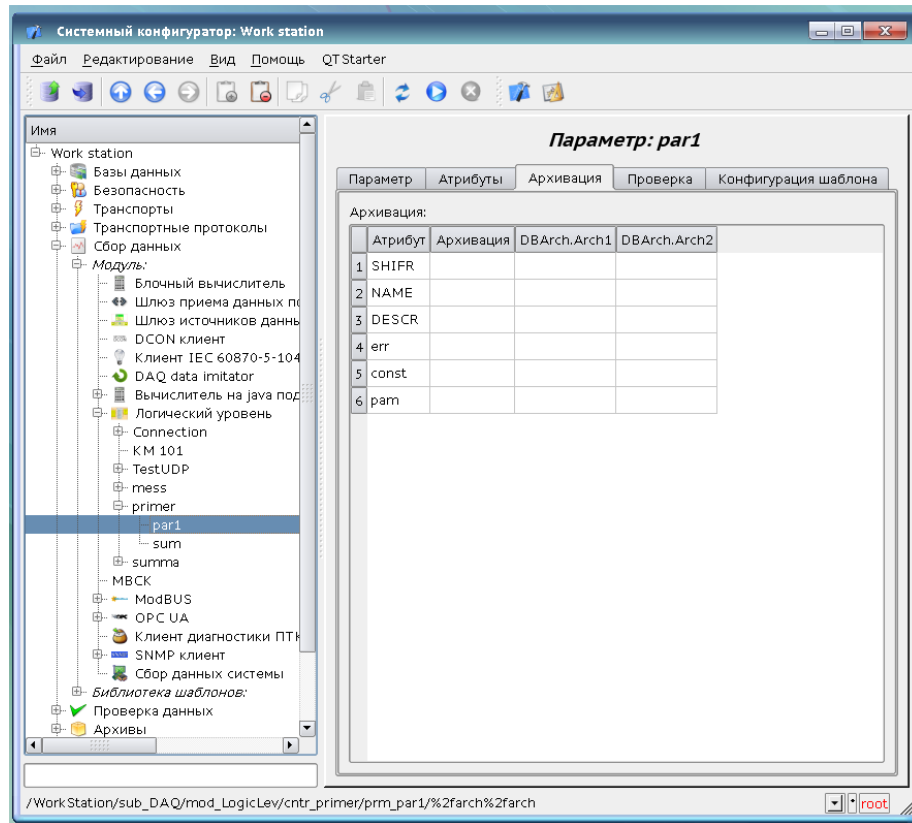


Рисунок 60

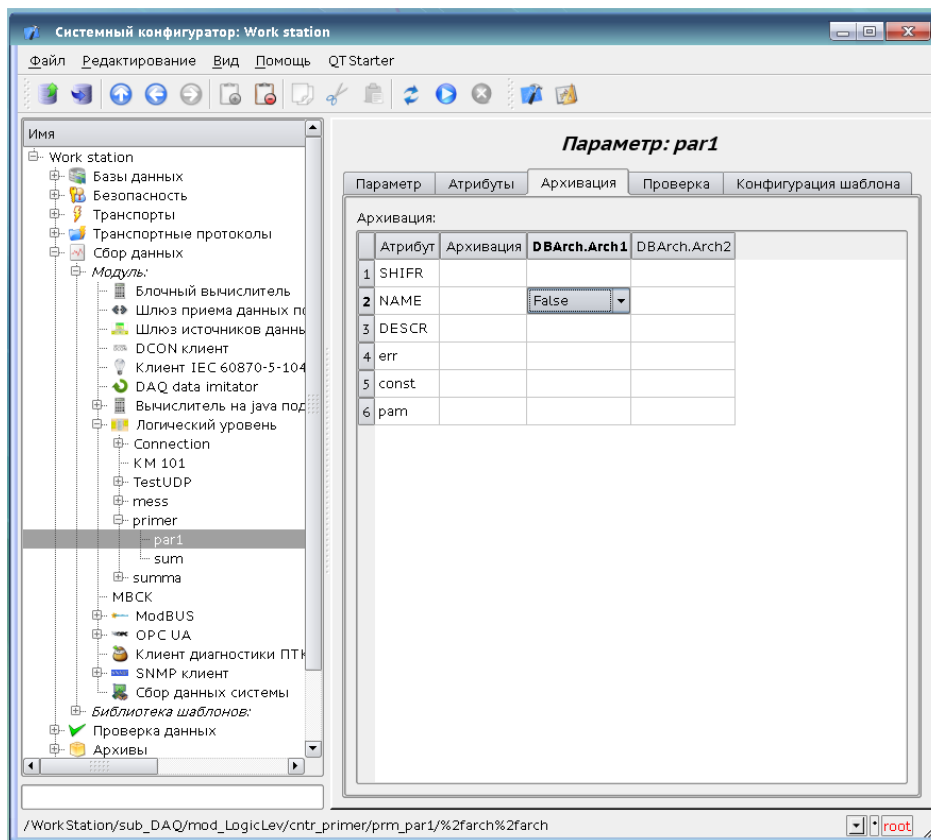


Рисунок 61

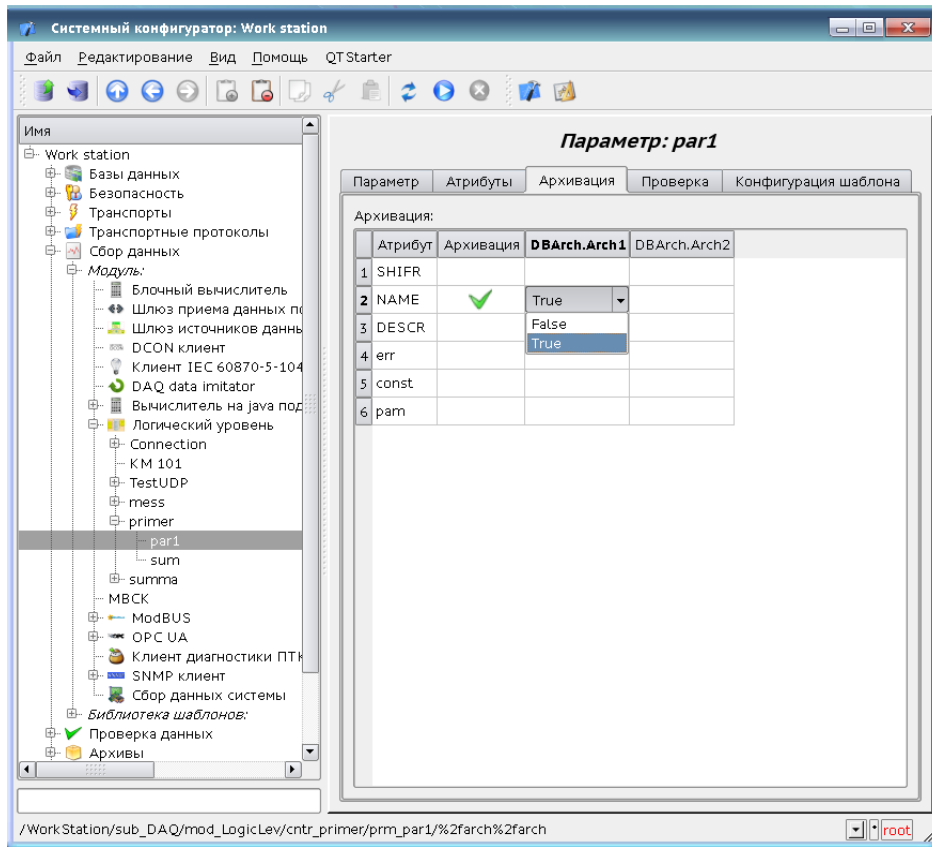


Рисунок 62

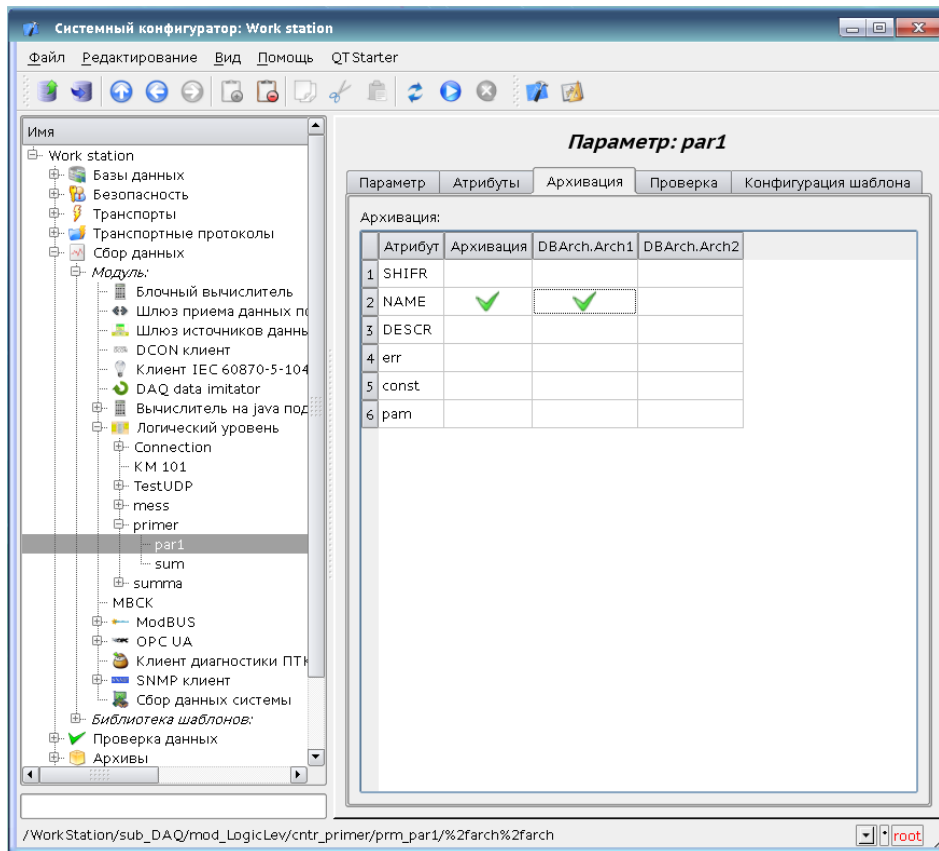


Рисунок 63

Вкладка «Проверка» (рисунок 64) содержит настройки для следующих проверок значений атрибутов:

- выход за границы диапазона;
- коды ошибок от устройства;
- скорость изменения атрибута.

Раздел «Атрибуты» содержит перечень атрибутов, для которых возможно настроить проверки и задать соответствующие сообщения о тревогах, которые будут поступать в журнал тревог и событий.

Раздел «Параметры проверки» предоставляет возможность выбора типа проверки:

- VldBorder – выход за границы диапазона, т.е. задание верхних и нижних границ диапазона и предупредительных и аварийных уставок (рисунок 64);
- VldFromDevice – коды ошибок от устройства (рисунок 65);
- VldSpeed – задание верхней и нижней границ скорости изменения атрибута (рисунок 67).

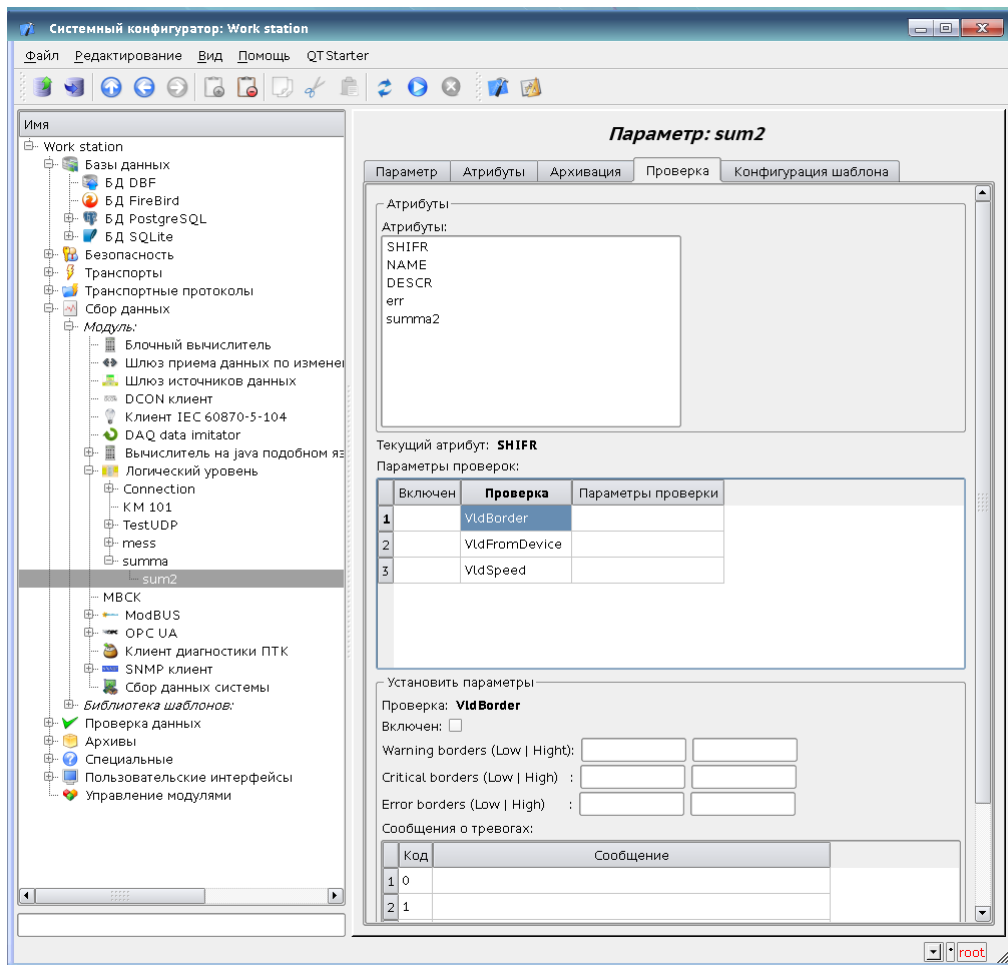


Рисунок 64

При выборе атрибута и типа проверки становятся доступными соответствующие поля настройки для выбранной проверки. Всплывающая подсказка предоставляет формат заполнения выбранного поля (рисунки 65 и 66).

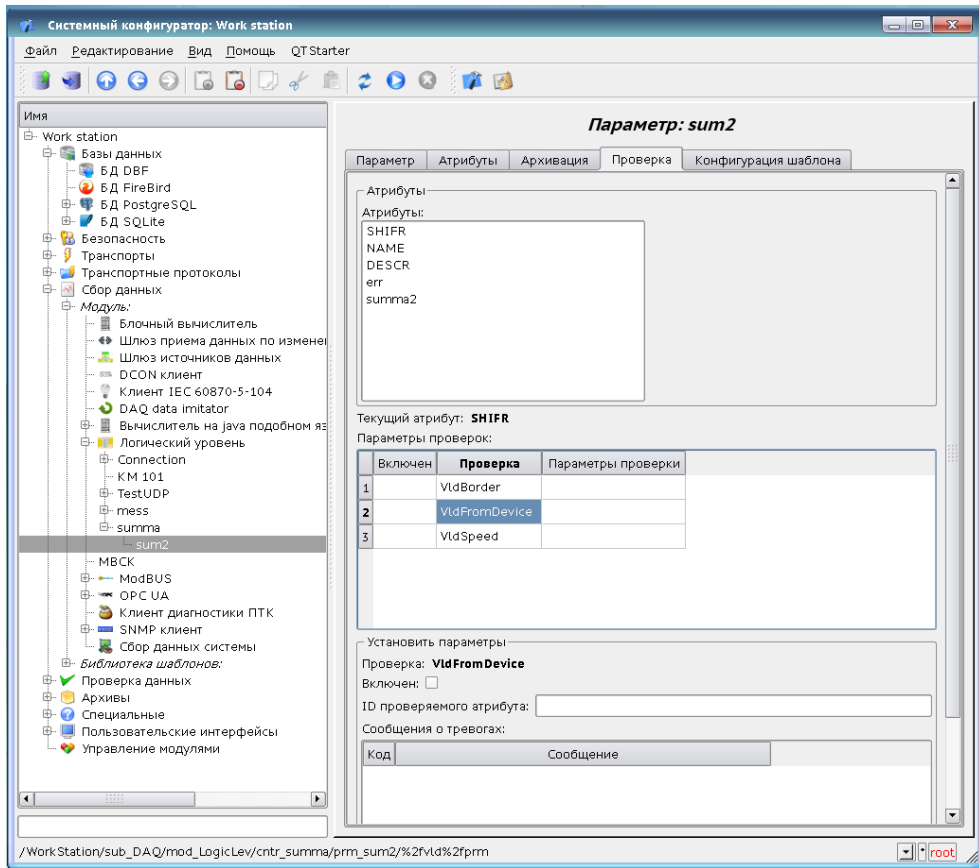


Рисунок 65

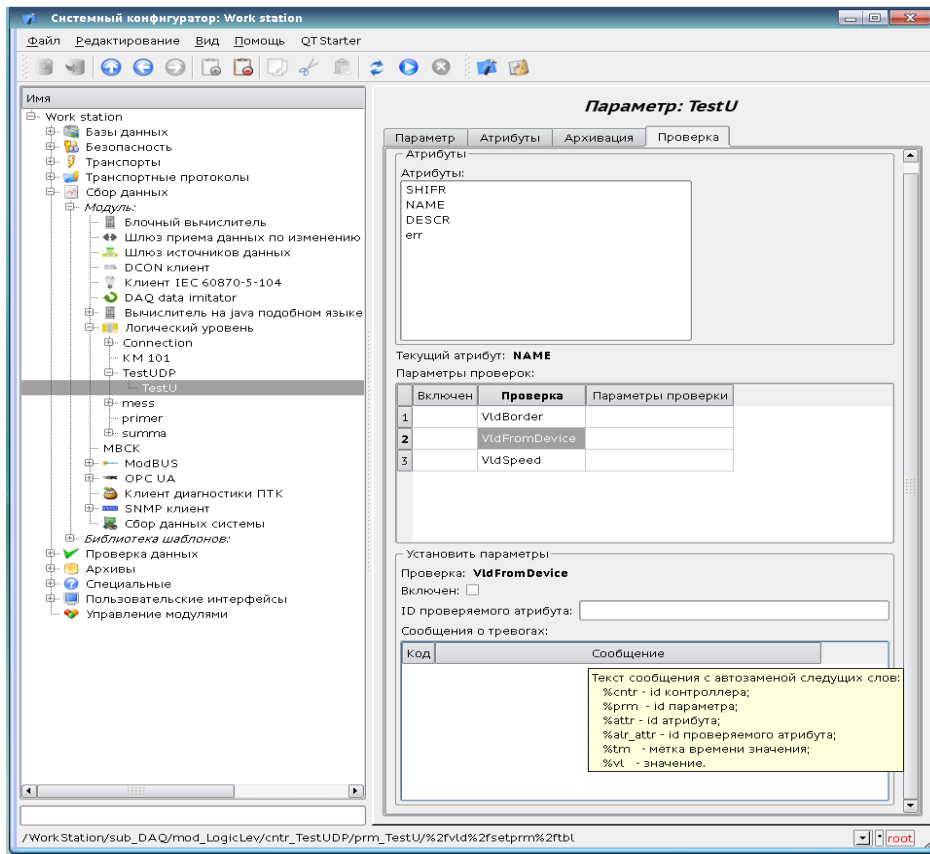


Рисунок 66

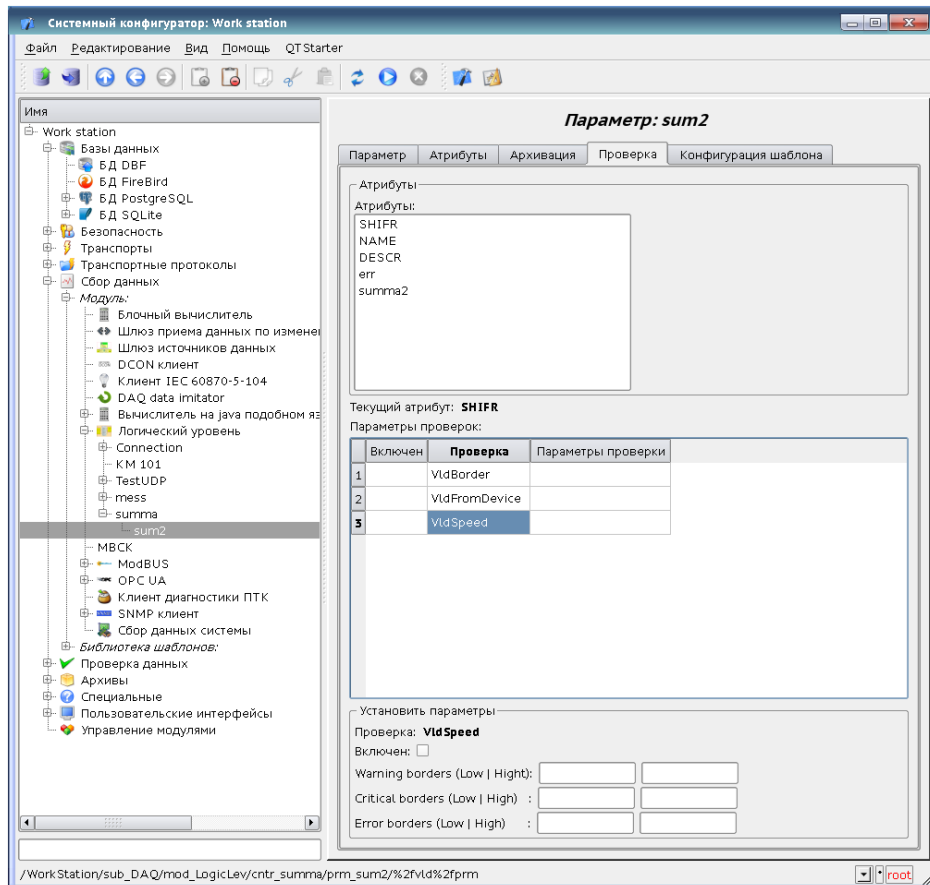


Рисунок 67

Для настройки проверки атрибута на выход за границы диапазона (VldBorder) необходимо задать верхние и нижние значения границ диапазона для предупредительных, критических и ошибочных уставок. Выбор опции «Проверка на EVAL» позволяет системе выдавать сообщение об ошибке при значении атрибута равном <EVAL>. Для настройки сообщений об ошибках необходимо заполнить поле «Сообщение» в таблице «Сообщения о тревогах». Для запуска проверки необходимо установить флаг «Включен» (рисунок 68).

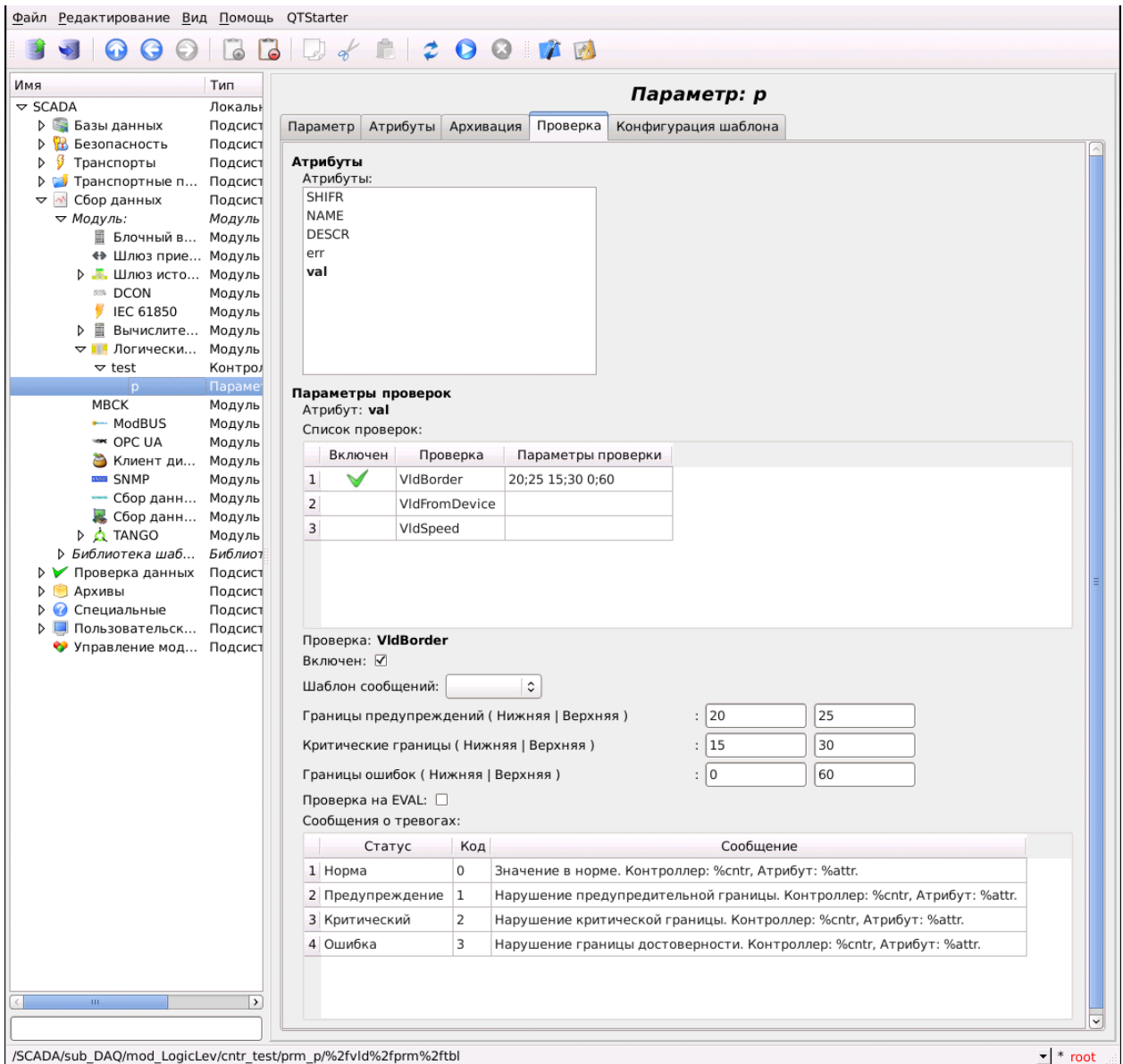


Рисунок 68

Для настройки проверки кодов ошибок от устройства (VldFromDevice) необходимо указать ID проверяемого атрибута. Для преобразования приходящих от устройств значений при необходимости настроить фильтр проверяемого атрибута:

Формат поля: <оператор><фильтр>

Операторы: AND: &, OR: |, NOT: ~, XOR: ^

Далее, указать коды проверяемого атрибута:

Формат поля: [опер1]([опер2][код1;][опер3][код2;][опер4][код-3-код6;])

Операторы: AND: &, OR: |, NOT: ~, XOR: ^, <EVAL>: E

Для проверки атрибута на EVAL необходимо указывать код с оператором E. Если проверяемый атрибут содержит несколько битов ошибки, необходимо выбрать опцию «Битовый код ошибки». Для настройки сообщений об ошибках необходимо в таблице «Сообщения о тревогах» добавить запись (правой кнопкой), выбрать соответствующий оператор, указать код ошибки и сообщение. Для запуска проверки установить флаг «Включен» (рисунок 69).

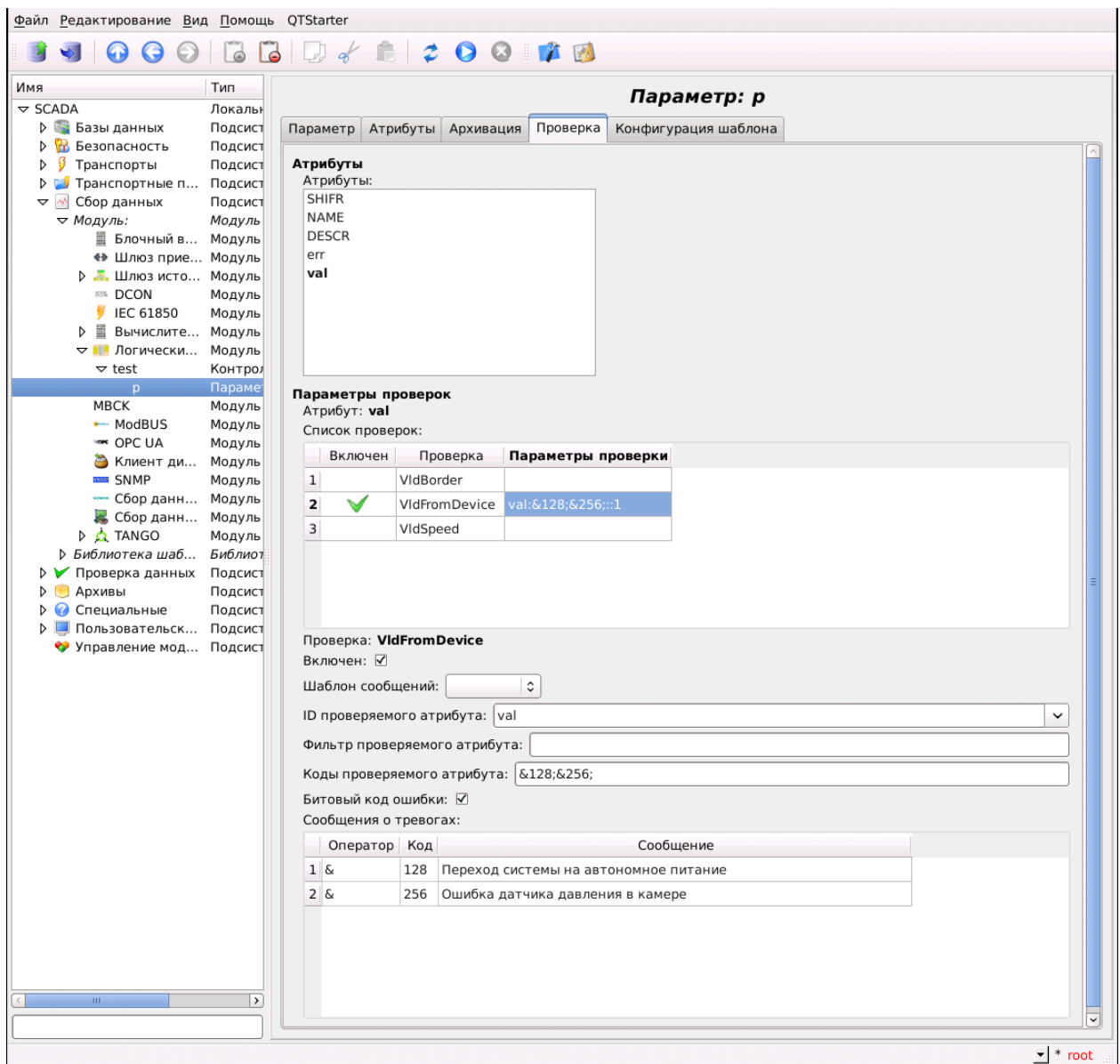


Рисунок 69

Для настройки проверки скорости изменения атрибута (VldSpeed) необходимо задать верхние и нижние границы диапазона для предупредительных, критических и ошибочных уставок. Для настройки сообщений об ошибках необходимо заполнить поле «Сообщение» в таблице «Сообщения о тревогах». Для запуска проверки выбрать опцию «Включен» (рисунок 70).

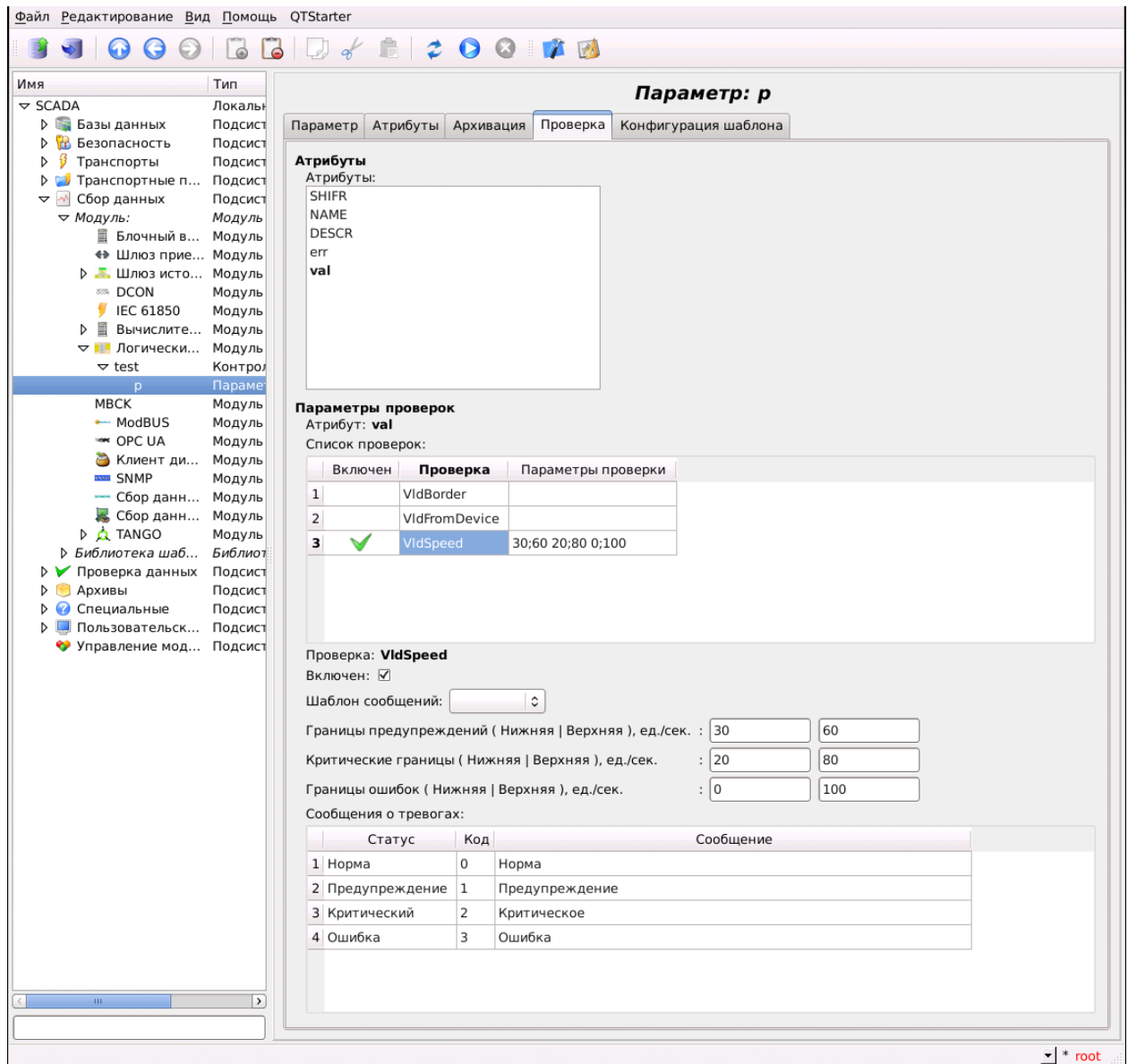


Рисунок 70

Вкладка "Конфигурация шаблона" (рисунок 71) содержит конфигурационные поля в соответствии с шаблоном. В примере это групповая связь на внешний параметр, а именно вычисление суммы двух слагаемых. Эту связь можно установить, указав путь к параметру (выбирается мышью из списка), (рисунок 72). Параметры и атрибуты шаблона настраиваются с помощью Библиотеки шаблонов на вкладке «Ю» (см. рисунок 54).

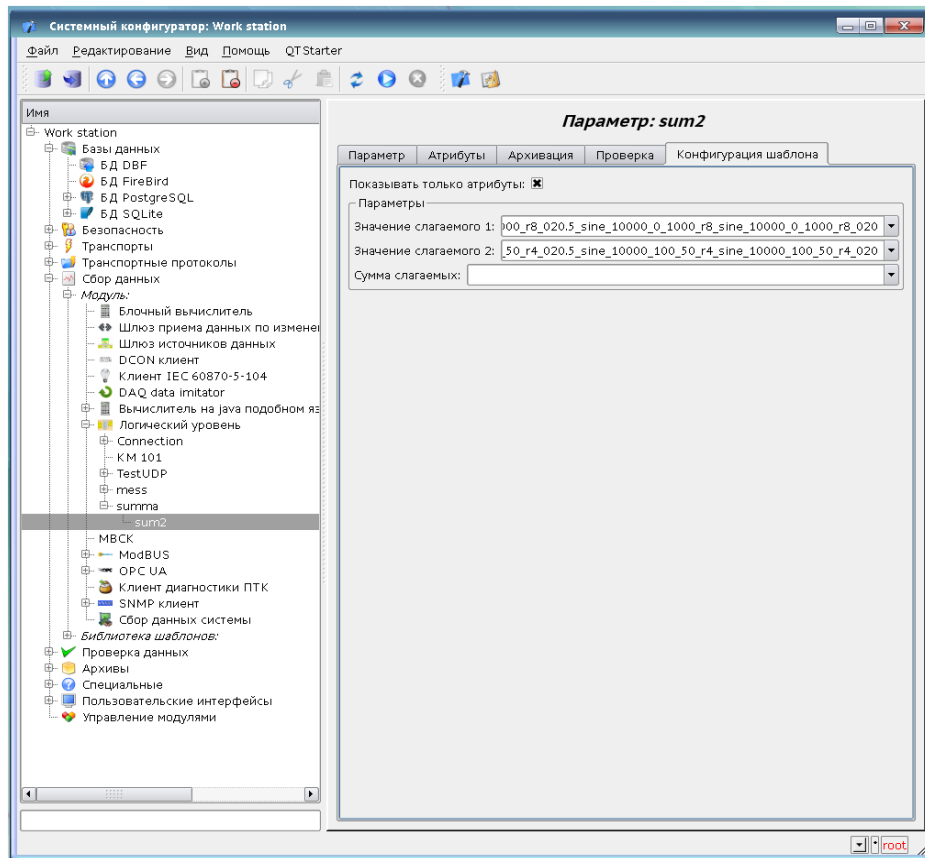


Рисунок 71

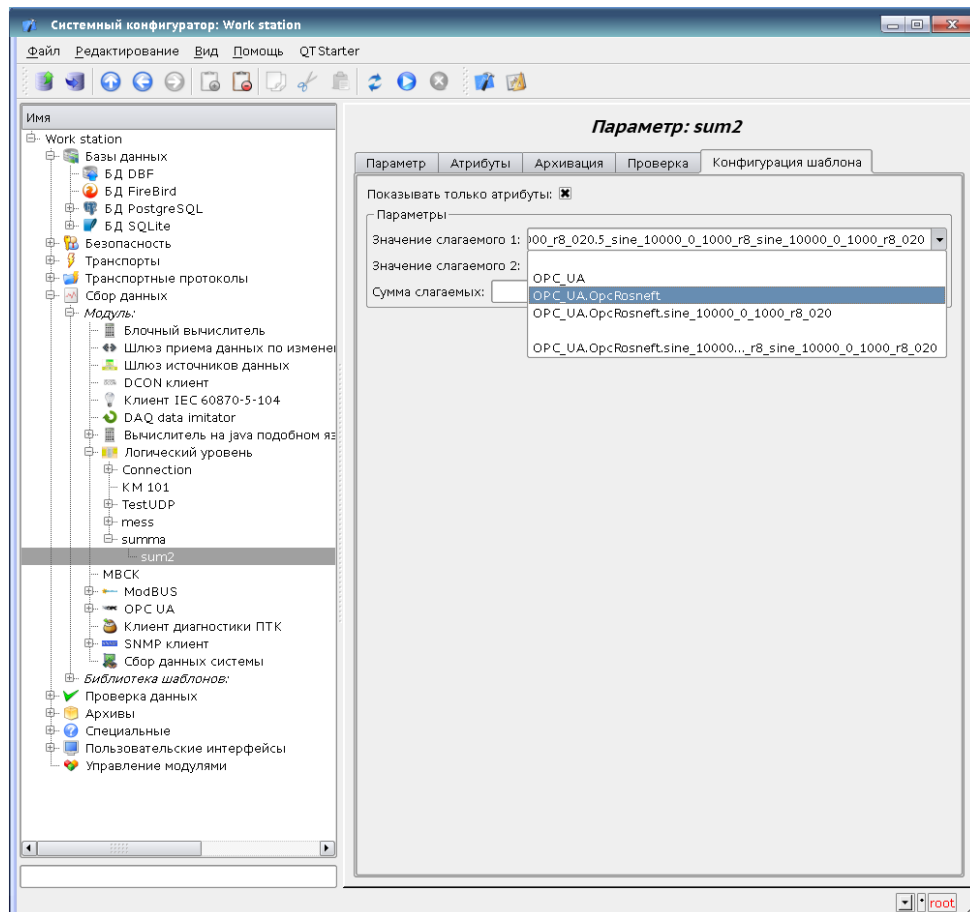


Рисунок 72

3.8 Подсистема «Проверка данных»

Подсистема «Проверка данных» предназначена для создания шаблонов для проверки значений атрибутов модулей подсистемы «Сбор данных» на:

- выход за границы диапазона;
- коды ошибок от устройства;
- скорость изменения атрибута.

Шаблоны проверок можно применять для каждого источника данных подсистемы "Сбор данных" выбрав необходимый в строке "Шаблон сообщений".

Для конфигурации подсистемы предусмотрена корневая страница подсистемы "Проверка данных", содержащая вкладки "Проверки", "Модули" и "Помощь".

Вкладка «Проверки» содержит список ранее созданных проверок, поля для настройки периода проверки данных и установки приоритета задачи проверки данных. В контекстном меню списка проверок предоставляется возможность перехода к нужной проверке. Вид вкладки "Проверки" показан на рисунке 73.

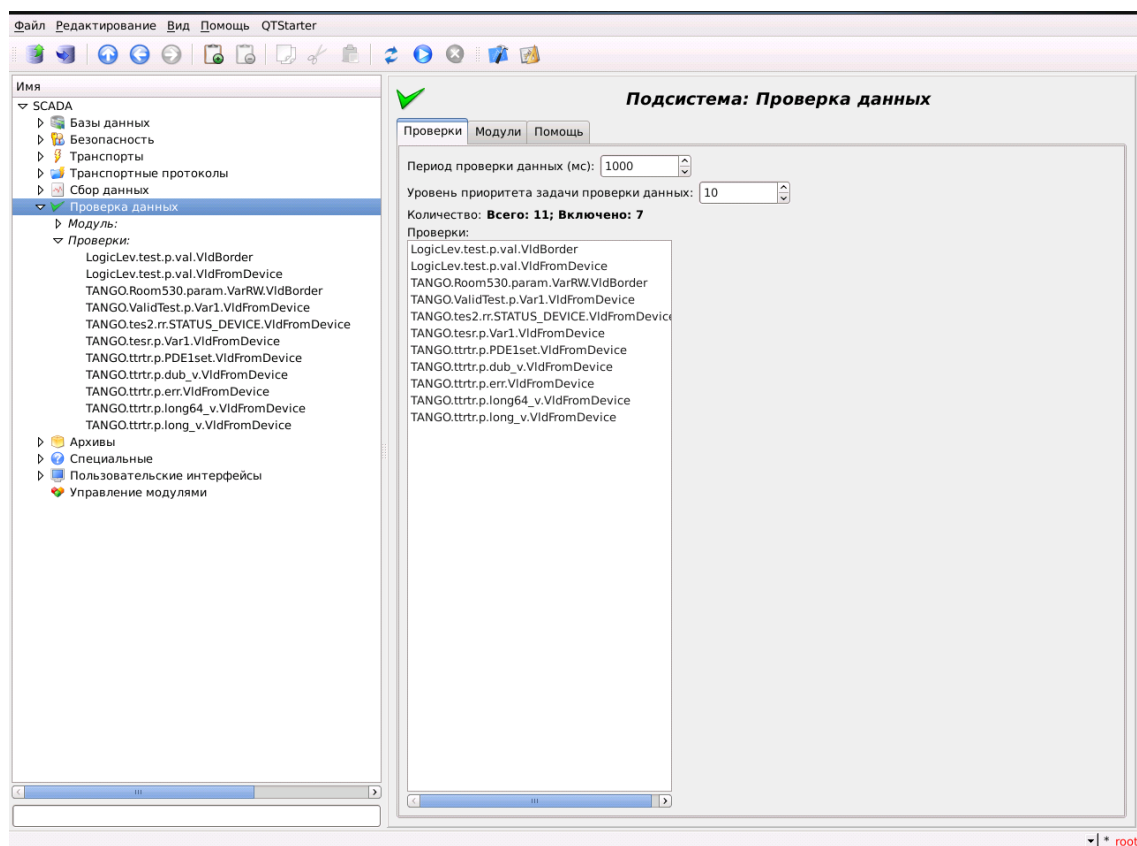


Рисунок 73

Вкладка "Модули" содержит список модулей проверок: проверка значения на границы, проверка значения полученного от устройства, проверка скорости изменения значения

(рисунок 74). Для перехода к контекстному меню списка шаблонов необходимо дважды щелкнуть левой кнопкой мыши по модулю.

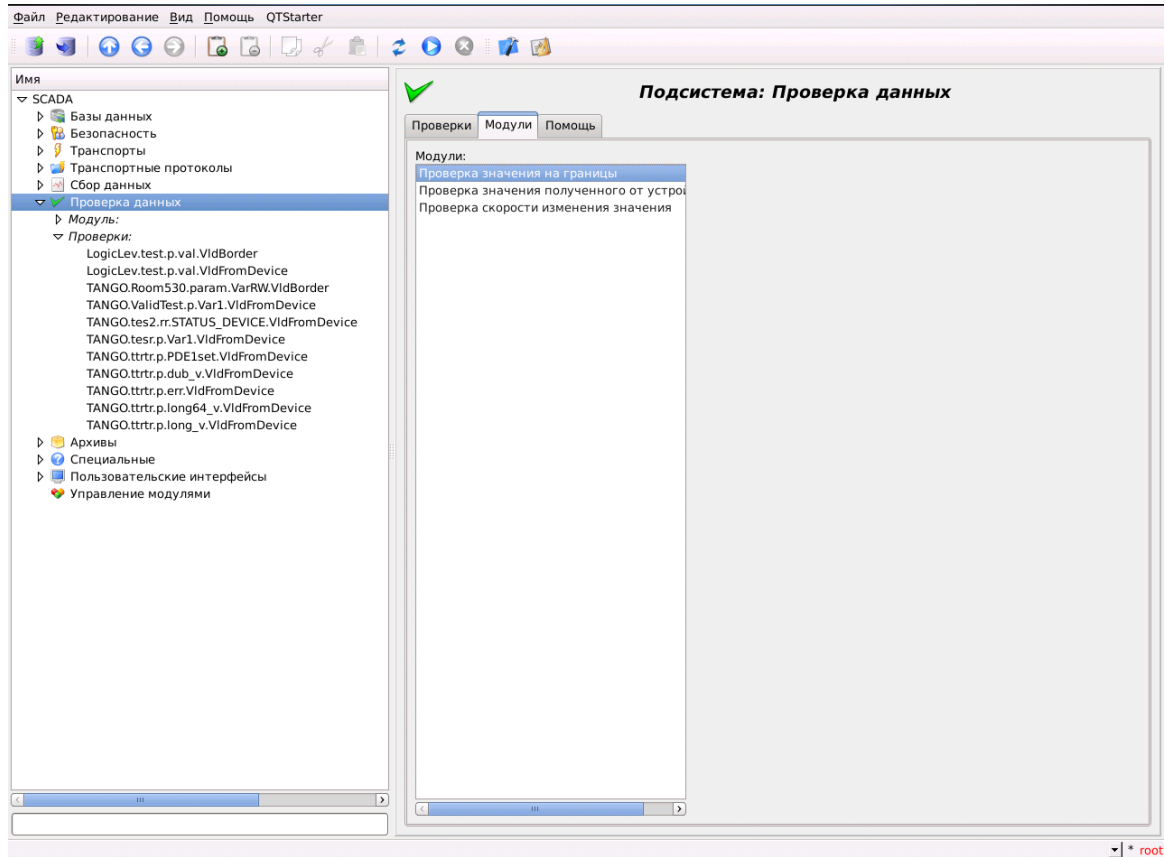


Рисунок 74

В контекстном меню списка шаблонов пользователю предоставляется возможность добавления, удаления и перехода к нужному шаблону сообщений проверок (рисунок 75). Для добавления шаблона нужно щелкнуть правой клавишей мыши в любом месте списка и в появившемся списке выбрать пункт «Добавить». В окне «Установка имени элемента» заполнить поля «ID» и «Имя» (рисунок 76). В результате новый шаблон появится в списке (рисунок 77).

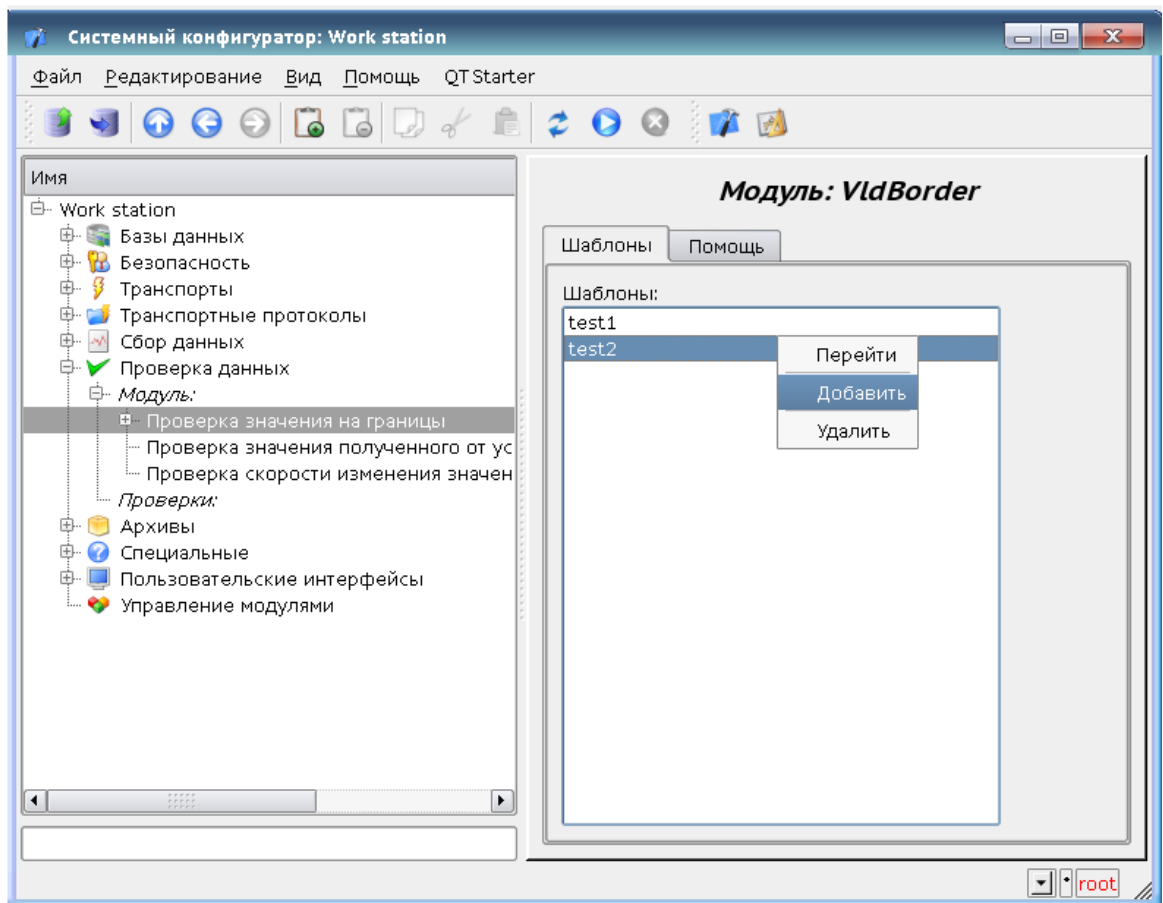


Рисунок 75

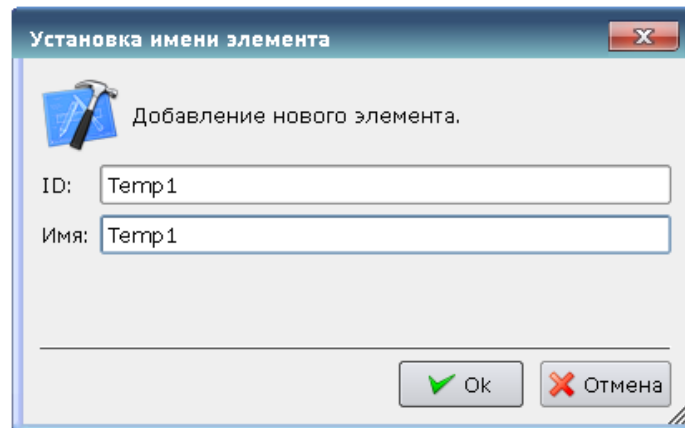


Рисунок 76

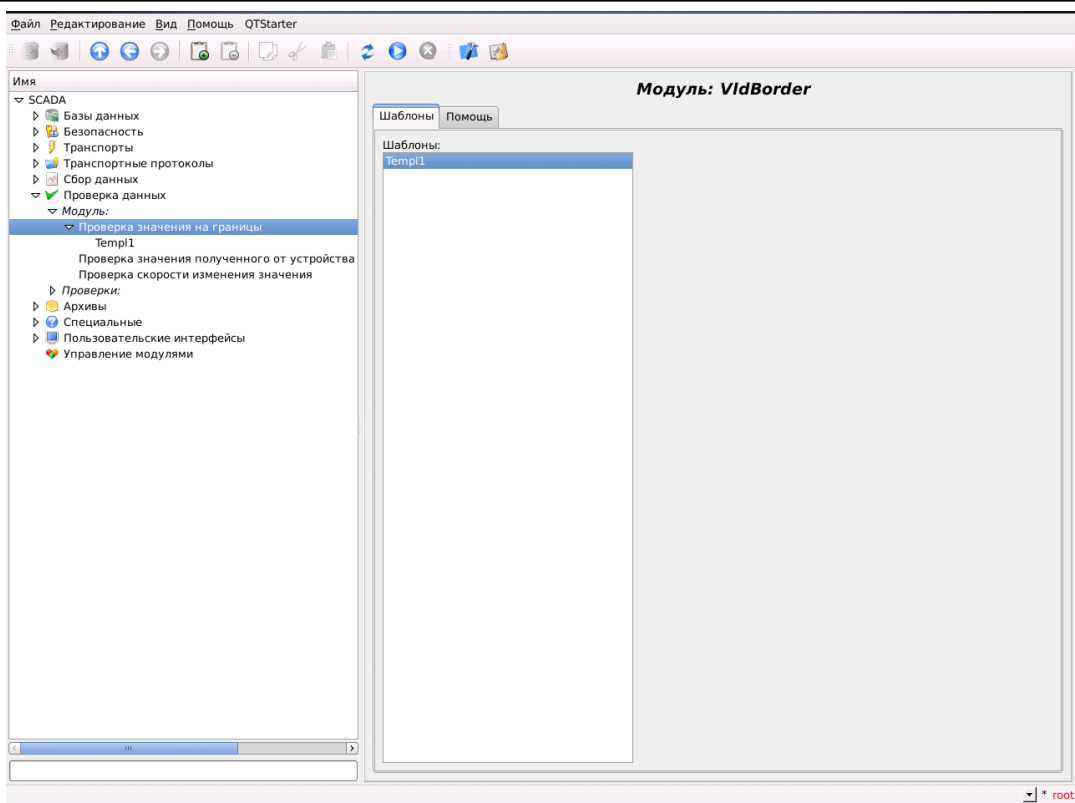


Рисунок 77

При переходе на созданный шаблон «Temp1» (двойной клик левой кнопкой мыши) откроется вкладка шаблон для настройки шаблона сообщения проверки (рисунок 78).

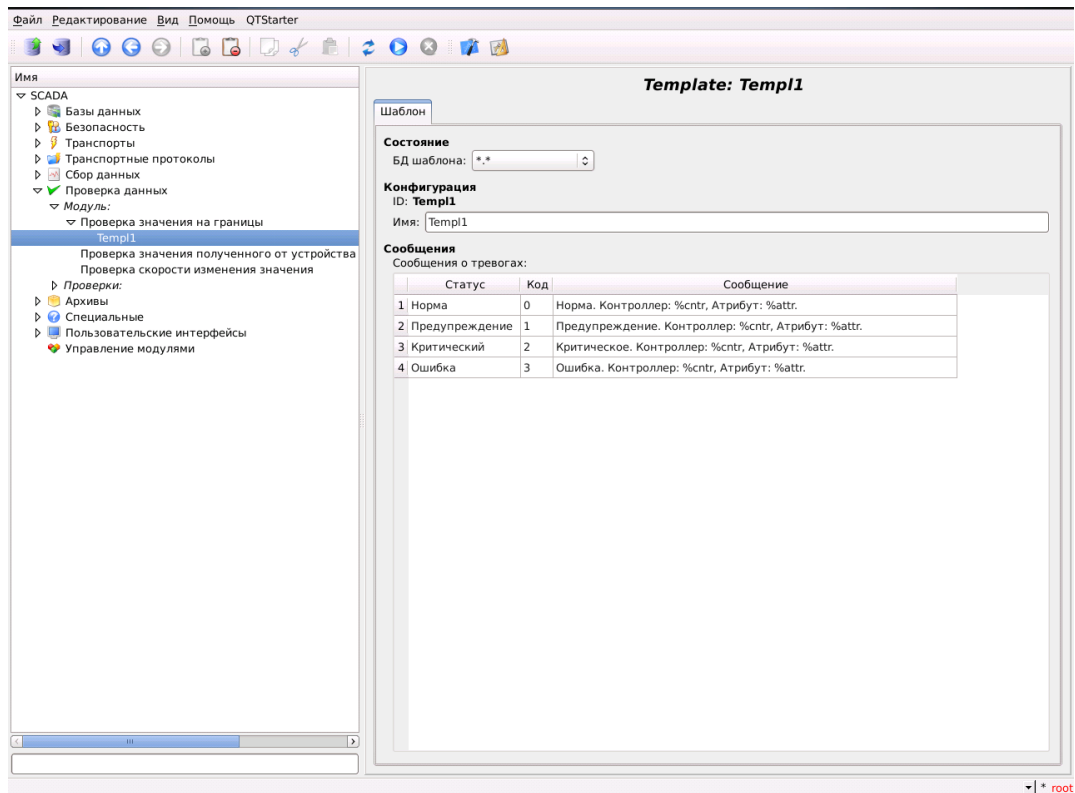


Рисунок 78

3.9 Подсистема «Архивы»

3.9.1 Общие сведения

Для решения задач архивирования потоков данных в СКАДА предусмотрена подсистема "Архивы". Подсистема "Архивы" позволяет вести как архивы сообщений, так и архивы значений. Подсистема "Архивы" является модульной. Модульным объектом, содержащимся в подсистеме "Архивы", выступает тип архиватора. Тип архиватора определяет способ хранения данных, т.е. хранилище (файловая система, СУБД). Каждый модуль подсистемы "Архивы" может реализовывать как архивирование сообщений, так и архивирование значений.

Подсистема "Архивы" может содержать множество архивов, обслуживаемых различными модулями подсистемы.

Сообщение в СКАДА характеризуется датой, уровнем важности, категорией и текстом сообщения. Дата сообщения указывает на время создания сообщения. Уровень важности указывает на степень важности сообщения. Категория определяет адрес или условный идентификатор источника сообщения. Обычно, категория содержит полный путь к источнику сообщения в системе. Текст сообщения несёт смысловую нагрузку сообщения.

В процессе архивирования сообщения пропускаются через фильтр. Фильтр работает по уровню важности и категории сообщения. Уровень сообщения в фильтре указывает, что нужно пропускать сообщения с указанным или более высоким уровнем важности. Для фильтрации по категории применяются шаблоны или регулярные выражения, которые определяют какие сообщения пропускать. Каждый архиватор содержит собственные настройки фильтра.

Архив значений в СКАДА выступает как независимый компонент, который включает буфер обрабатываемый архиваторами. Основным параметром архива значения является источник данных. В роли источника данных могут выступать атрибуты параметров СКАДА, а также другие внешние источники данных (пассивный режим). Другими источниками данных могут быть сетевые архиваторы удалённых СКАДА - систем, среда программирования СКАДА и др.

Ключевым компонентом архивирования значений непрерывных процессов является буфер значений. Буфер значений предназначен для промежуточного хранения массива значений, полученных с определённой периодичностью. Буфер значений используется для непосредственного хранения больших массивов значений в архивах

значений перед непосредственным «сбросом» на физические носители, а также для манипуляций с кадрами значений, т.е. в функциях покадрового запроса значений и их помещения в буфера архивов.

3.9.2 *Архиватор на БД*

Модуль предназначен для архивирования сообщений и значений системы СКАДА на одну из баз данных, поддерживаемых СКАДА.

Архивы условно можно разделить на два типа: архивы сообщений и архивы значений.

Особенностью архивов сообщений является то, что архивируются события. Характерным признаком события является его время возникновения. Архивы сообщений обычно используются для архивирования сообщений в системе, т.е. ведения логов и протоколов. В зависимости от источника сообщения могут классифицироваться по различным критериям. Например, это могут быть протоколы аварийных ситуаций, протоколы действий операторов, протоколы сбоев связи и др.

Особенностью архивов значений является их периодичность, определяемая промежутком времени между двумя смежными значениями. Архивы значений применяются для архивирования истории непрерывных процессов. Поскольку процесс непрерывный, то и архивировать его можно только путём введения понятия квантования времени опроса, поскольку иначе мы получаем архивы бесконечных размеров ввиду непрерывности самой природы процесса.

Подсистема в целом является модульной, что позволяет создавать архивы, основанные на различной природе и способах хранения данных. Данный модуль предоставляет механизм архивирования в БД как для потока сообщений, так и для потока значений.

3.9.2.1 Архиватор сообщений

Архивы сообщений формируются архиваторами. Архиваторов может быть множество с индивидуальными настройками, позволяющими разделять архивирование различных классов сообщений.

Архиватор сообщений этого модуля хранит данные в таблице БД, которая именуется таким образом: DBAMsg_{ArchID}. Где:

- ArchID — идентификатор архиватора сообщений.

Размер таблицы архива может ограничиваться по времени. После превышения лимита старые записи начнут удаляться!

Модулем предоставляются дополнительные параметры настройки процесса архивирования. У данного модуля таких параметров всего один, и он определяет размер архива по времени.

Таблица БД архиватора сообщений имеет следующую структуру: {TM, TMU, CATEG, MESS, LEV}. Где:

- TM — UTC время сообщения, секунды от эпохи (01.01.1970). В БД, содержащих специализированный тип для хранения даты и времени, может использоваться этот специализированный тип;
- TMU — микросекунды времени;
- CATEG — категория сообщения;
- MESS — текст сообщения;
- LEV — уровень сообщения.

3.9.2.2 Архиватор значений

Архивы значений формируются архиваторами значений индивидуально для каждого источника. Архиваторов может быть множество с индивидуальными настройками, позволяющими разделить архивы по различным характеристикам, например по точности и глубине.

Архив значений является независимым компонентом, который включает буфер, обрабатываемый архиваторами. Основным параметром архива значения является источник данных. В роли источника данных могут выступать атрибуты параметров СКАДА, а также другие внешние источники данных (пассивный режим). Источниками данных также могут быть: сетевые архиваторы удалённых СКАДА систем, среда программирования системы СКАДА и др. Не менее важными параметрами архива являются параметры его буфера. Буфер создается свой для каждого источника данных. От параметров буфера зависит возможность работы архиваторов. Так, периодичность опроса буфера архиватором не должна быть менее периодичности поступления данных в буфер от источника. А размер должен выбираться в зависимости от периодичности опроса архиватором, для исключения переполнения буфера. В противном случае возможны потери данных!

Архиватор значений этого модуля хранит данные в таблице БД, которая именуется таким образом: DBAV1_{ArchID}_{ArchiveID}. Где:

- ArchID — Идентификатор архиватора значений;

- ArchiveID — Идентификатор архива значений.

Размер таблицы архива может ограничиваться по времени. После превышения лимита старые записи начнут удаляться!

Модулем предоставляются дополнительные параметры настройки процесса архивирования. У данного модуля таких параметров всего один, и он определяет размер архива по времени.

Таблица БД архиватора значений имеет следующую структуру: {TM, TMU, VAL}.

Где:

- TM — UTC время значения, секунды от эпохи (01.01.1970). В БД, содержащих специализированный тип для хранения даты и времени, может использоваться этот специализированный тип;
- TMU — Время значения, микросекунды;
- VAL — Значение, тип значения определяет тип данной колонки.

Для хранения начала, конца архива и иной информации об архиве в архивных таблицах создаётся информационная таблица с именем данного модуля: «DBArch». Данная таблица имеет структуру: {TBL, BEGIN, END, PRM1, PRM2, PRM3}. Где:

- TBL — Имя таблицы архива;
- BEGIN — Начало данных в архиве;
- END — Конец данных в архиве;
- PRM1 — Дополнительный параметр 1;
- PRM2 — Дополнительный параметр 2;
- PRM3 — Дополнительный параметр 3.

3.9.3 *Архиватор на ФС*

Данный модуль предоставляет механизм архивирования на файловую систему, как для потока сообщений, так и для потока значений.

3.9.3.1 Архиватор сообщений

Архивы сообщений формируются архиваторами. Архиваторов может быть множество с индивидуальными настройками, позволяющими разделять архивирование различных классов сообщений.

Архиватор сообщений этого модуля позволяет хранить данные как в файлах формата языка XML, так и в формате плоского текста. Язык разметки XML является стандартным

форматом, который с лёгкостью понимают многие сторонние приложения. Однако открытие и разбор файлов в таком формате требует значительных ресурсов. С другой стороны, формат плоского текста требует значительно меньше ресурсов, хотя и не является унифицированным, а также требует знания его структуры для разбора.

В любом случае, поддерживаются оба формата, и пользователь может выбрать любой из них, в соответствии со своими требованиями.

Файлы архивов именуются архиваторами, исходя из даты первого сообщения в архиве. Например так: <2006-06-21 17:11:04.msg>. Файлы архивов могут ограничиваться по размеру и времени. После превышения лимита создаётся новый файл. Максимальное количество файлов в директории архиватора также может быть ограничено. После превышения лимита на количество файлов, старые файлы начнут удаляться!

С целью оптимизации использования дискового пространства архиваторы поддерживают упаковку старых архивов упаковщиком gzip. Упаковка производится после продолжительного неиспользования архива.

При использовании архивов в формате языка XML соответствующие файлы загружаются целиком! Для выгрузки неиспользуемых продолжительное время архивов применяется таймаут доступа к архиву после превышения которого, архив выгружается из памяти, а затем и пакуется.

В число этих параметров входят:

- выбор XML-формата архивных файлов;
- максимальный размер одного файла архива;
- максимальное количество архивных файлов;
- ограничение размера файла архива по времени;
- таймаут упаковки файлов архива;
- периодичность проверки файлов архиватором на предмет поиска новых архивов и удаление старых;
- включение использования информационных файлов для упакованных файлов;
- команда немедленной проверки директории архиватора. Может использоваться при размещении в директории архиватора файлов архивов другой станции.

3.9.3.2 Формат файлов архива сообщений

В таблице ниже приведен синтаксис файла архива, построенного на XML-языке:

Таблица 4

Тег	Описание	Атрибуты	Содержит
FSArch	Корневой элемент. Идентифицирует файл как принадлежащий данному модулю.	Version — версия файла архива; Begin — время начала архива (hex – UTC в секундах от 01/01/1970); End — время окончания архива (hex – UTC в секундах от 01/01/1970).	(m)
m	Тег отдельного сообщения.	tm — время создания сообщения (hex – UTC в секундах от 01/01/1970); tmu — микросекунды времени сообщения; lv — уровень сообщения; cat — категория сообщения.	Текст сообщения

Архивный файл на основе плоского текста состоит из:

- заголовка в формате: [FSArch <vers> <charset> <beg_tm> <end_tm>]

Где:

- <vers> — версия модуля архивирования;
- <charset> — кодировка файла (обычно UTF8);
- <beg_tm> — UTC время начала архива с эпохи 01.01.1970 в шестнадцатеричной форме;
- <end_tm> — UTC время конца файла архива с эпохи 01.01.1970 в шестнадцатеричной форме.

- записей сообщений в формате: [<tm> <lev> <cat> <mess>]

Где:

- <tm> — время сообщения в виде: <utc_sec:usec>, где:
- utc_sec — UTC время с эпохи 01.01.1970 в шестнадцатеричной форме;
- usec — микросекунды времени в десятичной форме;
- <lev> — уровень важности сообщения;
- <cat> — категория сообщения;
- <mess> — текст сообщения.

Текст сообщения и категория кодируются с целью исключения символов разделителей (символ пробела).

3.9.3.3 Архиватор значений

Архивы значений формируются архиваторами значений индивидуально для каждого зарегистрированного архива. Архиваторов может быть множество с индивидуальными настройками, позволяющими разделить архивы по различным параметрам, например, по точности и глубине.

Файлы архивов именуются архиваторами, исходя из даты первого значения в архиве и идентификатора архива. Например, таким образом: <MemInfo_use 2006-06-17 17:32:56.val>. Файлы архивов могут ограничиваться по времени. После превышения лимита создаётся новый файл. Максимальное количество файлов в директории архиватора также может ограничиваться. После превышения лимита на количество файлов старые файлы начнут удаляться!

С целью экономии дискового пространства архиваторы поддерживают упаковку в дополнении к последовательной упаковке старых архивов упаковщиком gzip. Упаковка производится после продолжительного неиспользования архива. Для обеспечения возможности быстрого подключения больших архивов к другой системе можно включить использование информационных файлов для упакованных файлов, что предотвратит предварительную распаковку всех файлов на другой системе.

3.9.3.4 Формат файлов архива значений

Для реализации архивирования на файловую систему предъявляются следующие требования:

- быстрый (простой) доступ на добавление в архив и чтение из архива;
- возможность изменения значений в существующем архиве (с целью заполнения дыр в дублированных системах);
- цикличность (ограничение размера);
- возможность сжатия методом упаковки последовательности одинаковых значений, сохраняющим возможность быстрого доступа (последовательная упаковка);
- возможность упаковки устаревших данных стандартными архиваторами (gzip, bzip2 ...) с возможностью распаковки при обращении.

В соответствии с вышеизложенными требованиями организовано архивирование методом множественности файлов (для каждого источника). Цикличность архива реализуется на уровне файлов, т.е. создается новый файл, а самый старый удаляется. Для быстрого сжатия используется метод притягивания к последнему одинаковому значению.

Для этих целей в файле архива предусматривается битовая таблица упаковки размером один в один с количеством хранимых данных. Т.е. каждый бит соответствует одному значению в архиве. Значение бита указывает на наличие значения. Для потока одинаковых значений биты обнулены. В случае с архивом строк таблица является не битовой, а байтовой и содержит длину соответствующего значения. В случае поступления потока одинаковых значений, длина будет нулевой, и читаться будет первое одинаковое значение. Поскольку таблица байтовая, то архив сможет хранить строки длиной не более 255 символов. Таким образом, методики хранения можно разделить на методику данных фиксированного и нефиксированного размера. Общая структура файла архива приведена на рисунке 80.

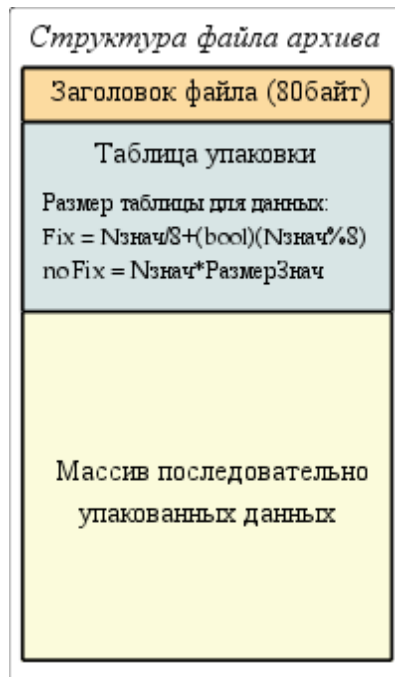


Рисунок 80

При создании нового файла архива формируется: заголовок (структура заголовка в таблице 5), нулевая битовая таблица упаковки архива и первое недостоверное значение. Таким образом, получится архив, инициализированный недостоверными значениями. В дальнейшем новые значения будут вставляться в область значений с корректировкой индексной таблицы упаковки. Из этого следует, что пассивные архивы будут вырождаться в файлы размером в заголовок и битовую таблицу.

Таблица 5

Поле	Описание	Размер, байт (бит)
f_tp	Системное имя архива («SCADA Val Arch.»)	20
archive	Имя архива, которому принадлежит файл.	20
beg	Время начала архивных данных (мкс)	8

Поле	Описание	Размер, байт (бит)
end	Время конца архивных данных (мкс)	8
period	Периодичность архива (мкс)	8
vtp	Тип значения в архиве (Логический, Целый, Вещественный, Строка)	(3)
hgrid	Признак использования жёсткой сетки в буфере архива	(1)
hres	Признак использования времени высокого разрешения (мкс) в буфере архива	(1)
reserve	Резерв	14
term	Символ окончания заголовка архива (0x55)	1

Разъяснение механизма последовательной упаковки приведено на рисунке 81. Как можно видеть из рисунка, признак упаковки содержит длину (не фиксированные типы) или признак упаковки (фиксированные типы) отдельно взятого значения. Это значит, что для получения смещения нужного значения необходимо сложить длины всех предыдущих действительных значений.

Выполнение данной операции каждый раз и для каждого значения является крайне трудоёмкой операцией. Поэтому был внедрён механизм кеширования смещений значений. Механизм кеширует смещения значений через predetermined их количество, а также кеширует смещение последнего значения к которому производился доступ (отдельно на чтение и запись).

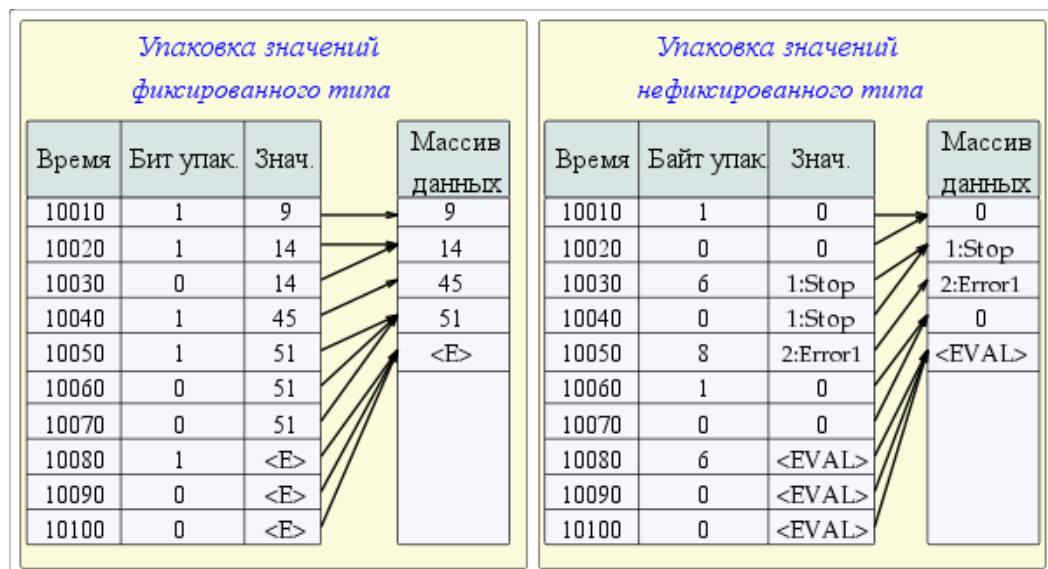


Рисунок 81

Изменения значений внутри существующего архива также предусмотрено. Однако, учитывая необходимость выполнения сдвига хвоста архива, рекомендуется выполнять эту операцию как можно реже и как можно большими блоками.

3.9.4 Конфигурирование подсистемы «Архивы»

Для конфигурации подсистемы предусмотрена корневая страница подсистемы "Архивы", содержащая вкладки "Архив сообщений", "Архивы значений", "Модули" и "Помощь".

Для получения доступа на модификацию объектов этой подсистемы необходимы права пользователя в группе "Archive" или права привилегированного пользователя.

Вид вкладки "Архив сообщений" показан на рисунке 82.

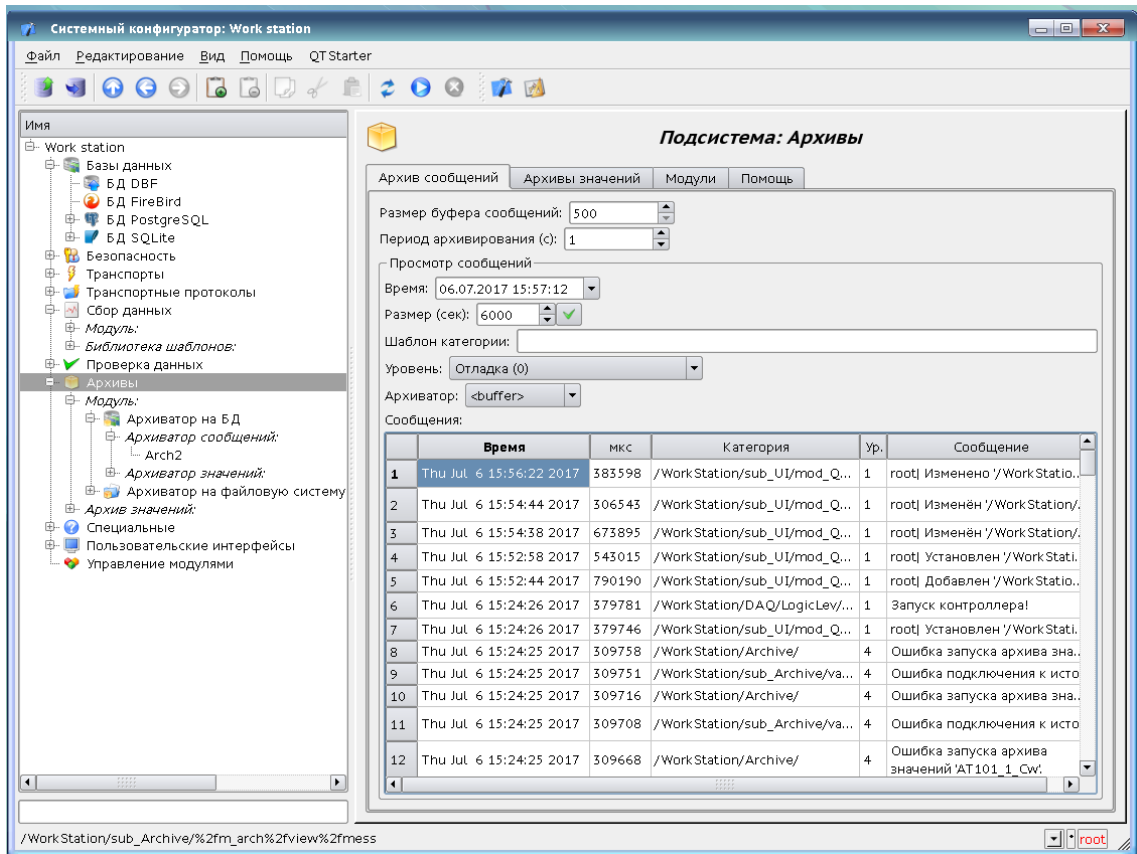


Рисунок 82

Вкладка "Архив сообщений" содержит конфигурацию архива сообщений и форму запроса сообщений из архива.

Конфигурация архива сообщений представлена полями:

- *Размер буфера сообщений* - указывает на размерность области оперативной памяти, зарезервированной под промежуточный буфер сообщений. Сообщения из буфера запрашиваются для просмотра и архивируются архиваторами сообщений;

- *Период архивирования сообщений (с)* - периодичность, с которой архиваторы выбирают сообщения из буфера для их архивирования.

Форма запроса сообщений содержит конфигурационные поля запроса и таблицу результата.

Конфигурационные поля запроса:

- *Время* - указывает время запроса;
- *Размер (сек)* - указывает размер или глубину запроса в секундах;
- *Шаблон категории* - указывает категорию запрошенных сообщений. В категории можно указывать элементы выборки по шаблону, а именно символы "*" - для любой подстроки и "?" - для любого символа;
- *Уровень* - указывает на минимальный уровень сообщений, т.е. запрос будет обработан для сообщений с уровнем, более или равным указанному;
- *Архиватор* - указывает архиватор сообщений, для которого обрабатывать запрос. Если значение отсутствует, то запрос будет обработан для буфера и всех архиваторов. Если указано <buffer>, то запрос будет обработан только для буфера сообщений.

Таблица результата содержит строки сообщений с колонками:

- *Время* - время сообщения;
- *Категория* - категория сообщения;
- *Уровень* - уровень сообщения;
- *Сообщение* - текст сообщения.

Вид вкладки "Архивы значений" показан на рисунке 83.

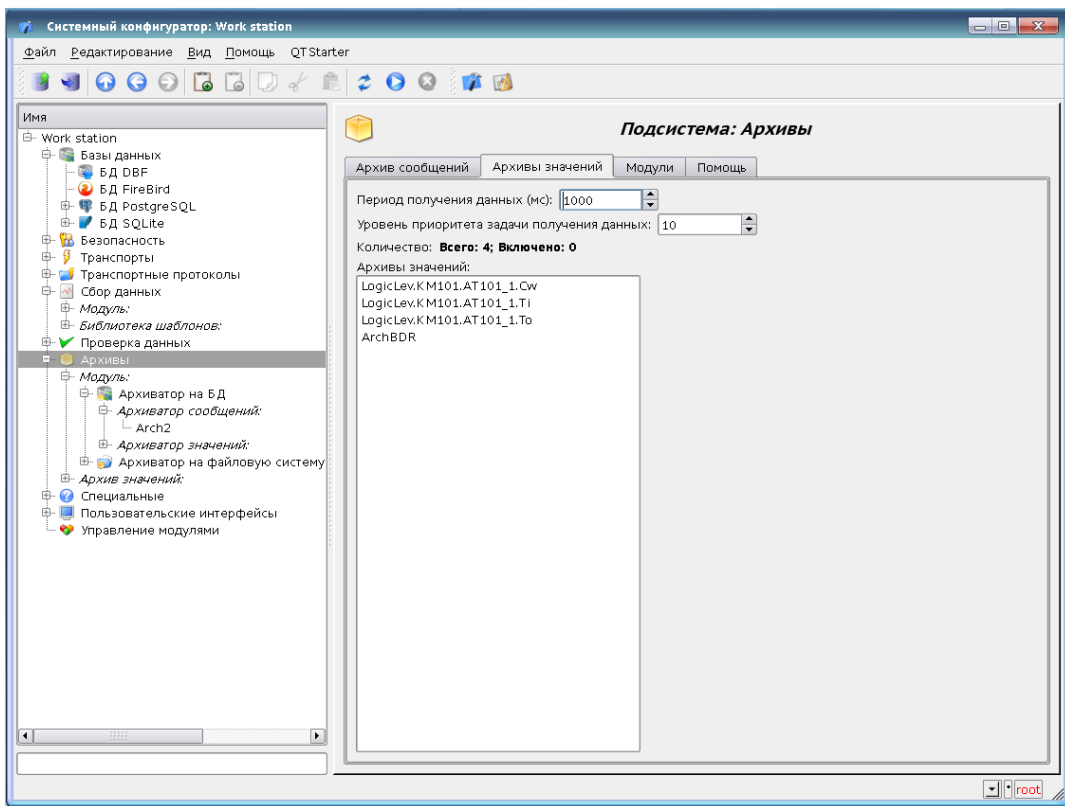


Рисунок 83

Вкладка "Архивы значений" содержит общую конфигурацию архивирования значений и список архивов значений. В контекстном меню списка значений пользователю

предоставляется возможность добавления, удаления и перехода к нужному архиву. Общая конфигурация архивирования представлена полями:

- *Период получения данных (мс)* - указывает периодичность задачи активного архивирования. Фактически, максимальная детализация или минимальный период, активных архивов определяется этим значением.

- *Уровень приоритета задачи получения данных* - устанавливает приоритетность задачи активного архивирования. Используется при планировании задач операционной системой. В случае исполнения станции от имени суперпользователя "root" это поле включает планирование задачи архивирования в режиме реального времени и с указанным приоритетом.

- *Архивы значений* - список имен архивов значений, которые используются в системе.

Вкладка "Модули" содержит список модулей подсистемы "Архивы" и идентична для всех модульных подсистем. Вкладка "Помощь" содержит краткую помощь для данной страницы.

Архив значений подсистемы "Архивы" предоставляет конфигурационную страницу с вкладками "Архив", "Архиваторы" и "Значения".

Вид вкладки "Архив" показан на рисунке 84.

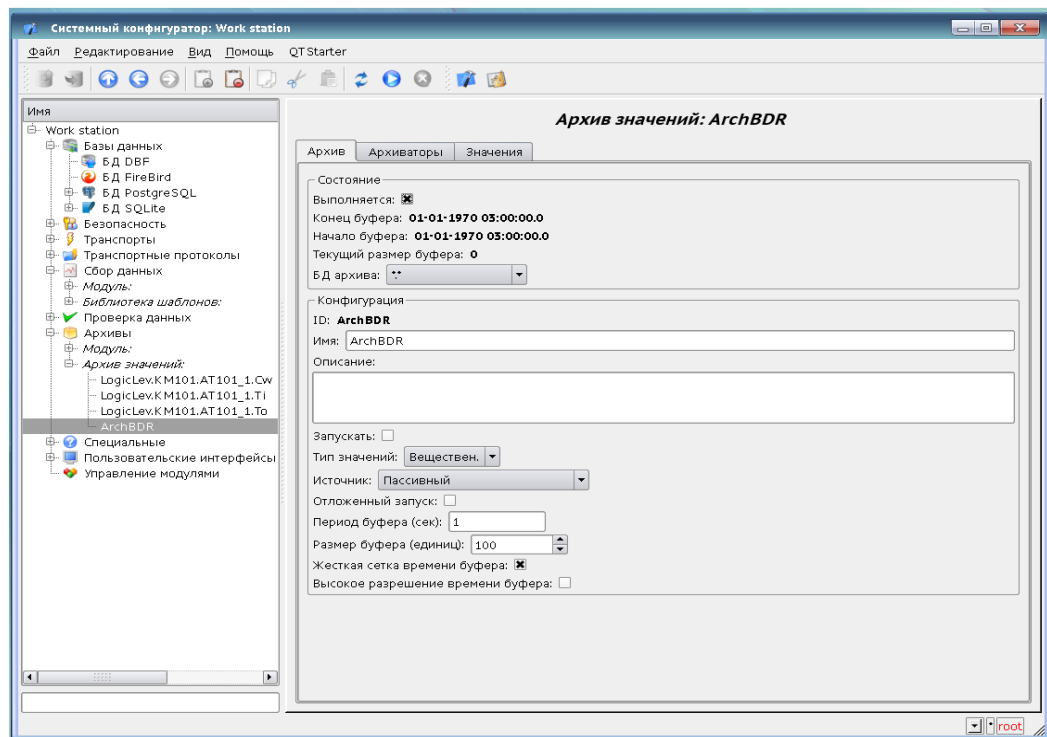


Рисунок 84

Вкладка "Архив" содержит основные настройки архива в составе:

- Раздел "Состояние" - содержит свойства, характеризующие состояние архива:

- *Выполняется* - состояние параметра "Выполняется". Исполняющийся архив собирает данные в буфер и обслуживается архиваторами;

- *БД архива* - адрес БД для хранения данных архива, выбирается из списка щелчком мыши.

- Раздел "Конфигурация" - непосредственно содержит поля конфигурации:

- *ID* - информация об идентификаторе архива;

- *Имя* - указывает имя архива;

- *Описание* - краткое описание архива и его назначения;

- *Запустить* - указывает на состояние "Выполняется", в которое переводить архив при загрузке;

- *Тип значений* - указывает на тип значений, хранящихся в архиве, из списка: "Логический", "Целый", "Вещественный" и "Строка";

- *Источник* - указывает на тип и адрес источника. Тип источника указывается из списка: "Пассивный", "Пассивный атрибут параметра" или "Активный атрибут параметра". Пассивный архив не имеет ассоциированного источника значений; данные в такой архив источник передаёт самостоятельно. Типы с атрибутом параметра в поле адреса указывают на параметр подсистемы "Сбор данных" как на источник. Пассивный атрибут параметра направляет данные в архив самостоятельно с собственным периодом сбора данных. Активный атрибут параметра опрашивается задачей архивирования этой подсистемы;

- *Отложенный запуск* – позволяет при запуске системы не выводить предупредительное сообщение, если источник отсутствует;

- *Период буфера (сек)* - указывает на периодичность значений в буфере архива;

- *Размер буфера (единиц)* - указывает размерность или глубину буфера архива.

Размерность обычно устанавливается в пересчёте на 60 сек периодичности задачи архивирования с запасом;

- *Жёсткая сетка времени буфера* - указывает на режим буфера. Режим жёсткой сетки подразумевает резервирование памяти под каждое значение, но без метки времени. Такой режим исключает возможность упаковки смежно-одинаковых значений, но экономит на хранении метки времени. Иначе буфер работает в режиме хранения значения и метки времени и поддерживает упаковку смежно-одинаковых значений;

- *Высокое разрешение времени буфера* - указывает на возможность хранения значений с периодичностью до 1 микросекунды, иначе значения могут храниться с периодичностью до 1 секунды.

Вид вкладки "Архиваторы" показан на рисунке 85.

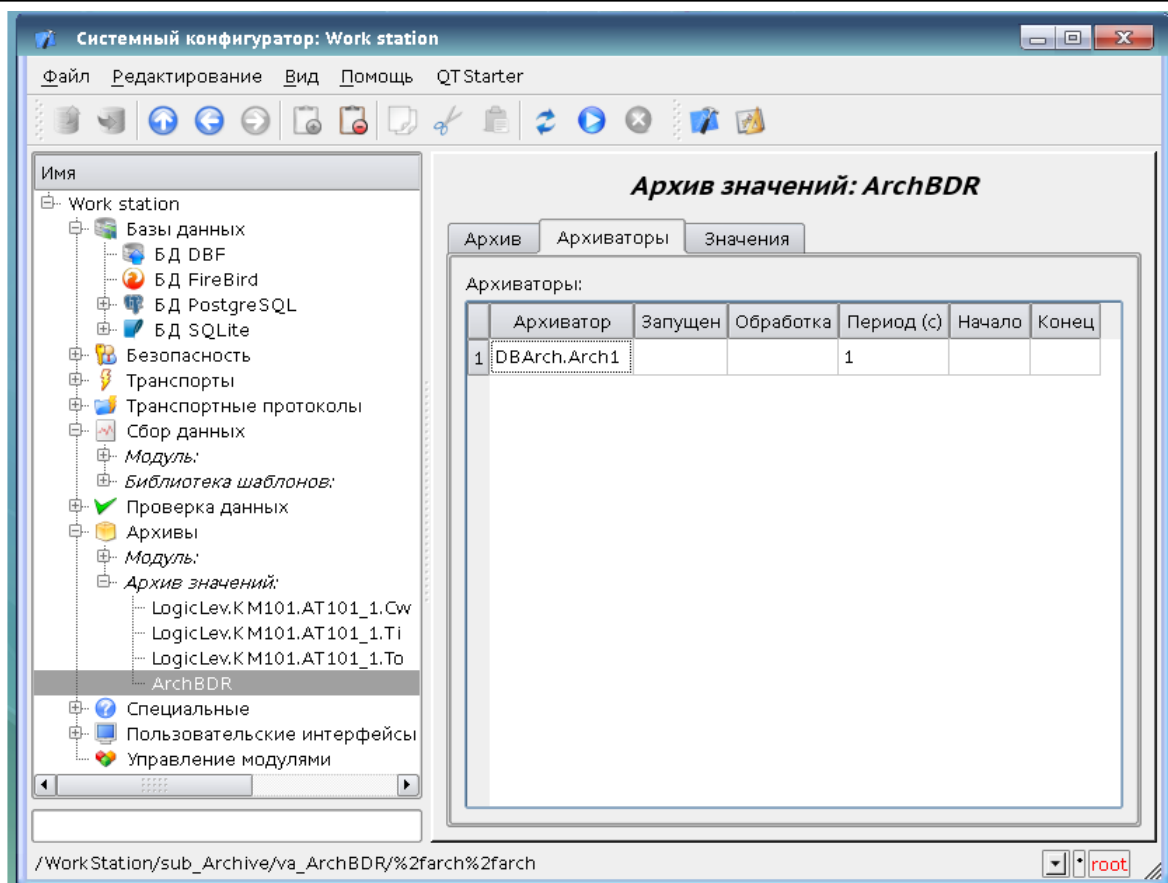


Рисунок 85

Вкладка "Архиваторы" содержит таблицу с конфигурацией процесса обработки данного архива доступными архиваторами. В строках расположены доступные архиваторы, а в колонках параметры:

- *Архиватор* - информация об адресе архиватора;
- *Запущен* - информация о состоянии "Запущен" архиватора;
- *Обработка* - признак обработки данного архива архиватором. Поле доступно для модификации пользователем;
- *Период (с)* - информация о периодичности архиватора;
- *Начало* - дата начала данных архива в архиваторе;
- *Конец* - дата конца данных архива в архиваторе.

Вид вкладки "Значения" показан на рисунке 86.

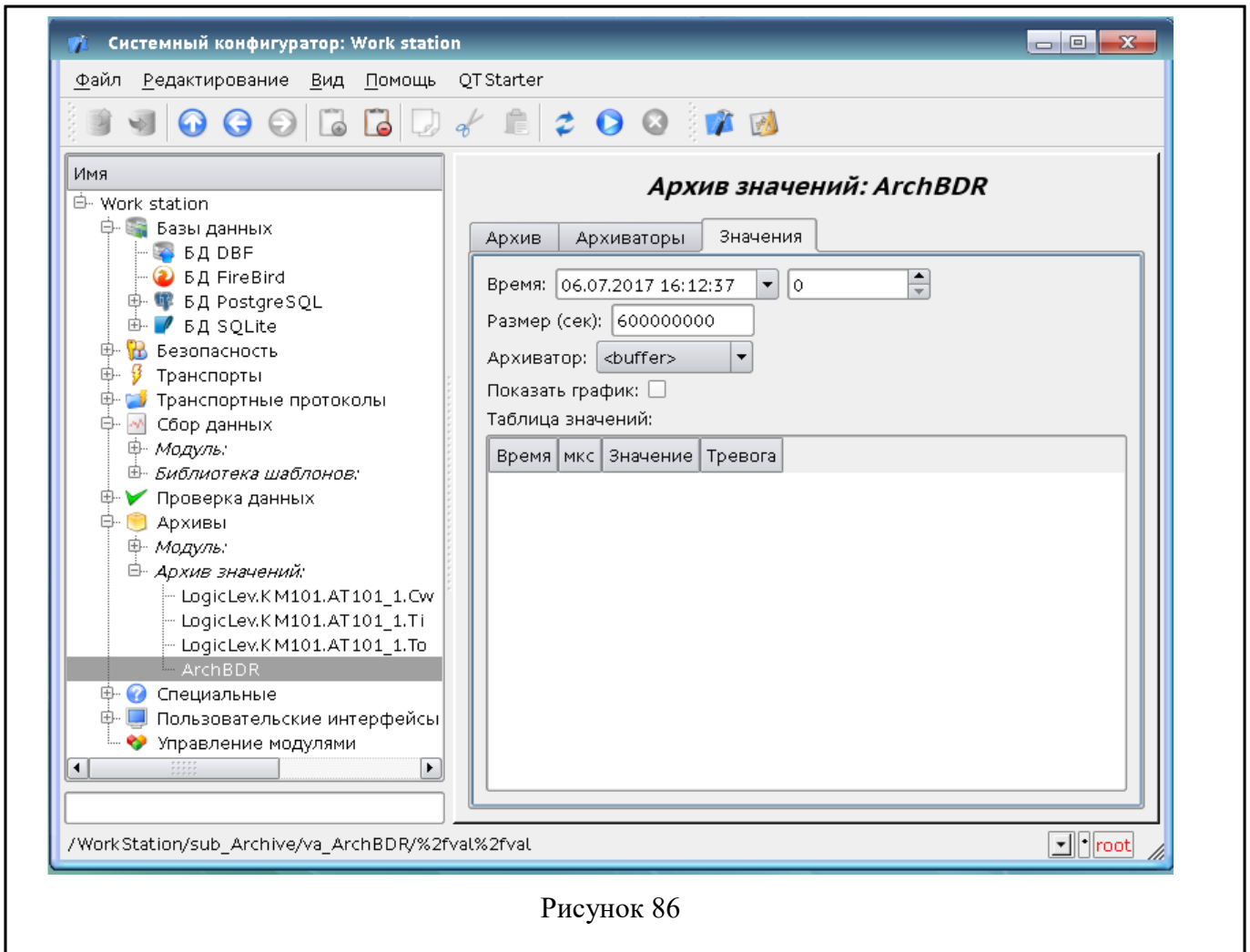


Рисунок 86

Вкладка "Значения" содержит запрос значений в архиве и результат в виде таблицы значений или изображения тренда. Запрос значений содержит поля:

- *Время* - указывает время запроса. Содержит два поля: поле дата+время и микросекунды;

- *Размер* - указывает размер или глубину запроса в секундах;

- *Архиватор* - указывает архиватор значений, для которого обрабатывать запрос. Если значение отсутствует, то запрос будет обработан для буфера и всех архиваторов. Если указано <buffer>, то запрос будет обработан только для буфера архива;

- *Показать график* - указывает на необходимость представления данных архива в виде графика (тренда), иначе результат представляется в таблице, содержащей только время и значение. В случае установки этого поля формируется и отображается график (рисунок 87), кроме того появляются дополнительные конфигурационные поля настройки изображения:

- *Размер изображения* - указывает ширину и высоту формируемого изображения в пикселах;

- *Шкала значения* - указывает нижнюю и верхнюю границу шкалы значения. Если оба значения установлены в 0 или равны, то шкала будет определяться автоматически в зависимости от значений.

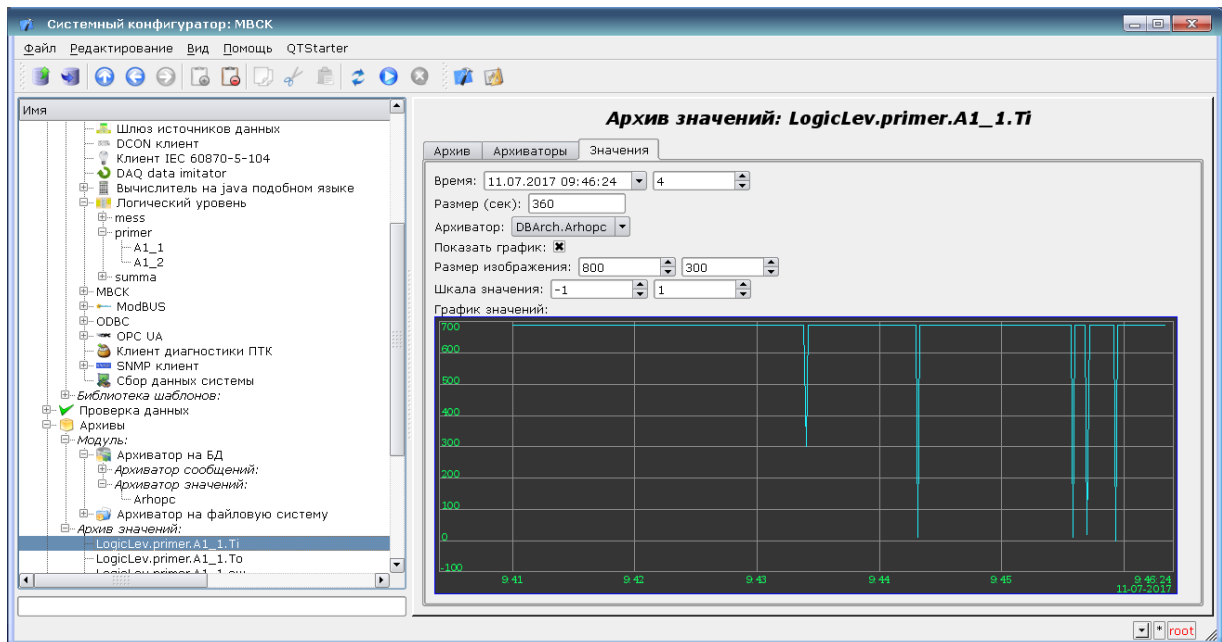


Рисунок 87

Каждый модуль подсистемы "Архивы" предоставляет конфигурационную страницу с вкладками "Архиваторы" и "Помощь". Вкладка "Архиваторы" содержит списки архиваторов сообщений и значений, зарегистрированных в модуле (рисунок 88). В контекстном меню списков пользователю предоставляется возможность добавления, удаления и перехода к нужному архиватору (рисунок 89). Во вкладке "Помощь" содержится информация о модуле подсистемы "Архивы", состав которой идентичен для всех модулей (рисунок 90).

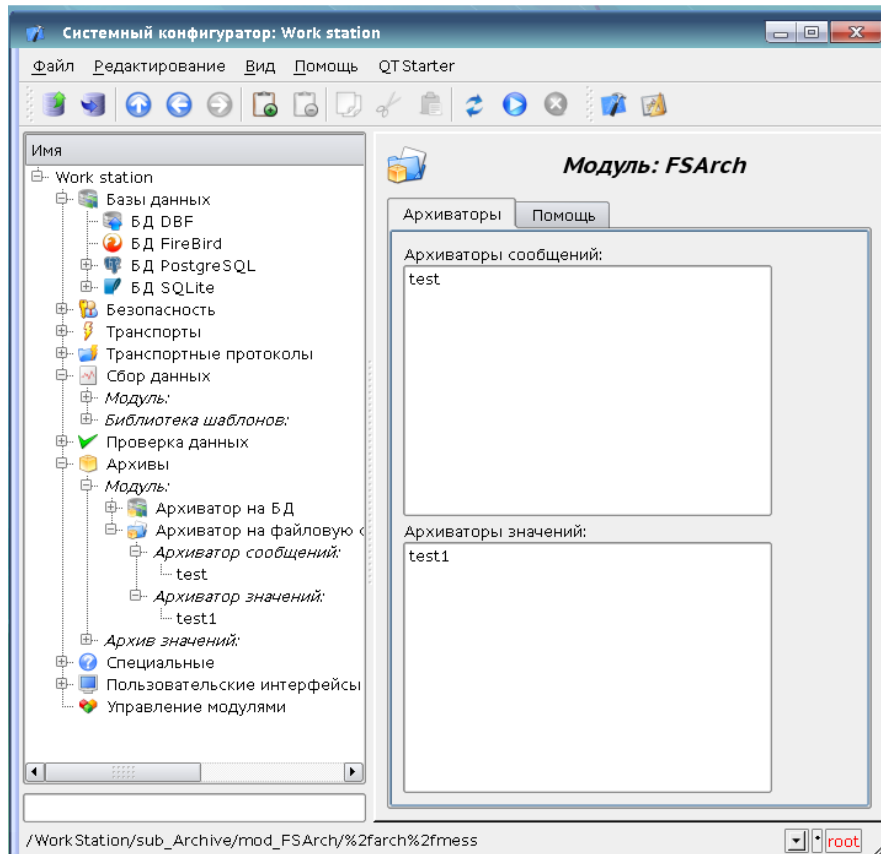


Рисунок 88

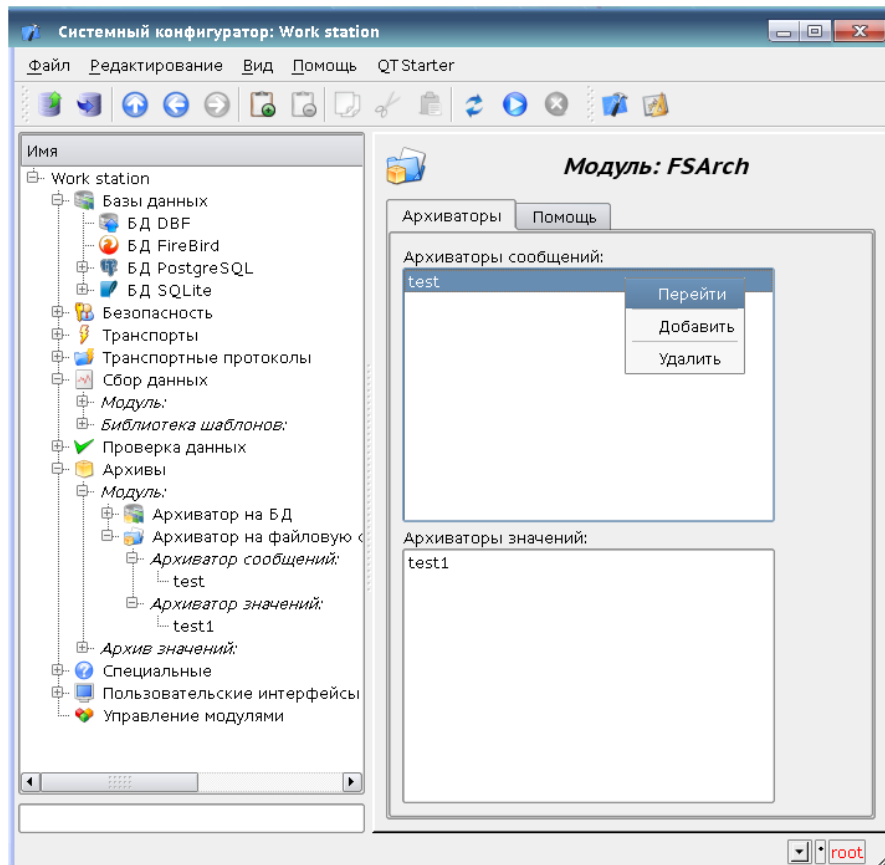


Рисунок 89

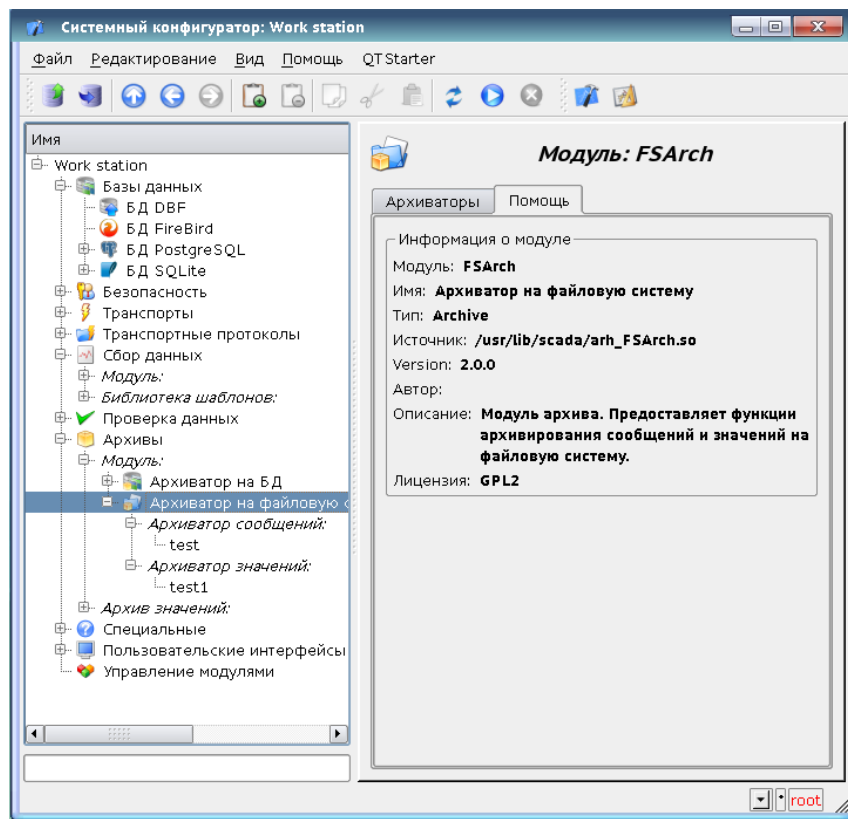


Рисунок 90

Архиваторы сообщений содержат собственную страницу конфигурации с вкладками "Архиватор" и "Сообщения".

Вид вкладки "Архиватор" показан на рисунке 91.

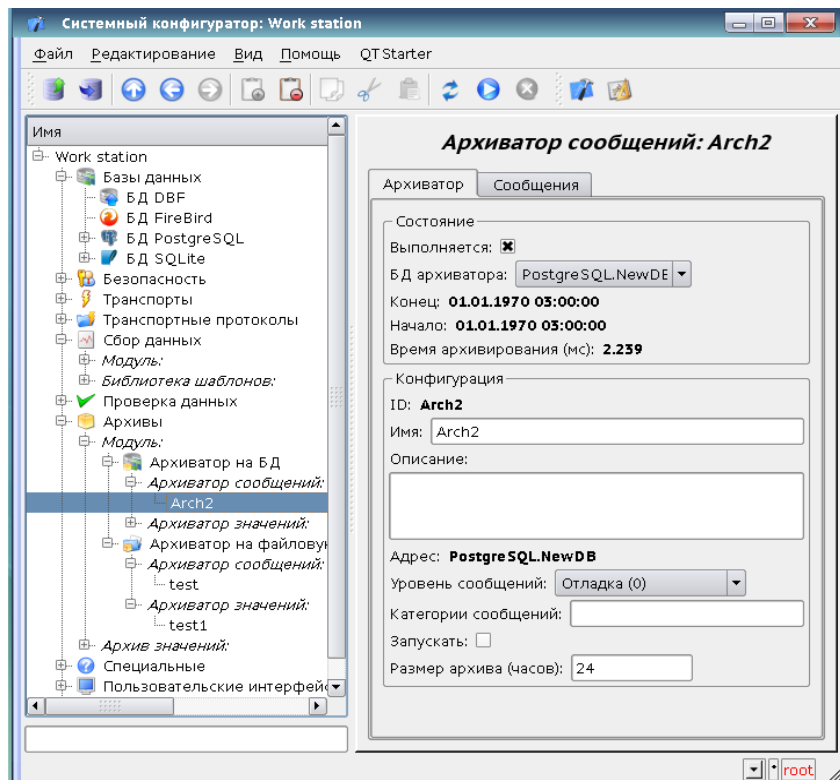


Рисунок 91

Вкладка "Архиватор" содержит основные настройки. Состав этих настроек может несколько отличаться от одного модуля этой подсистемы к другому, о чём можно узнать в собственной документации модулей. В качестве примера рассматриваются настройки архиватора сообщений у модуля архива на БД. Настройки:

Раздел "Состояние" - содержит свойства характеризующие состояние архиватора:

- *Выполняется* - состояние архиватора "Выполняется". Исполняющийся архиватор обрабатывает буфер архива сообщений и помещает его данные в своё хранилище, а также обслуживает запросы на доступ к данным в хранилище;

- *БД архиватора* - адрес БД для хранения данных архиватора, выбирается из списка щелчком мыши;

- *Конец* – отображается дата+время последних данных в хранилище архиватора;

- *Начало* – отображается дата+время первых данных в хранилище архиватора;

- *Время архивирования (мс)* - время, затраченное на архивирование данных архива сообщений.

Раздел "Конфигурация" - непосредственно содержит поля конфигурации:

- *ID* - информация об идентификаторе архиватора;

- *Имя* - указывает имя архиватора;

- *Описание* - краткое описание архиватора и его назначения;

- *Адрес* - адрес хранилища в специфичном для типа архиватора (модуля) формате.

Описание формата записи адреса архиватора, как правило, доступно во всплывающей подсказке этого поля. В примере это относительный путь к директории хранилища;

- *Уровень сообщений* - указывает на уровень сообщений архиватора. Сообщения с уровнем более или равным указанному обрабатываются архиватором;

- *Категории сообщений* - список категорий сообщений, разделённый символом ';'. Сообщения, попавшие под шаблоны категорий, будут обрабатываться архиватором. В категории можно указывать элементы выборки по шаблону, а именно символы '*' – для любой подстроки и '?' - для любого символа;

- *Запускать* - указывает на состояние "Выполняется", в которое следует перевести архиватор при загрузке;

- *Размер архива (часов)* – указывается максимальный размер архива в часах.

Вид вкладки "Сообщения" показан на рисунке 92.

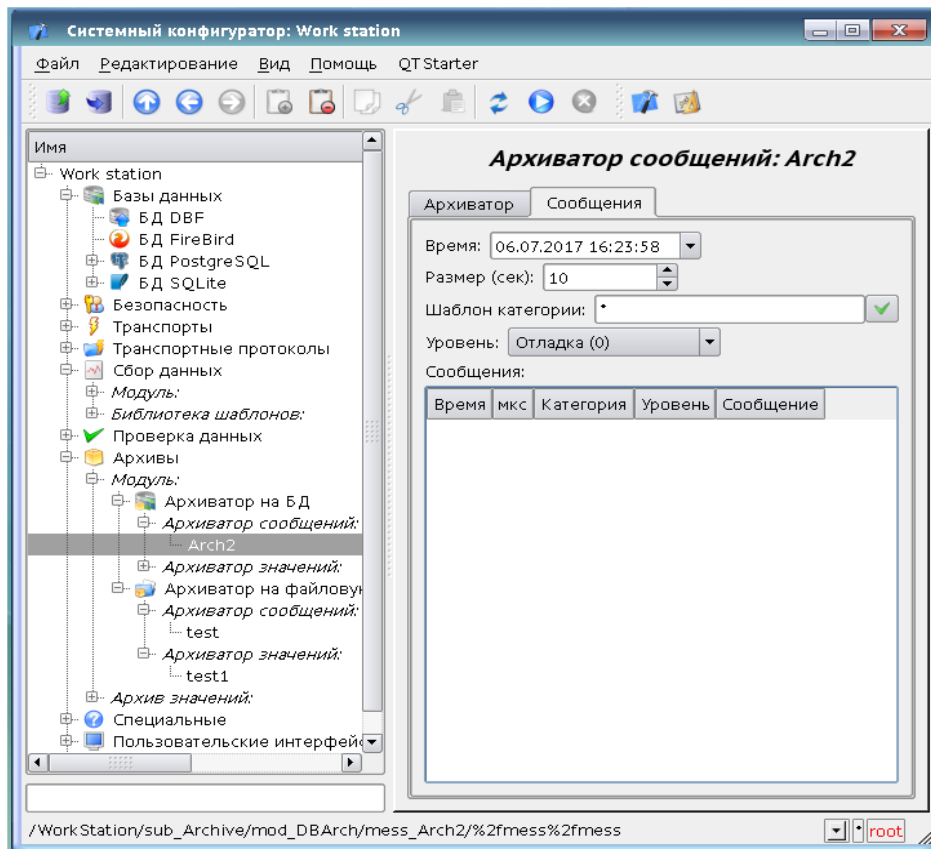


Рисунок 92

Вкладка "Сообщения" содержит форму запроса сообщений из архива данного архиватора:

- *Время* - указывает время запроса;
- *Размер (сек)* - указывает размер или глубину запроса, в секундах;
- *Шаблон категории* - указывает категорию запрошенных сообщений. В категории можно указывать элементы выборки по шаблону, а именно символы '*' - для любой подстроки и '?' - для любого символа;
- *Уровень* - указывает на минимальный уровень сообщений, т.е. запрос будет обработан для сообщения с уровнем более или равному указанному.

Таблица результата содержит строки сообщений с колонками:

- *Время* - время сообщения;
- *Категория* - категория сообщения;
- *Уровень* - уровень сообщения;
- *Сообщение* - текст сообщения.

Архиваторы значений содержат собственную страницу конфигурации с вкладками "Архиватор" и "Архивы".

Вид вкладки "Архиватор" показан на рисунке 93.

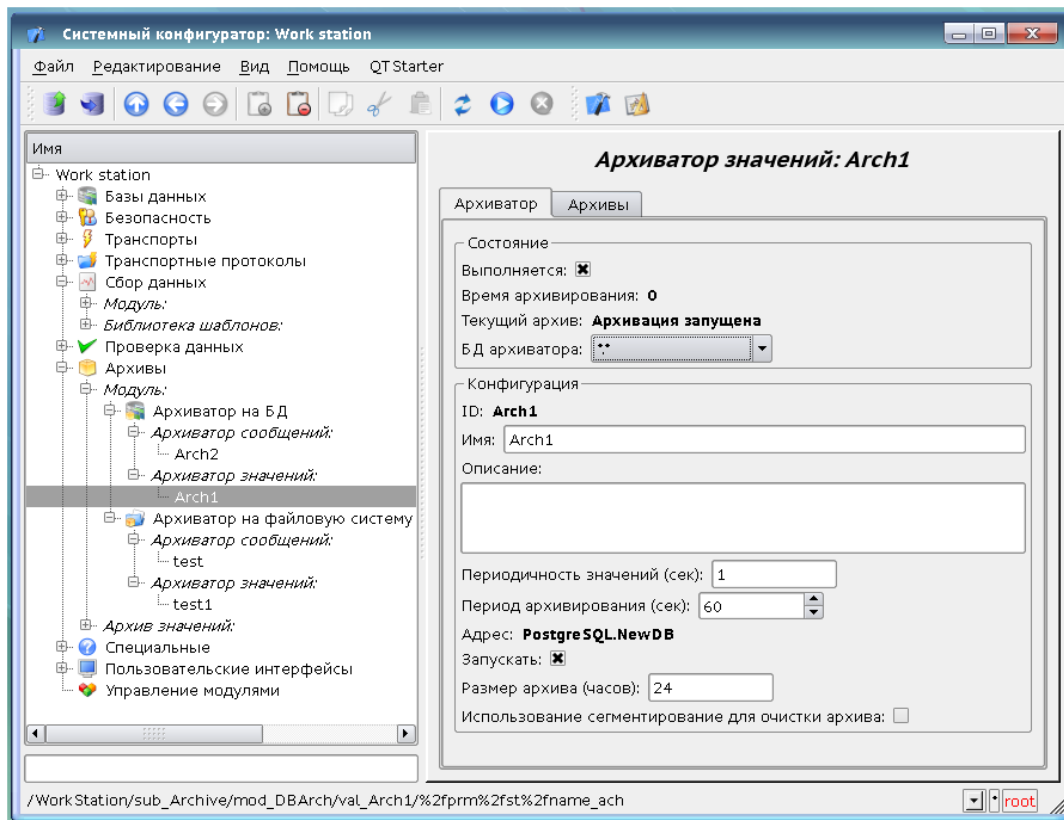


Рисунок 93

Вкладка "Архиватор" содержит основные настройки. Состав этих настроек может несколько отличаться от одного модуля этой подсистемы к другому, о чём можно узнать в собственной документации модулей. В качестве примера рассматриваются настройки архиватора значений у модуля архива на БД.

Раздел "Состояние" - содержит свойства, характеризующие состояние архиватора:

- *Выполняется* - состояние архиватора "Выполняется". Исполняющийся архиватор обрабатывает буфера архивов значений и помещает их данные в своё хранилище, а также обслуживает запросы на доступ к данным в хранилище;

- *Время архивирования (мс)* - информация о времени затраченном на архивирование данных буферов архивов. Периодичность архивирования указывается в поле "Период архивирования" раздела "Конфигурация" этой вкладки;

- *БД архиватора* - адрес БД для хранения данных архиватора, выбирается из списка щелчком мыши.

Раздел "Конфигурация" - непосредственно содержит поля конфигурации:

- *ID* - информация об идентификаторе архиватора;

- *Имя* - указывает имя архиватора;

- *Описание* - краткое описание архиватора и его назначения;

- *Периодичность значений (сек)* - указывает периодичность значений, которые содержатся в хранилище архиватора;

- *Период архивирования (сек)* - указывает периодичность задачи архивирования данных буферов архивов. Размерность буферов архивов во временном выражении должна быть не менее, а лучше несколько больше, периодичности задачи архивирования;

- *Адрес* - адрес хранилища в специфичном для типа архиватора (модуля) формате. Описание формата записи адреса архиватора, как правило, доступно во всплывающей подсказке этого поля. В примере это относительный путь к директории хранилища;

- *Запустить* - указывает на состояние "Выполняется", в которое переводить архиватор при загрузке;

- *Размер архива (часов)* – указывается максимальный размер архива в часах;

- *Использование сегментирования для очистки архива* – чек-бокс для выбора использования сегментирования.

Внимание! При отключении флага «Выполняется» во время работы СКАДА удаляет все архивы, привязанные к этому архиватору!

Вид вкладки "Архивы" для модуля «Архиватор на БД» показан на рисунке 94.

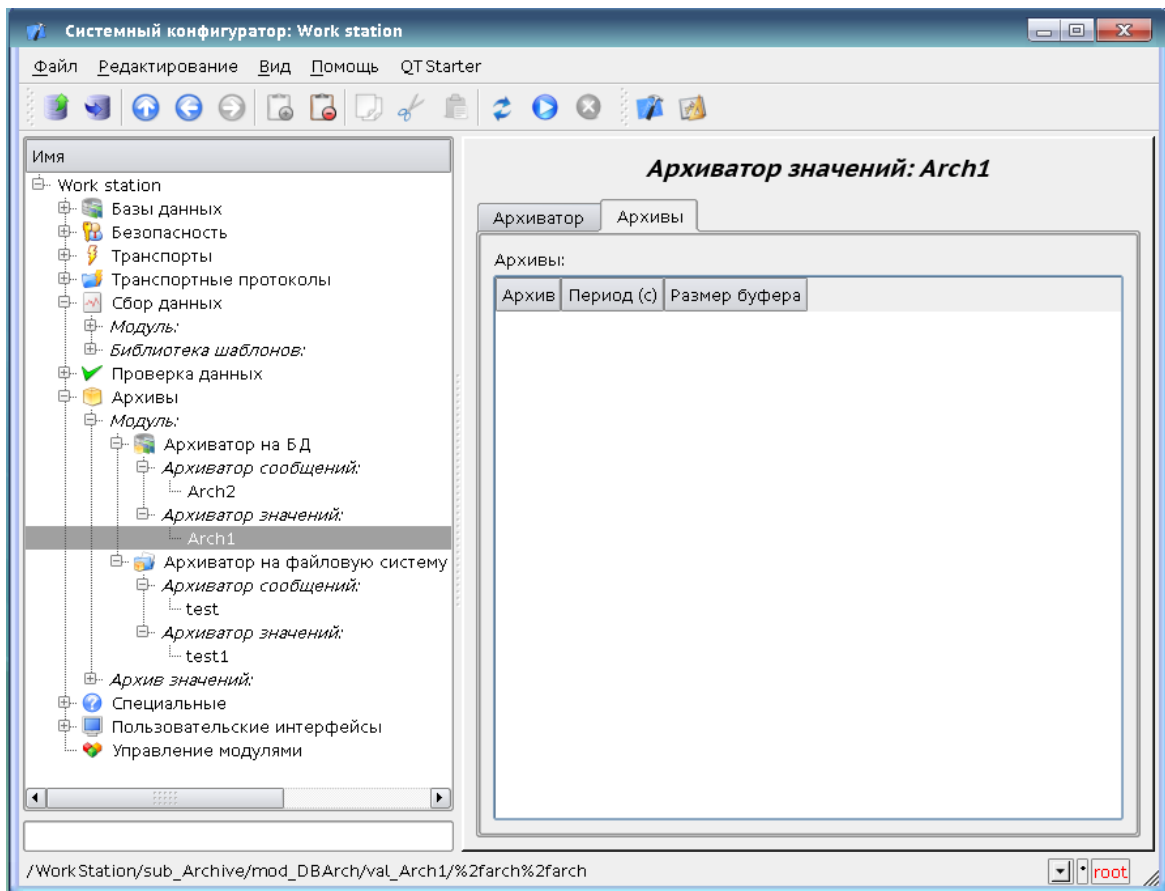


Рисунок 94

Вкладка "Архивы" содержит таблицу с информацией об архивах, обрабатываемых архиватором. В строках таблица содержит архивы, а в колонках информацию:

- *Архив* - имя архива;
- *Период (с)* - периодичность архива в секундах;
- *Размер буфера* - размерность буфера в единицах;
- *Размер файлов (Мб)* - специфичное для модуля «Архиватор на файловую систему»

поле с информацией о суммарной размерности файлов хранилища архиватора для архива.

В случае с модулем «Архиватор на файловую систему» в этой вкладке ещё содержится форма экспорта данных архиватора, содержащая следующие параметры:

- *Архив* - имя архива для экспорта, выбирается мышью из списка;
- *Начало* - дата и время начала данных для экспорта;
- *Конец* - дата и время конца данных для экспорта;
- *Тип* - тип экспортируемого файла (ascii или wav), выбирается мышью из списка;
- *В файл* - имя файла для экспорта.

После настройки всех вышеуказанных параметров можно осуществить экспорт данных нажатием мыши на кнопку «Экспорт».

3.10 Подсистема "Пользовательские интерфейсы"

3.10.1 Общие сведения

Подсистема является модульной. Для конфигурации подсистемы предусмотрена корневая страница подсистемы "Пользовательские интерфейсы", содержащая вкладки "Модули" и "Помощь".

Вкладка "Модули" содержит список модулей подсистемы и идентична для всех модульных подсистем. Вкладка "Помощь" содержит краткую помощь для данной страницы.

Каждый модуль подсистемы "Пользовательские интерфейсы" предоставляет конфигурационную страницу с вкладками "Пользовательский интерфейс" и "Помощь".

Вид вкладки "Пользовательский интерфейс" показан на рисунке 95.

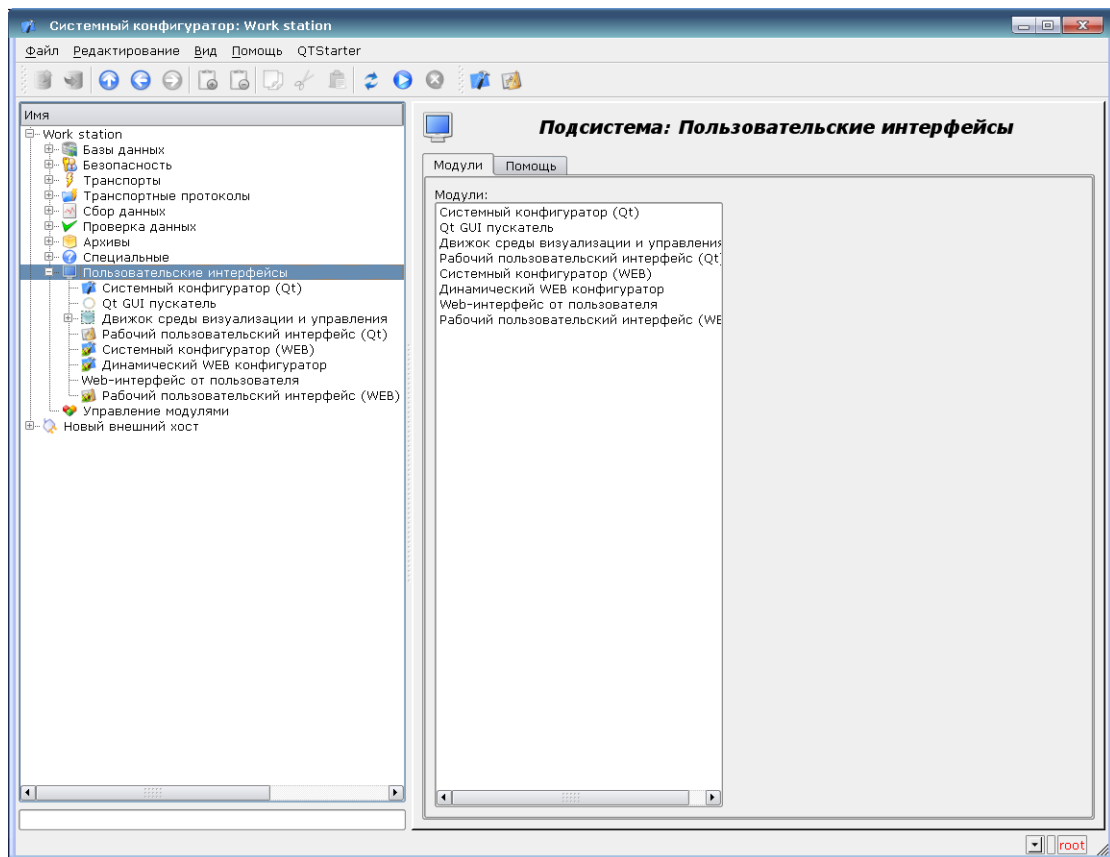


Рисунок 95

Вкладка "Пользовательский интерфейс" предоставляет параметр для контроля состояния "Выполняется" модуля, а также разделы конфигурации, специализированные для модулей этой подсистемы. Во вкладке "Помощь" содержится информация о модуле подсистемы "Пользовательские интерфейсы", состав которой идентичен для всех модулей.

3.10.2 Рабочий пользовательский интерфейс (QT)

Модуль «Vision» предоставляет механизм конечной визуализации среды визуализации и управления (СВУ) в СКАДА. Модуль основан на многоплатформенной библиотеке графического пользовательского интерфейса (GUI) QT. В своей работе модуль использует данные движка СВУ (модуль VCAEngine).

Среда визуализации и управления (СВУ) является неотъемлемой составляющей СКАДА. Она применяется на клиентских станциях с целью доступного предоставления информации об объекте управления и выдачи управляющих воздействий на объект.

Данный модуль непосредственной визуализации СВУ предназначен для формирования и исполнения интерфейсов СВУ в среде графической библиотеки QT.

Модуль обеспечивает:

- три уровня сложности в формировании интерфейса визуализации, позволяющие органично осваивать и применять инструментарий от простого к сложному;
- формирование из шаблонных кадров путём назначения динамики (без графической конфигурации);
- графическое формирование новых кадров путём использования готовых элементов визуализации из библиотеки (мнемосхемы);
- построение интерфейсов визуализации различной сложности, начиная от простых плоских интерфейсов мониторинга и заканчивая полноценными иерархическими интерфейсами, используемыми в СКАДА системах;
- смену динамики в процессе исполнения;
- построение новых шаблонных кадров на уровне пользователя и формирование специализированных под область применения библиотек кадров (например, включение кадров параметров, графиков и других элементов с увязкой их друг с другом), в соответствии с теорией повторного использования и накопления;
- построение новых пользовательских элементов визуализации и формирование специализированных под область применения библиотек кадров в соответствии с теорией повторного использования и накопления;
- описание логики новых шаблонных кадров и пользовательских элементов визуализации, как простыми связями, так и лаконичным, полноценным языком пользовательского программирования;
- возможность включения в пользовательские элементы визуализации функций (или кадров вычисления функций) объектной модели СКАДА, практически связывая представление с алгоритмом вычисления (например, визуализируя библиотеку моделей аппаратов ТП для последующего визуального построения моделей ТП);

- разделение данных пользовательских интерфейсов и интерфейсов представления этих данных, позволяющее строить интерфейс пользователя в одной среде, а исполнять во многих других (QT, Java ...);
- возможность подключение к исполняющемуся интерфейсу для наблюдения и коррекции действий (например, при обучении операторов и контроля в реальном времени за его действиями);
- визуальное построение различных схем с наложением логических связей и последующим централизованным исполнением в фоне (визуальное построение и исполнение математических моделей, логических схем, релейных схем и иных процедур);
- предоставление функций объектного API в систему СКАДА, может использоваться для управления свойствами интерфейса визуализации из пользовательских процедур;
- построение серверов кадров, элементов визуализации и проектов интерфейсов визуализации с возможностью обслуживания множественных клиентских соединений;
- организацию клиентских станций на различной основе с подключением к центральному серверу;
- полноценный механизм разделения полномочий между пользователями, позволяющий создавать и исполнять проекты с различными правами доступа к его компонентам;
- гибкое формирование правил сигнализаций и уведомления, с учётом и поддержкой различных способов уведомления;
- поддержку пользовательского формирования палитры и шрифтовых предпочтений для интерфейса визуализации;
- поддержку пользовательского формирования карт событий под различное оборудование и пользовательские предпочтения;
- гибкое хранение и распространение библиотек виджетов, кадров и проектов интерфейсов визуализации в БД, поддерживаемых СКАДА. Практически пользователю нужно только зарегистрировать полученную БД с данными.

3.10.3 Конфигурирование модуля «Рабочий пользовательский интерфейс»

Вкладка «Пользовательский интерфейс» (рисунок 96) содержит следующие опции:

- Станция движка СВУ - выбирается мышью из списка «Local, Loop, Loop SSL». Определяет станцию, на которой будет запускаться движок СВУ;
- Стартовый пользователь - выбирается мышью из списка всех пользователей;
- Время жизни страниц в кеше - время в часах, определяет интервал неактивности для закрытия страниц в кеше. Нулевое значение времени исключает закрытие страниц в кеше;
- Перечень запускаемых проектов - перечень автоматически запускаемых проектов, разделённый символом ';'. Для открытия окна проекта на нужном дисплее (1) используется имя проекта в формате «PrjName-1»;
- Переход к конфигурации перечня удалённых станций - переход к настройкам подсистемы «Транспорты».

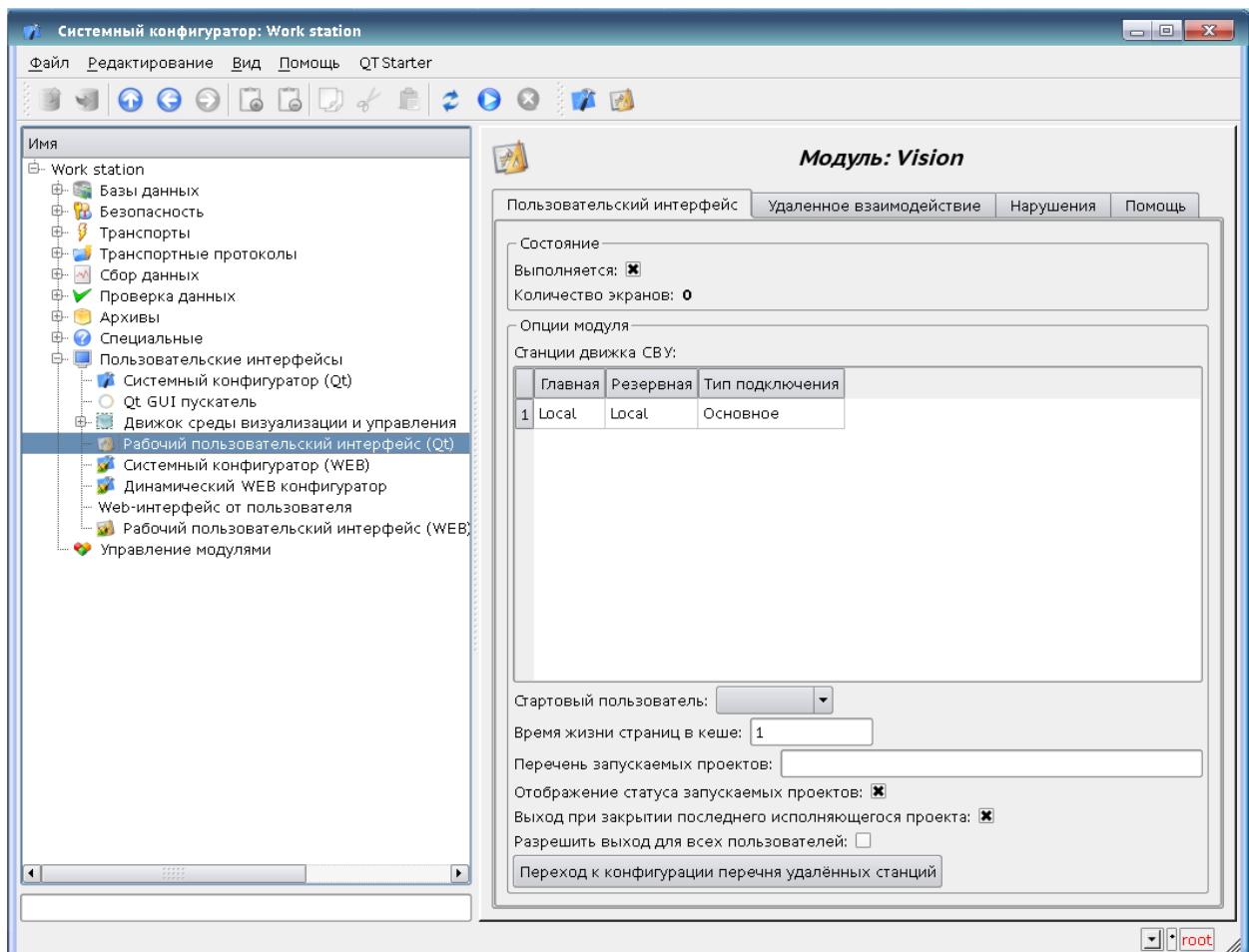


Рисунок 96

Вкладка «Удаленное взаимодействие» предоставляет возможность конфигурирования вывода видеокладов на экраны табло коллективного пользования

(рисунок 97). Для этого выбрать в списке станций для удаленного открытия видеокладов на экранах «Local». Задать для каждого экрана группу пользователей, для которых разрешено управление отображением видеокладов на этом экране. Также можно изменить имена экранов, отредактировав их в таблице. Кнопка «Обновить» заново определяет количество подключенных мониторов и отображает эту информацию в таблице.

Чтобы настроить вывод видеокладов на экраны табло для удаленного рабочего места, необходимо выбрать название этого рабочего места из списка станций для удаленного открытия видеокладов. Затем задать количество экранов и их имена для выбранного рабочего места. Также можно запросить имена экранов с удаленной станции, нажав кнопку «Обновить».

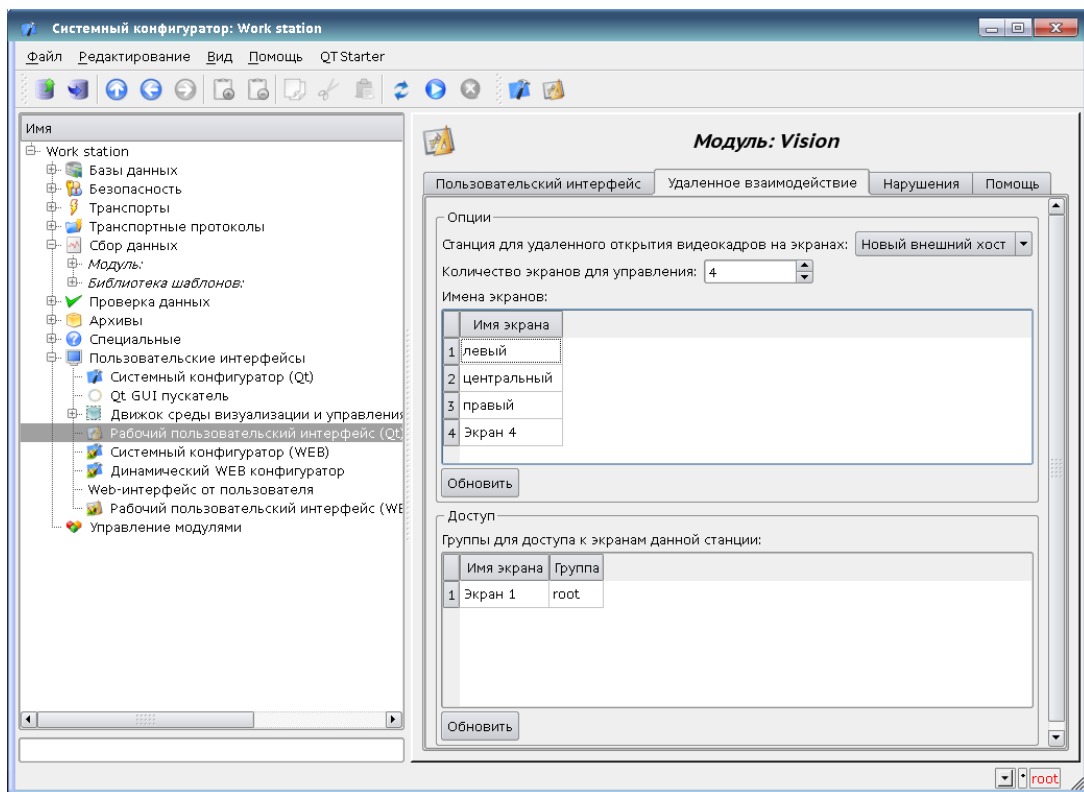


Рисунок 97

Вкладка «Нарушения» содержит настройку «Команда проигрывания» формата «play -q %f» - командная строка для вызова проигрывания звука. Для вставки имени файла источника используется «%f». Если файл источника не указан, то данные направляются в канал.

3.10.4 Системный конфигурактор (QT)

Модуль "QTCfg" предоставляет конфигурактор системы СКАДА. Конфигуратор реализован на основе многоплатформенной библиотеки графического пользовательского интерфейса (GUI) QT.

В основе модуля лежит язык интерфейса управления системой СКАДА, предоставляющий единый интерфейс конфигурации. Обновление модуля может потребоваться только в случае обновления спецификации языка интерфейса управления. Для запроса контекста страницы используется групповой запрос интерфейса управления, что позволяет оптимизировать время удалённого доступа по высоколатентным и медленным каналам связи.

Вкладка «Пользовательский интерфейс» (рисунок 98) предоставляет следующие опции:

- Стартовый путь конфигулятора - директория в формате «/path», где хранятся настройки запуска конфигулятора;
- Стартовый пользователь конфигулятора - пользователь, с настройками которого запускается конфигурактор при запуске, выбирается мышью из списка пользователей;
- Перейти к конфигурации списка удалённых станций - переход к настройкам подсистемы «Транспорты».

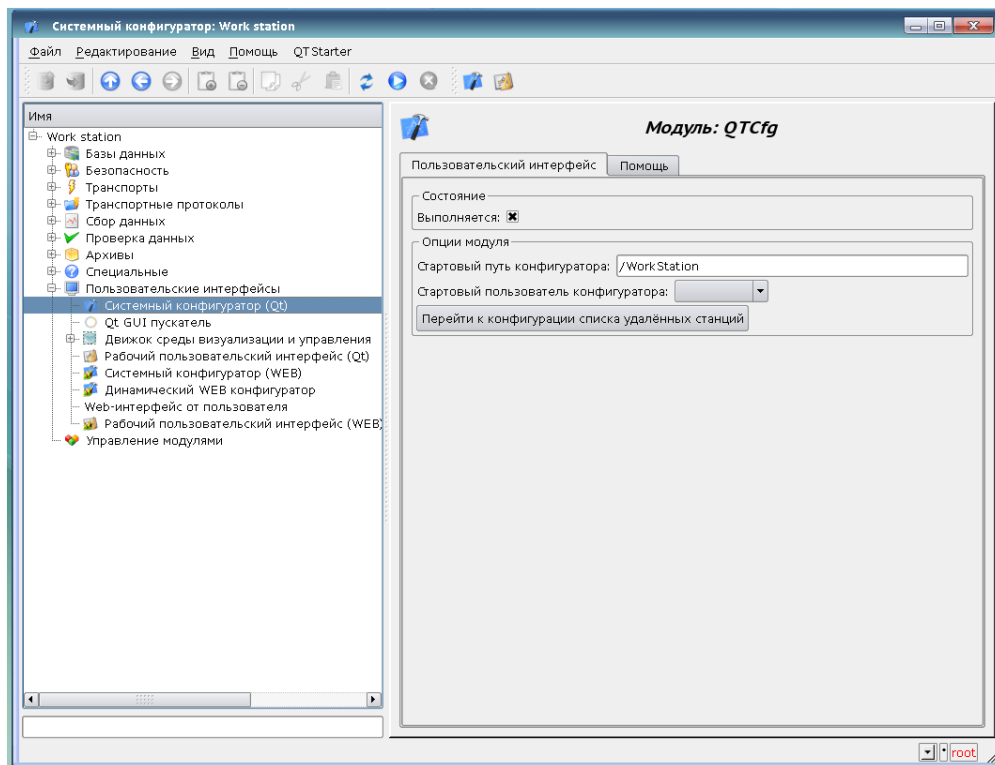


Рисунок 98

3.10.5 QT GUI пускатель

Модуль «QT GUI пускатель» предоставляет в СКАДА пускатель QT GUI модулей. Отдельный модуль для запуска QT GUI модулей понадобился по причине необходимости организации однопоточного исполнения всех компонентов и централизованной инициализации главного объекта QT-библиотеки - QApplication.

Для запуска QT GUI модулей используется расширенный интерфейс вызова функций модулей.

Данный интерфейс подразумевает экспортирование функций внешними модулями. В нашем случае QT GUI модули должны экспортировать следующие функции:

- QIcon icon(); - Передаёт объект иконки вызываемого модуля;
- QMainWindow *openWindow(); - Создаёт объект главного окна данного QT GUI модуля и передаёт его пускателю. Может возвращать NULL в случае невозможности создания нового окна.

Для идентификации QT GUI модуль должен определять информационный элемент модуля "SubType" как "QT". Исходя из этого признака, пускатель с ним работает.

После получения объекта главного окна пускатель добавляет свою панель управления и пункт меню в это окно и запускает его. Панель управления пускателя содержит иконки для вызова всех доступных QT GUI модулей. Для исключения добавления панели управления или пункта меню модуль, содержащий окно, может указать свойства "QTStarterToolDis" или "QTStarterMenuDis" соответственно.

Для указания QT GUI модулей, запускаемых при старте, модуль «QT GUI пускатель» содержит конфигурационное поле StartMod. В данном поле записываются идентификаторы запускаемых модулей через ';'. Конфигурационное поле StartMod можно описать в конфигурационном файле, а также в системной таблице БД через диалог конфигурации модуля (рисунок 99).

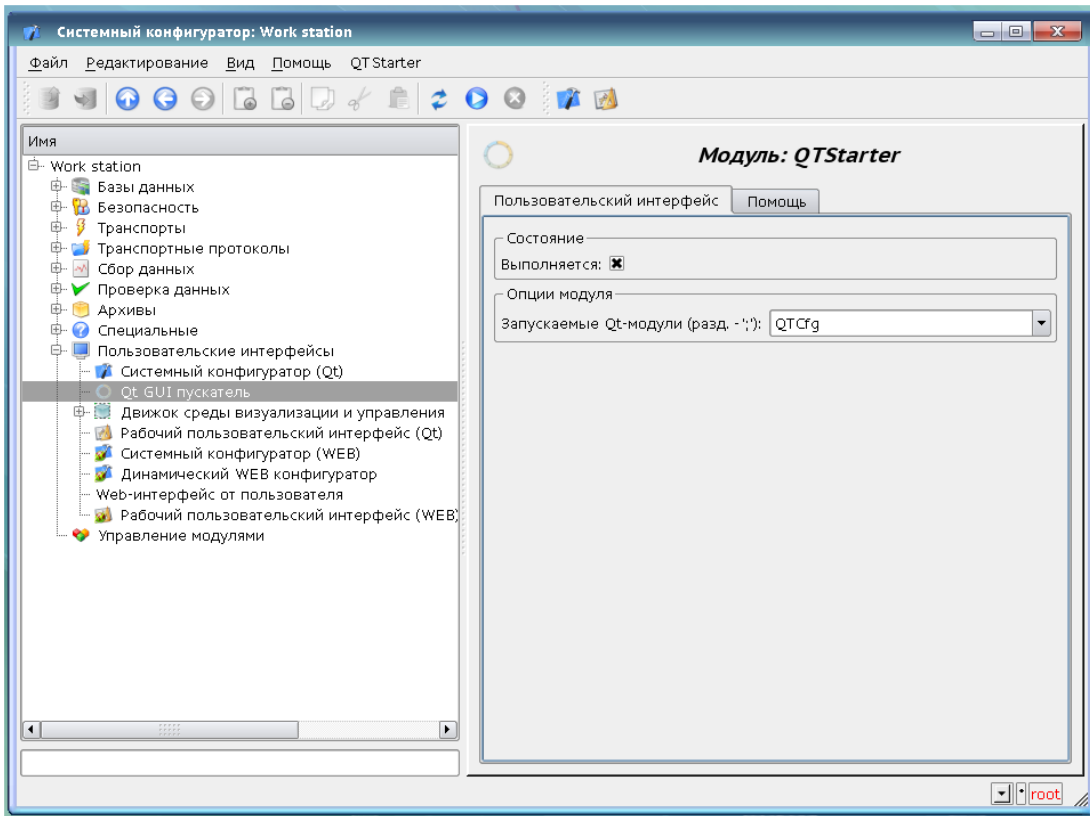


Рисунок 99

Опции модуля:

Запускаемые QT-модули (разд. - ;) - список модулей, разделённый ';', которые будут доступны для запуска с помощью QT-пускателя.

В случае закрытия окон всех QT GUI модулей QT-пускатель создаёт своё диалоговое окно, которое предлагает выбрать доступные QT GUI модули или завершить работу СКАДА. Вид диалогового окна приведен на рисунке 100.

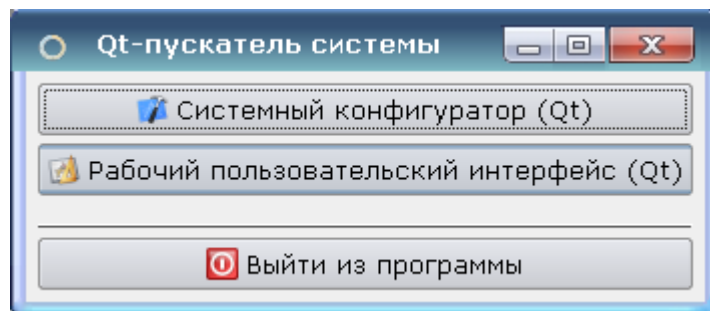


Рисунок 100

3.10.6 Движок среды визуализации и управления

Модуль VCAEngine предоставляет движок среды визуализации и управления (СВУ) в СКАДА. Данный модуль движка СВУ предназначен для формирования логической структуры СВУ и исполнения сеансов отдельных экземпляров проектов СВУ. Также модуль

предоставляет все необходимые данные конечным визуализаторам СВУ как посредством локальных механизмов взаимодействия СКАДА, так и посредством интерфейса управления СКАДА для удалённого доступа.

Сам модуль не реализует визуализации СВУ, а содержит данные в соответствии с идеологией "Модель/данные – Интерфейс". Визуализация данных этого модуля выполняется модулями визуализации СВУ, например, модулем Vision.

Любая СВУ может работать в двух режимах – разработки и исполнения. В режиме разработки формируется интерфейс СВУ, его компоненты и определяются механизмы взаимодействия. В режиме исполнения выполняется формирование интерфейса СВУ и производится взаимодействие с конечным пользователем на основе разработанных СВУ.

Интерфейс СВУ формируется из кадров, каждый из которых, в свою очередь, формируется из элементов примитивов или пользовательских элементов интерфейса. При этом пользовательские элементы интерфейса также формируются из примитивов или других пользовательских элементов. Таким образом, обеспечивается иерархичность и повторное использование уже разработанных компонентов.

Кадры и пользовательские элементы размещаются в библиотеках виджетов. Из элементов этих библиотек формируются проекты интерфейсов конечной визуализации СВУ. На основе же этих проектов формируются сеансы визуализации

Данная архитектура СВУ позволяет реализовать поддержку трёх уровней сложности процесса разработки интерфейсов управления:

1) формирование интерфейса ВУ (визуализации и управления) с помощью библиотеки шаблонных кадров путём помещения шаблонов кадров в проект и назначения динамики.

2) в дополнении к первому уровню производится формирование собственных кадров на основе библиотеки производных и базовых виджетов. Возможно как прямое назначение динамики в виджете, так и последующее её назначение в проекте.

3) в дополнении ко второму уровню производится самостоятельное формирование производных виджетов, новых шаблонных кадров, а также кадров с использованием механизма описания логики взаимодействия и обработки событий на одном из языков пользовательского программирования системы СКАДА.

Конфигурация модуля движка СВУ (рисунок 101):

- Проект - список доступных в системе проектов;
- Библиотеки виджетов - список библиотек виджетов, доступных в системе.

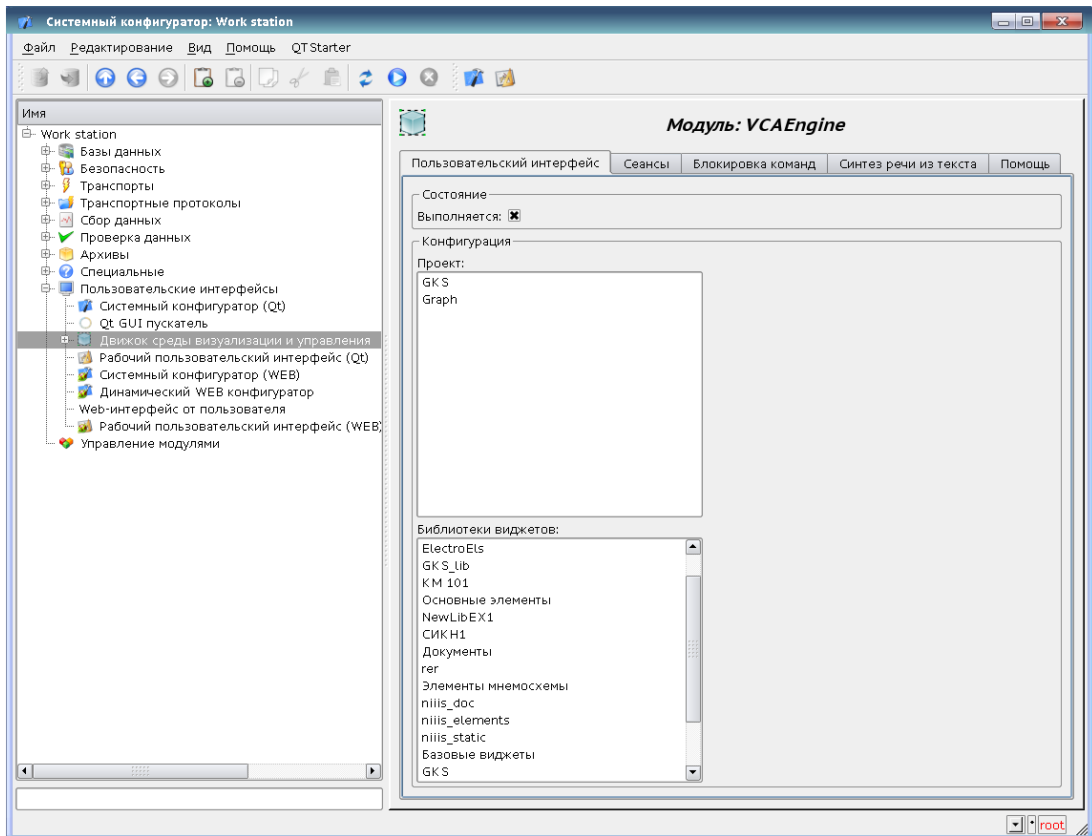


Рисунок 101

Вкладка «Сеансы» (рисунок 102) содержит следующие настройки:

- Сеансы - содержит список идентификаторов доступных в системе сеансов;
- Автоматическое создание и запуск - содержит настройки сеансов, которые будут автоматически создаваться и запускаться при старте системы.

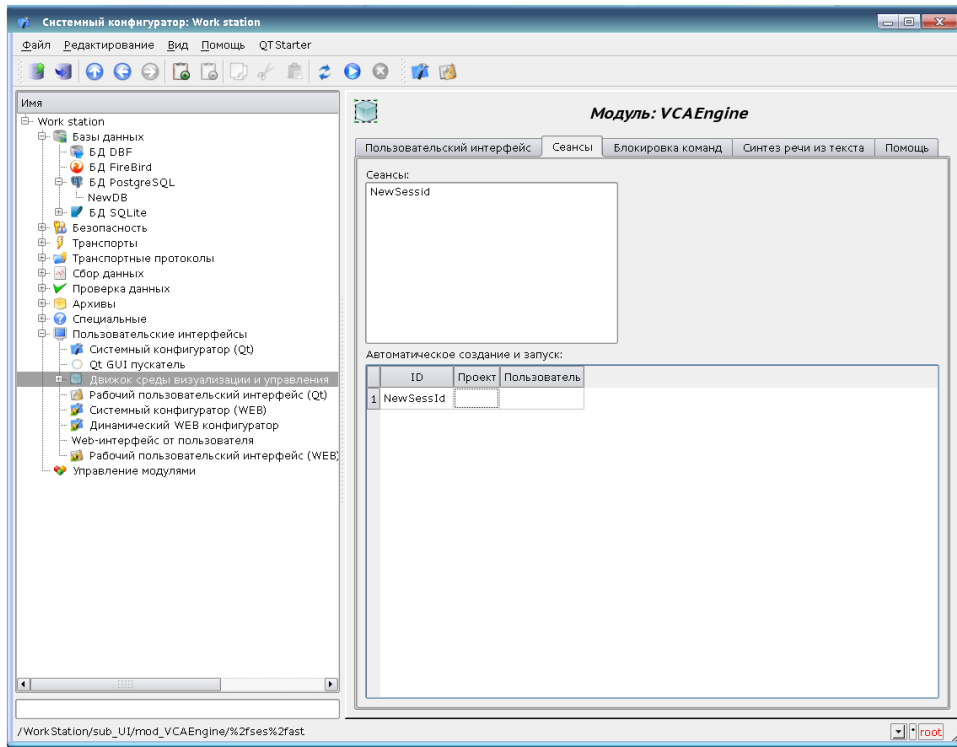


Рисунок 102

Вкладка «Блокировка команд» предоставляет механизм для блокировки управления при двухуровневой системе управления.

Сервера верхнего уровня являются основными подключениями, сервера нижнего уровня – дополнительными подключениями. Соответственно, при выборе блокировки основных подключений управление возможно с АРМ нижнего уровня, при выборе блокировки дополнительных подключений – управление с АРМ нижнего уровня блокируется.

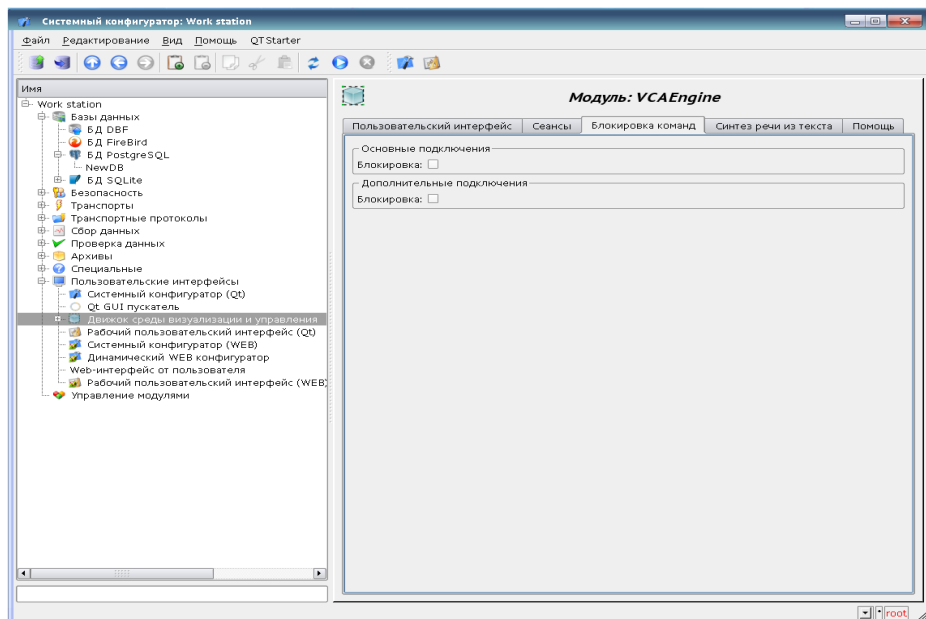


Рисунок 103

Вкладка «Синтез речи из текста» (рисунок 104) содержит следующие поля:

Команда - командная строка для вызова движка синтеза речи из текста, где «%t» - синтезируемый текст, «%f» - имя результирующего файла. Сама команда выбирается мышью из списка. Если файл результата не используется, то результат читается из канала. Если используется файл результата, но не используется «%t», то синтезируемый текст отправляется в канал.

Кодировка текста - кодировка текста движка, в которую будет перекодироваться текст.

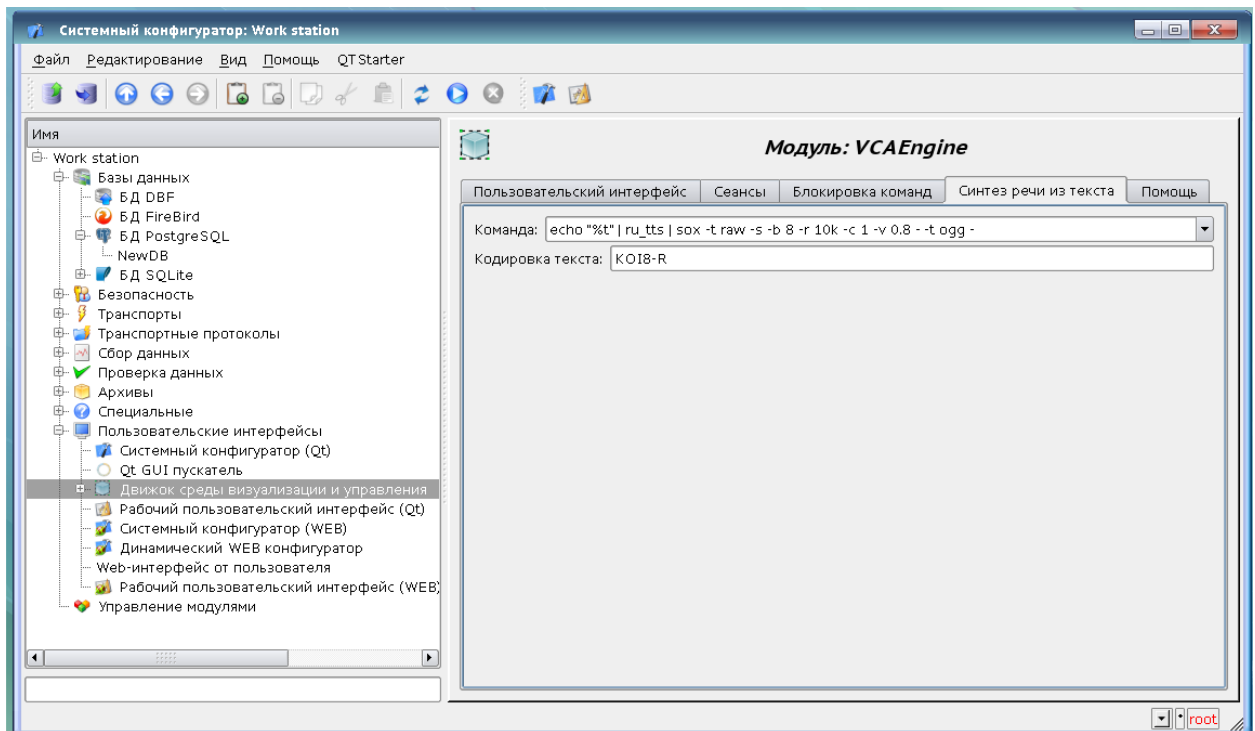


Рисунок 104

3.10.7 Модули, реализованные на основе WEB-технологий

Подсистема «Пользовательские интерфейсы» включает в себя следующие модули, реализованные на основе WEB технологий:

- системный конфигуризатор (WEB);
- динамический WEB конфигуризатор;
- WEB-интерфейс от пользователя;
- рабочий пользовательский интерфейс (WEB).

Системный конфигуризатор (WebCfg) предоставляет конфигуризатор системы, для работы которого достаточно обычного WEB-браузера (Opera, Mozilla и др.). Он также является модулем подсистемы транспортного протокола "HTTP": вызов "WebCfg"

производится из "HTTP" посредством расширенного механизма коммуникации через функции: `HttpGet()` и `HttpSet()`.

Динамический WEB конфигуратор (WebCfgD) предоставляет конфигуратор системы «СКАДА А-СОФТ», реализованный на основе следующих Web-технологий:

- HTTP — протокол передачи гипертекста;
- XHTML — расширенный язык разметки гипертекстовых документов;
- CSS — каскадные таблицы стилей гипертекстовых документов;
- JavaScript — встроенный в гипертекстовый документ язык программирования браузера;
- DOM — объектная модель документа внутренней структуры браузера;
- AJAX — механизм асинхронных и синхронных запросов из JavaScript к серверу;
- XML — расширяемый язык разметки.

Интерфейс конфигуратора формируется в WEB-браузере путём обращения к WEB-серверу и получения от него XHTML-документа по протоколу HTTP. В роли WEB-сервера выступает СКАДА система, которая поддерживает стандартные коммуникационные механизмы TCP-сетей (модуль Сокеты подсистема «Транспорты»), протокол передачи гипертекста (модуль HTTP подсистема «Транспортные протоколы»), а также шифрование трафика между браузером и сервером (модуль SSL подсистема «Транспорты»). Для получения доступа к интерфейсу конфигурирования необходимо настроить транспорт сокет или SSL в связке с протоколом HTTP. По умолчанию интерфейс модуля доступен по URL: `http://localhost:10002` или `http://localhost:10004`.

После получения XHTML-документа запускается программа на JavaScript для формирования динамического интерфейса конфигуратора.

Модуль работает на трёх WEB-браузерах: Mozilla Firefox 3.0.4, Opera 9.6.2 и Konqueror 3.5.10.

WEB-интерфейс от пользователя (WebUser) предоставляет пользователю механизм создания Web-страниц, а также позволяет обрабатывать иные Web-запросы на одном из внутренних языков СКАДА (JavaLikeCalc), не прибегая к низкоуровневому программированию. Он также является модулем подсистемы «Транспортного протокола» HTTP. Вызов `WebUser` производится из модуля HTTP подсистемы «Транспортные протоколы» посредством расширенного механизма коммуникации через функции: `HttpGet()` и `HttpSet()`.

Рабочий пользовательский интерфейс (WEB) (WebVision) предоставляет механизм конечной визуализации среды визуализации и управления в систему «СКАДА А-СОФТ». Модуль основан на WEB технологиях: XHTML, JavaScript, CSS, AJAX. В своей работе

модуль использует данные движка СВУ. Данный модуль непосредственной визуализации СВУ предназначен только для исполнения интерфейсов СВУ в среде WEB-технологий.

Интерфейс пользователя формируется в WEB-браузере путём обращения к WEB-серверу и получения от него XHTML-документа по протоколу HTTP. В роли WEB-сервера выступает СКАДА система, которая поддерживает стандартные коммуникационные механизмы TCP-сетей (модуль Сокеты подсистема «Транспорты»), протокол передачи гипертекста (модуль HTTP подсистема «Транспортные протоколы»), а также шифрование трафика между браузером и сервером (модуль SSL подсистема «Транспорты»). Для получения доступа к интерфейсу конфигурирования необходимо настроить транспорт сокет или SSL в связке с протоколом HTTP. По умолчанию интерфейс модуля доступен по URL: <http://localhost:10002> или <http://localhost:10004>.

3.11 Подсистема "Специальные"

3.11.1 *Общее описание*

Подсистема "Специальные" является модульной и предназначена для добавления в систему СКАДА непредусмотренных функций путём модульного расширения. Например, тесты системы и её модулей или библиотеки функций пользовательского программирования.

СКАДА содержит среду программирования, позволяющую на уровне пользователя реализовывать:

- алгоритмы управления технологическими процессами;
- крупные динамические модели реального времени технологических, химических, физических и других процессов;
- адаптивные механизмы управления по моделям;
- пользовательские процедуры управления внутренними функциями системы, её подсистемами и модулями;
- гибкое формирование структур параметров на уровне пользователя, с целью создания параметров нестандартной структуры и заполнения её по алгоритму пользователя;
- вспомогательные вычисления.

Среда программирования системы СКАДА представляет собой комплекс средств, организующих вычислительное окружение пользователя. В состав комплекса средств входят:

- объектная модель системы;
- модули библиотек функций;
- вычислительные контроллеры подсистемы "Сбор данных" и другие вычислители.

Модули библиотек функций предоставляют множество функций расширяющих объектную модель системы. Библиотеки могут реализоваться как набором функций фиксированного типа, так и функциями, допускающими свободную модификацию и дополнение.

Библиотеки функций фиксированного типа могут предоставляться стандартными модулями системы, органично дополняя объектную модель. Функции таких библиотек будут представлять собой интерфейс доступа к средствам модуля на уровне пользователя. Например, «Среда визуального представления данных» может предоставлять функции для выдачи различных сообщений. Используя эти функции, пользователь может реализовывать интерактивные алгоритмы взаимодействия с системой.

Библиотеки функций свободного типа предоставляют среду написания пользовательских функций на одном из языков программирования. В рамках модуля библиотек функций могут предоставляться механизмы создания библиотек функций. Так, можно создавать библиотеки аппаратов технологических процессов и использовать их путём связывания.

На основе функций, предоставляемых объектной моделью, строятся вычислительные контроллеры, которые выполняют связывание функций с параметрами системы и механизмом вычисления.

Для конфигурации подсистемы предусмотрена корневая страница подсистемы "Специальные", содержащая вкладки "Модули" и "Помощь". Вкладка "Модули" содержит список модулей подсистемы:

- Библиотека функций Complex 1;
- Библиотека математических функций;
- Функции системного API;
- Тесты системы А-СОФТ.

Вкладка "Помощь" содержит краткую помощь для данной страницы.

Каждый модуль подсистемы "Специальные" предоставляет конфигурационную страницу с вкладками "Специальный модуль" и "Помощь". Вкладка "Специальный модуль" (рисунок 105) предоставляет параметр для контроля за состоянием "Выполняется" модуля, а также разделы конфигурации, специализированные для модулей этой подсистемы. Во вкладке "Помощь" содержится информация о модуле подсистемы "Специальные".

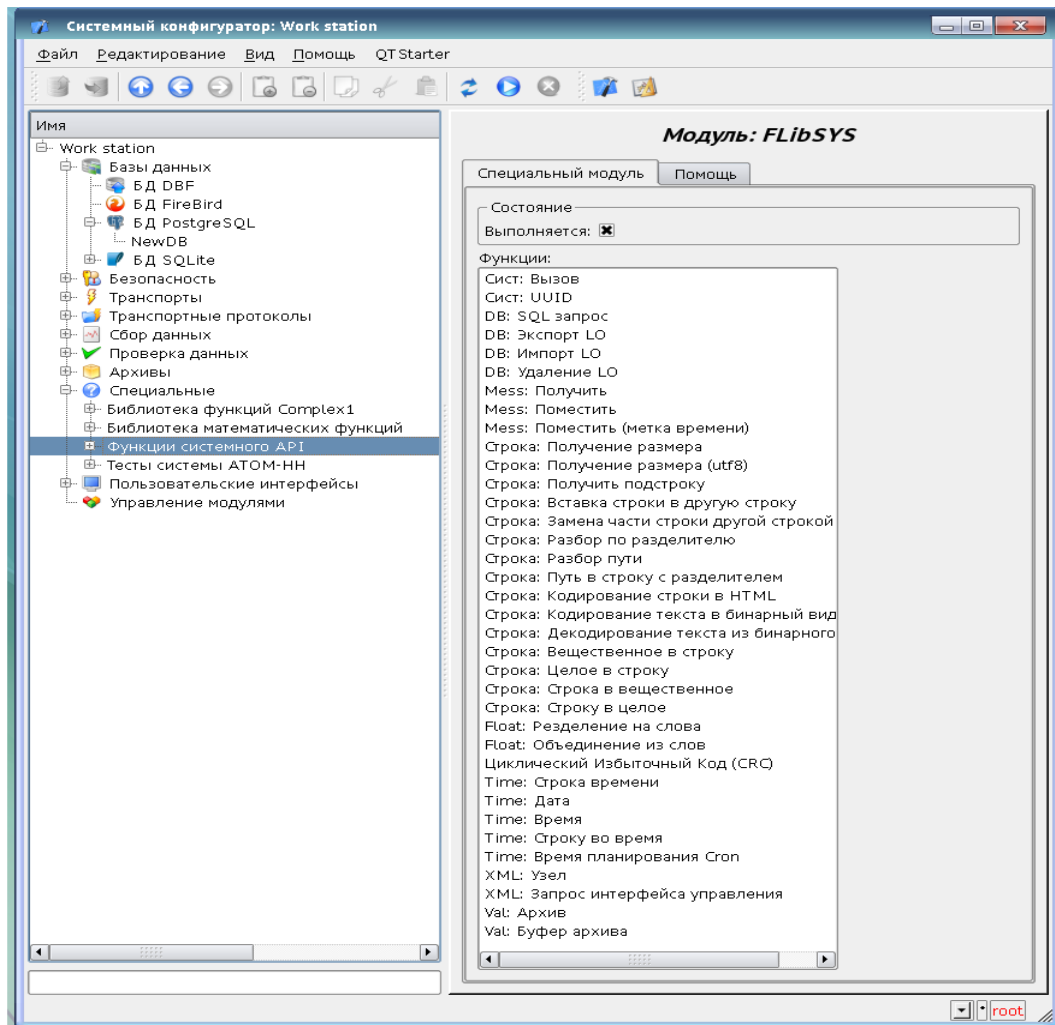


Рисунок 105

3.11.2 Библиотека стандартных математических функций

Специальный модуль FLibMath предоставляет в систему библиотеку стандартных математических функций.

Для адресации к функциям этой библиотеки необходимо использовать путь:

<Special.FLibMath.*>. Где '*' - идентификатор функции в библиотеке.

Вкладка «Специальный модуль» содержит список стандартных математических функций, доступных для использования (рисунок 106).

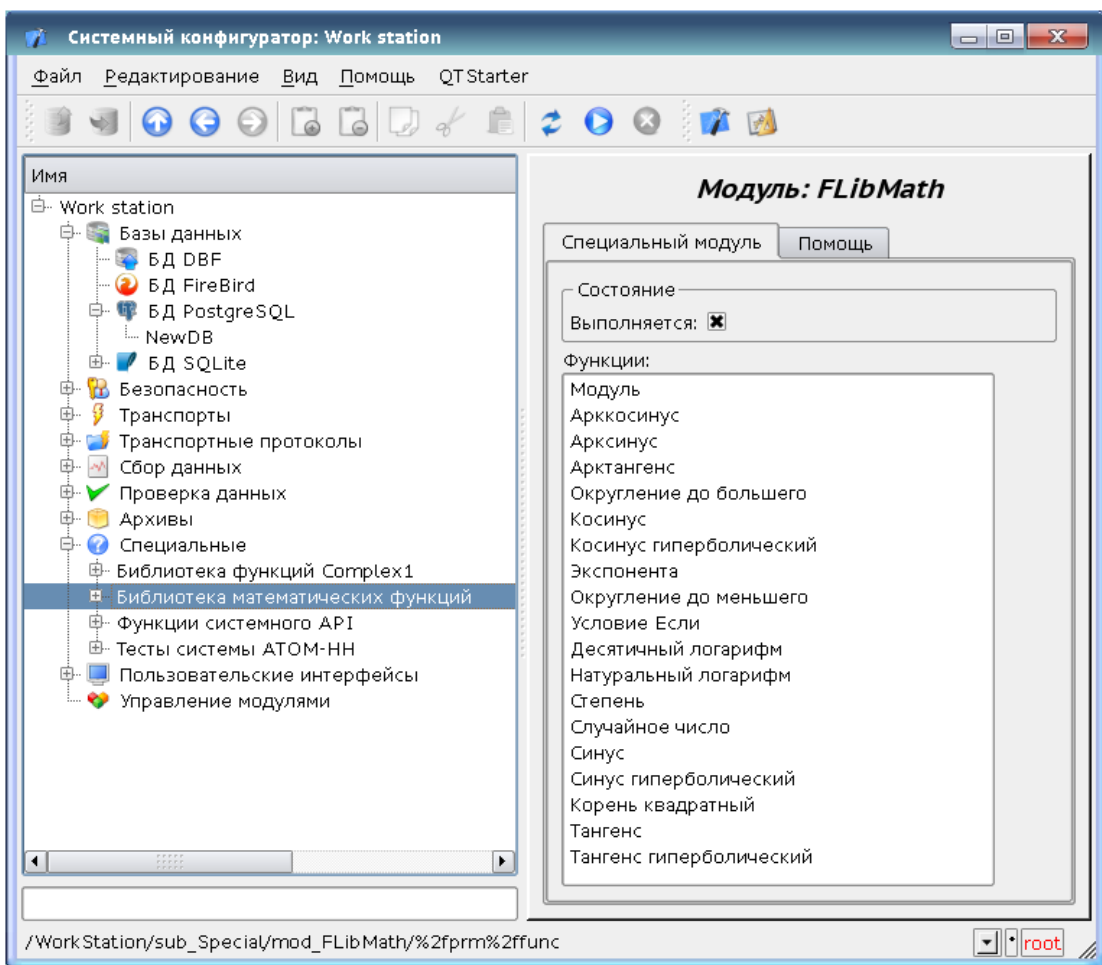


Рисунок 106

3.11.3 Библиотека функций системного API

Специальный модуль FLibSYS предоставляет в СКАДА статическую библиотеку функций для работы со СКАДА-системой, на уровне её системного API. Эти функции могут использоваться в среде пользовательского программирования СКАДА для организации нестандартных алгоритмов взаимодействия.

Для адресации к функциям этой библиотеки необходимо использовать путь:

<Special.FLibSYS.*>. Где '*' - идентификатор функции в библиотеке.

На рисунке 107 представлен вид вкладки «Специальный модуль», которая содержит список функций системного API СКАДА. Описание функций представлено в части 5 настоящего руководства.

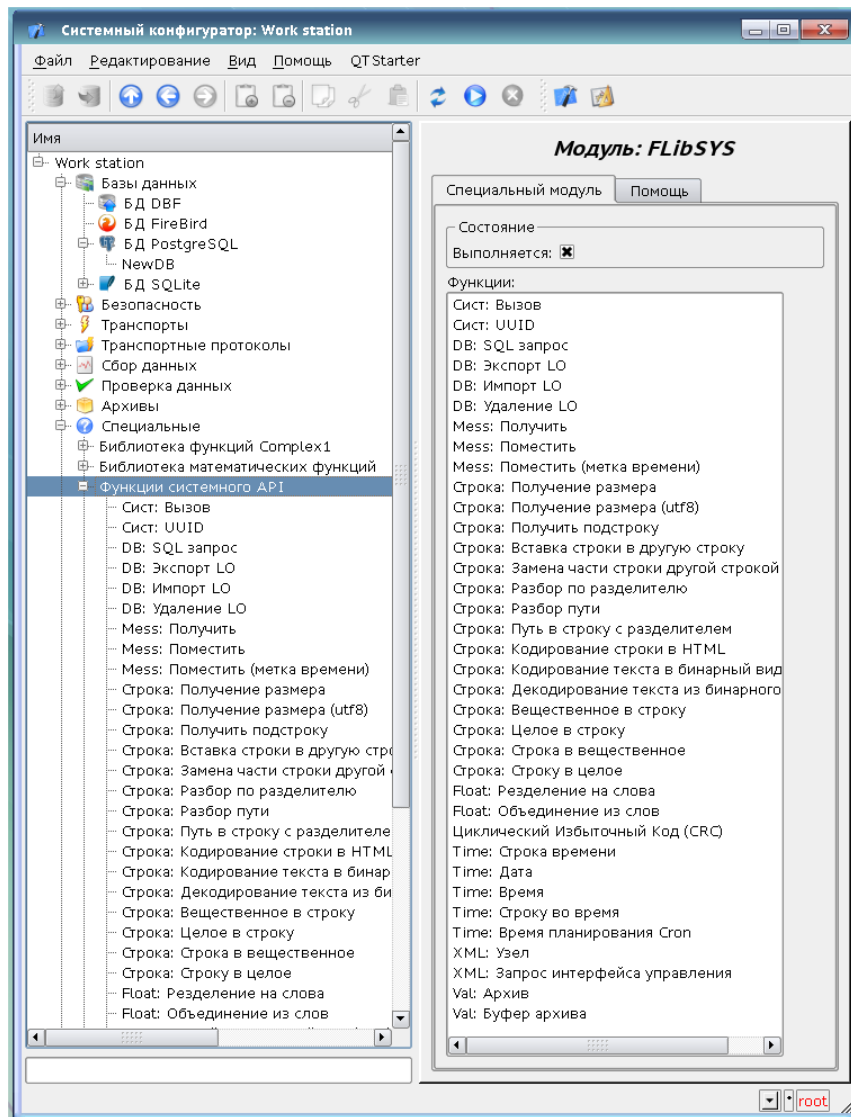


Рисунок 107

3.11.4 Тесты ПП «СКАДА А-СОФТ»

Специальный модуль SystemTests содержит набор тестов, предназначенных для тестирования различных подсистем и узлов СКАДА. Тесты выполнены в виде функций пользовательского API. Следовательно, тесты можно запускать как единовременно, во вкладке "Исполнить" страницы объекта функции, так и из пользовательских процедур, передавая в них нужные аргументы.

Кроме механизмов обычного исполнения функций пользовательского API предусмотрен автономный механизм. Этот механизм представлен отдельной задачей, исполняющейся с периодом в одну секунду, в которой осуществляется вызов функций тестов в соответствии с настройками в конфигурационном файле.

Конфигурационные поля тестов помещаются в секцию модуля SystemTests подсистемы «Специальные». Формат конфигурационных полей:

```
<prm id="Test Id" on="1" per="10" />
```

Где:

- id - идентификатор теста;
- on - признак “Тест включен”;
- per - период повторения теста (секунд).

Кроме основных атрибутов осуществляется отражение входных параметров функций тестов на одноимённые атрибуты тега "prm". Например, атрибут "name" функции "Param" можно указать в теге "prm".

Допускается указание множества тегов "prm" для одного или разных тестов с одинаковыми или различными параметрами, указывая тем самым на отдельный запуск теста с указанными параметрами.

Вид вкладки «Тесты», содержащей список доступных в системе тестов, представлен на рисунке 108.

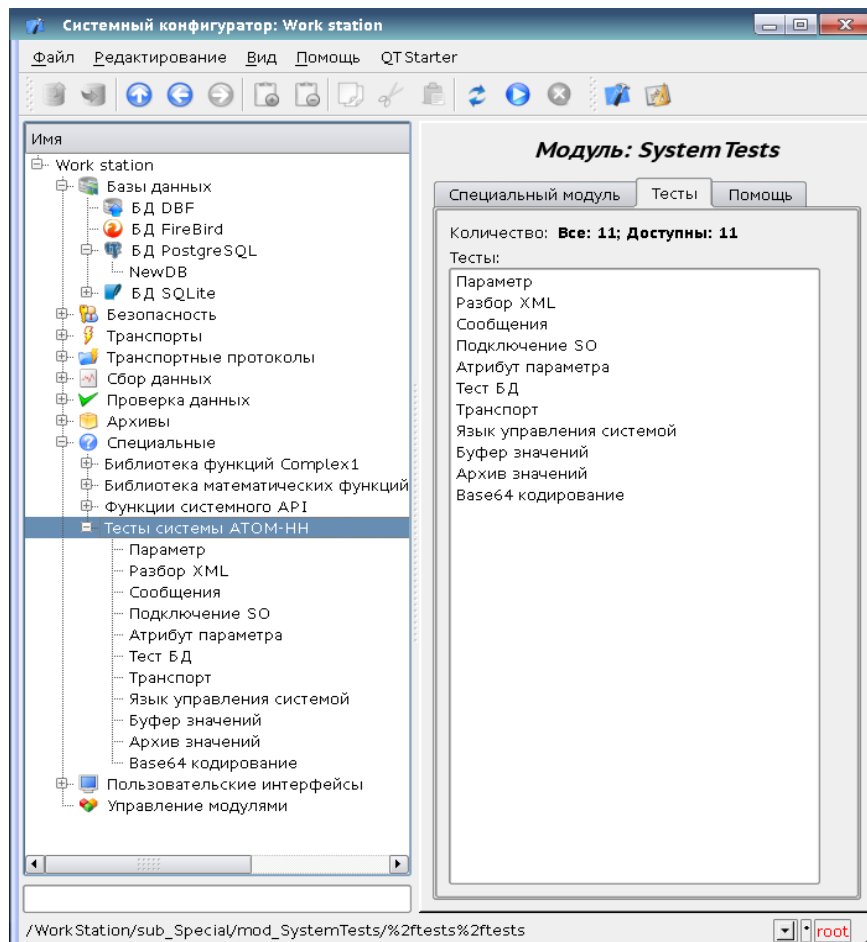


Рисунок 108

3.12 Подсистема "Управление модулями"

Вид страницы подсистемы "Управление модулями" показан на рисунке 109.

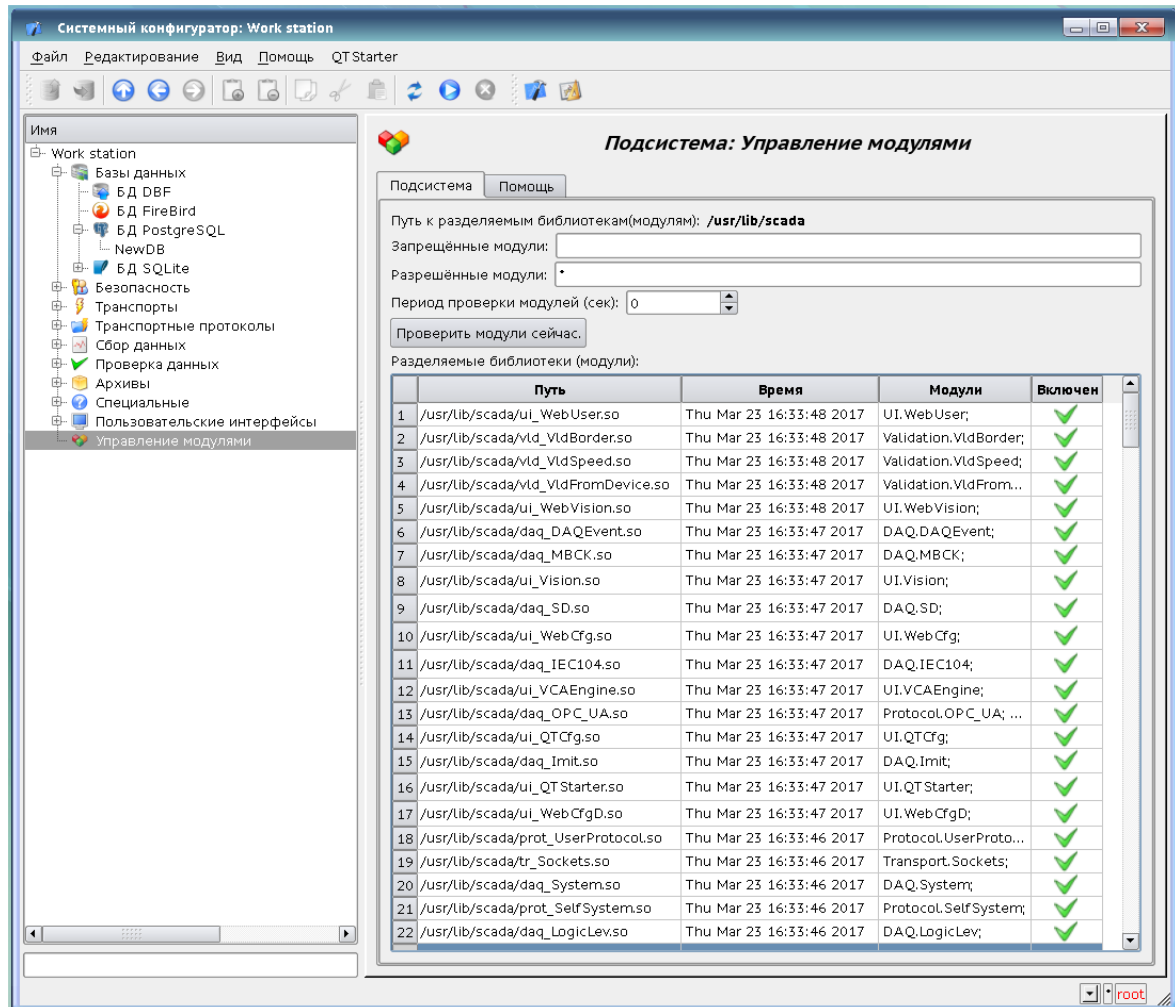


Рисунок 109

Подсистема не является модульной. Для конфигурации подсистемы предусмотрена страница подсистемы "Управление модулями", содержащая вкладки "Подсистема" и "Помощь". Вкладка "Подсистема" содержит основные настройки подсистемы. Вкладка "Помощь" содержит краткую помощь для данной страницы.

Состав вкладки "Подсистема":

- *Путь к разделяемым библиотекам (модулям)* - информация о расположении директории с модулями СКАДА. Устанавливается параметром *<ModDir>* станции, конфигурационного файла;

- *Запрещённые модули* - информация о списке модулей, запрещённых для автоматического подключения и обновления. Разделителем списка является символ ';'. Допускается пустое значение этого поля. Устанавливается параметром *<ModDeny>* раздела

подсистемы "sub_ModSched" в конфигурационном файле СКАДА. Список запрещённых модулей имеет больший приоритет, чем список разрешённых;

- *Разрешённые модули* - информация о списке модулей, разрешённых для автоматического подключения и обновления. Разделителем списка является символ ';'. Значение '*' используется для разрешения всех модулей. Устанавливается параметром <ModAllow> раздела подсистемы "sub_ModSched" в конфигурационном файле СКАДА;

- *Период проверки модулей (сек)* - указывает на периодичность проверки модулей на факт их обновления. Модули, допустимые для автоматического подключения и обновления, будут автоматически обновлены;

- *Проверить модули сейчас* - команда выполнить проверку модулей на факт их обновления. Модули, допустимые для автоматического подключения и обновления, будут автоматически обновлены.

- *Разделяемые библиотеки (модули)* - таблица с перечнем разделяемых библиотек с модулями, обнаруженными СКАДА. В строках расположены модули, а в колонках информация о них:

- *Путь* - информация о полном пути к разделяемой библиотеке;

- *Время* - информация о времени последней модификации файла разделяемой библиотеки;

- *Модули* - информация о перечне модулей в разделяемой библиотеке;

- *Включен* - состояние "Включен" разделяемой библиотеки. Привилегированным пользователям предоставляется возможность ручного включения/выключения разделяемых библиотек путём изменения этого поля.

3.13 Конфигурационный файл СКАДА и параметры командной строки вызова СКАДА

Конфигурационный файл СКАДА предназначен для хранения системной и общей конфигурации СКАДА-станции. Только в конфигурационном файле и через параметры командной строки можно указать часть ключевых системных параметров станции.

Называться конфигурационный файл СКАДА может произвольно, однако принято название `scada.xml` и производные от него. Конфигурационный файл может быть указан при запуске станции параметром командной строки:

```
--Config=/ path/scada.xml
```

где `path` – путь к конфигурационному файлу.

Если конфигурационный файл не указан, то используется стандартный конфигурационный файл: `/etc/scada.xml`.

Структурно конфигурационный файл организован на расширяемом языке разметки текста XML. Следовательно, требуется жёсткое соблюдение правил синтаксиса XML. Пример образца типового конфигурационного файла СКАДА, с узлами конфигурации большинства компонентов СКАДА, приведен в приложении 1.

Один конфигурационный файл может содержать конфигурацию нескольких станций в секциях `<station id="DemoStation"/>`.

Атрибутом указывается идентификатор станции. Использование той или иной секции станции, при вызове, указывается параметром командной строки `--Station=DemoStation`. Секция станции непосредственно содержит параметры станции и секции подсистем. Параметры конфигурации секции записываются в виде `<prm id="StName">Demo station</prm>`. Где в атрибуте `<id>` указывается идентификатор атрибута, а в теле тега указывается значение параметра "Demo station".

Перечень доступных параметров и их описание для станции и всех остальных секций можно получить в консоли, посредством вызова СКАДА с параметром `--help` или во вкладках "Помощь" страниц компонентов конфигурационных файлов СКАДА (рисунок 110).

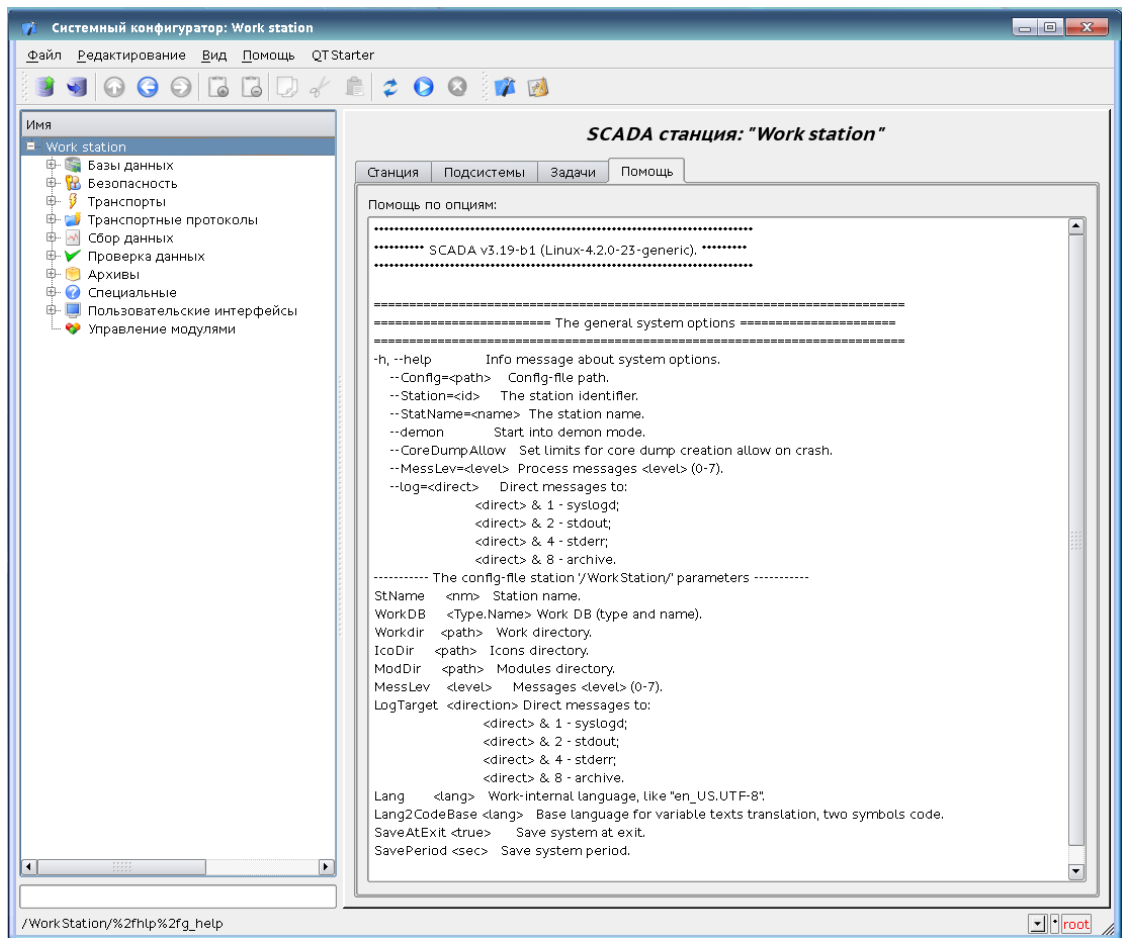


Рисунок 110

Результат вызова команды: # ./scada --help приведен в приложении 2.

Секции подсистем (`<node id="sub_DAQ" />`) содержат параметры подсистемы, секции модулей и секции таблиц отражения данных баз данных в конфигурационном файле. Секции модулей (`<node id="mod_DiamondBoards" />`) содержат индивидуальные параметры модулей и секции таблиц отражения данных баз данных в конфигурационном файле.

Секции таблиц отражения данных баз данных предназначены для размещения в конфигурационном файле записей таблиц БД для компонентов СКАДА. Рассмотрим таблицу входящих транспортов "Transport_in" подсистемы транспорты (`<node id="sub_Transport">`) из примера конфигурационного файла выше. Таблица содержит две записи с полями: ID, MODULE, NAME, DESCRIPT, ADDR, PROT, START. После загрузки с такой секцией и вообще без БД в подсистеме "Транспорты" модуля "Sockets" появятся два входных транспорта. Форматы структур таблиц основных компонентов включены в демонстрационные конфигурационные файлы.

4. ДОПОЛНИТЕЛЬНЫЕ ВОЗМОЖНОСТИ ПОЛЬЗОВАТЕЛЬСКОГО ИНТЕРФЕЙСА

4.1 Описание журнала активных тревог и журнала тревог и событий

4.1.1 *Функциональное назначение*

СКАДА обеспечивает выполнение следующих функций:

- отображение всех активных тревог в окне "Журнала активных тревог";
- отображение всех тревог и событий в окне "Журнале тревог и событий" с использованием определенных критериев выбора;
- выполнение квитирования аварийных или предупредительных тревог, имеющих в "Журнале активных тревог";
- сохранение информации о квитировании тревог в "Журнал тревог и событий".

4.1.2 *Описание журнала активных тревог и журнал тревог и событий*

Журнал тревог и событий содержит таблицу с сообщениями, полученными за заданный пользователем промежуток времени. Журнал активных тревог отображает активные квитированные и не квитированные тревоги до тех пор, пока атрибуты, вызвавшие сообщение, не войдут в норму.

Существует возможность применения фильтра по типам отображаемых сообщений и установка максимального количество отображаемых сообщений. В журнале активных тревог реализована возможность квитирования выбранного пользователем сообщения и перехода к видеокадру, содержащему элемент с вызвавшим тревогу атрибутом.

4.1.3 *Взаимодействие журнала активных тревог и журнала тревог и событий с модулями СКАДА*

Журнал активных тревог и журнал тревог и событий входят в модуль отображения ЭДЖ. Взаимодействие журнала активных тревог и журнала тревог и событий с модулями СКАДА представлено на рисунке 111. Средствами графического интерфейса этого модуля происходит конфигурация журналов: настройка фильтров типов отображаемых сообщений и максимальное количество строк в таблицах журналов.

Модуль отображения ЭДЖ создает подписку журнал активных тревог на получения сообщений о тревогах от менеджера сообщений модуля архивации. Менеджер сообщений передает вновь поступившие в буфер сообщения о тревогах в журнал активных тревог, также он сообщает о том, что вызвавший тревогу атрибут вошел в норму. Таким образом, после

квотирования сообщение либо удалится, если ранее менеджер сообщений передал информацию о том, что атрибут вошел в норму, либо окрасит его в другой цвет (удаление данного сообщения из журнала произойдет после того как атрибут к которому относится сообщение войдет в норму).

Журнал тревог и событий получает данные за заданный промежуток времени от модуля архивации. При этом если указанный интервал времени присутствует в буфере сообщений, то значение берется из буфера. В противном случае данные запрашиваются у базы данных архивов.

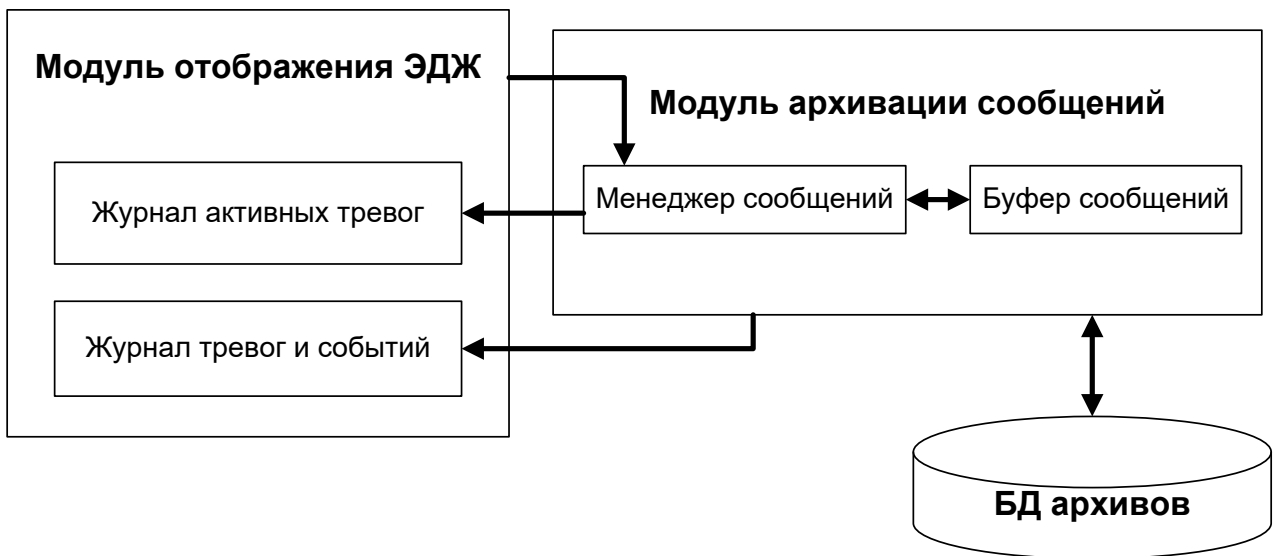


Рисунок 111

4.1.4 Элементы интерфейса журнала активных тревог и журнал тревог и событий

Для работы с журналом активных тревог и журналом тревог и событий интерфейс ЭДЖ содержит кнопки открытия журналов ("1" на рисунке 112) и вкладки, содержащие журнал активных тревог и журнал тревог и событий ("2" на рисунке 112).

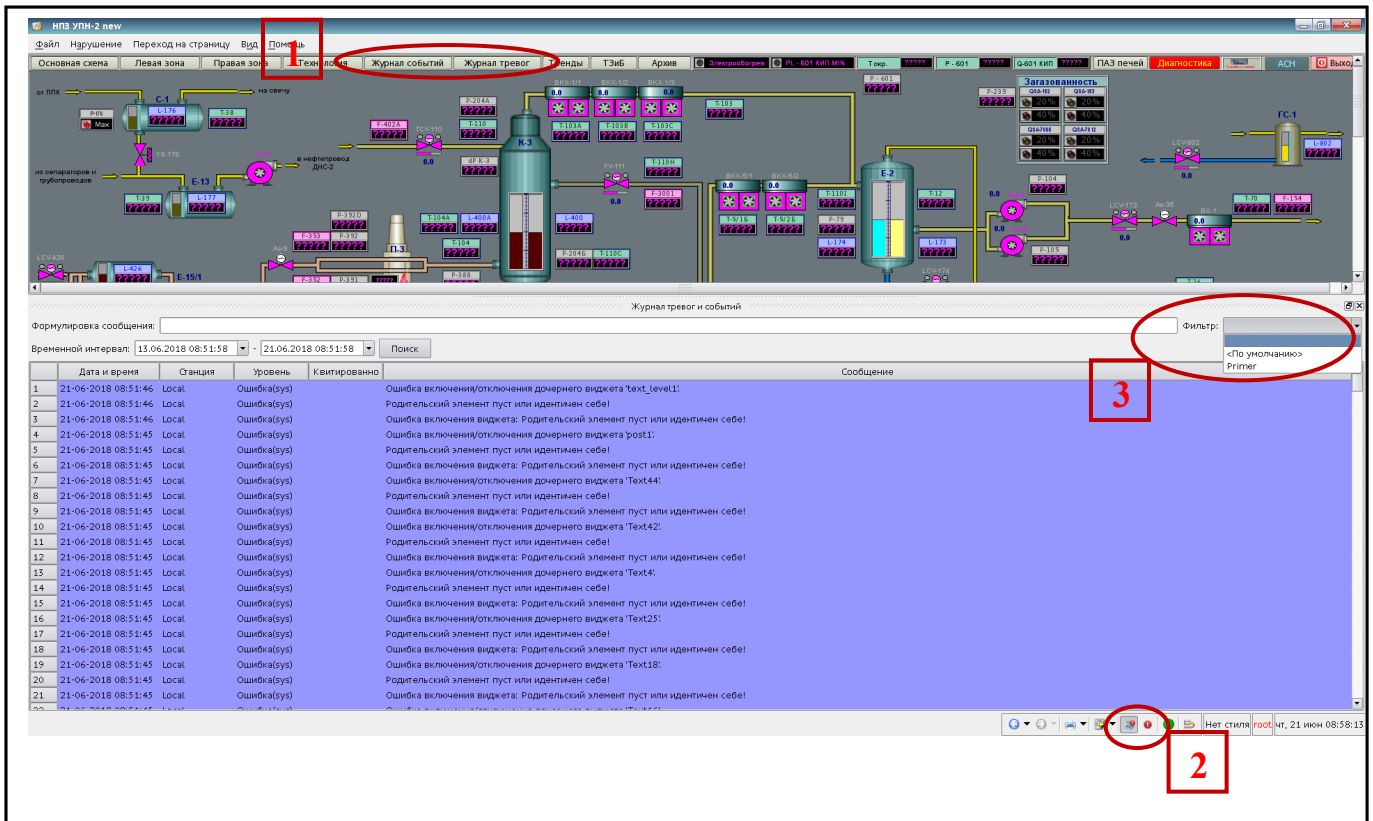


Рисунок 112

В строке журнала тревог и событий по умолчанию отображается следующая информация:

- дата и время;
- станция;
- уровень (ошибка, критическое сообщение, предупреждение, информационное сообщение);
- оборудование;
- отметка квитирования;
- пользователь;
- текст сообщения.

Для определения порядка отображения столбцов в окне журнала тревог и событий из контекстного меню необходимо выбрать строку «Настройка» и проставить крестик в чек-боксе напротив выводимых столбцов (рисунок 113).

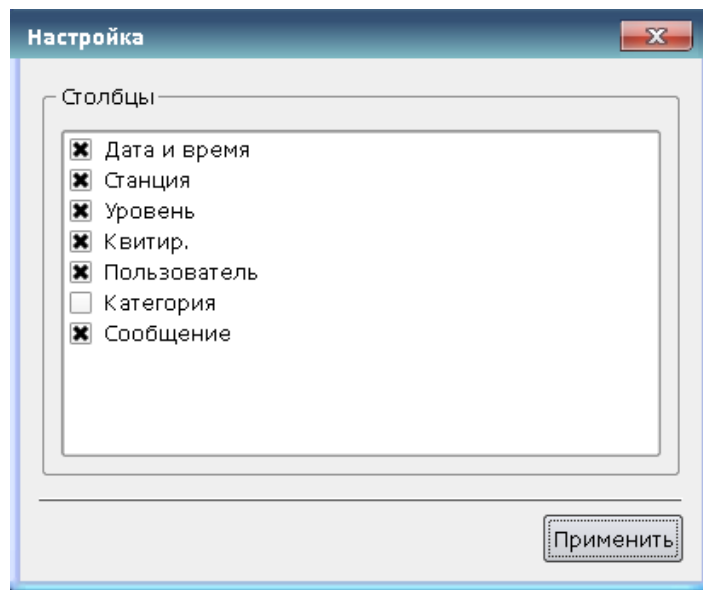


Рисунок 113

В правой части окна журнала тревог и событий находится поле «Фильтр» (см. "3" на рисунке 112), в котором можно настроить типы выводимых сообщений. На рисунке 114 показана настройка фильтра “Primer”. Добавление фильтра производится выбором строки «Добавить» контекстного меню и вводом ID и имени нового фильтра. После чего необходимо выбрать уровни (типы) воспроизводимых сообщений (поставить крестик в соответствующем чек-боксе), сохранить изменения и закрыть окно редактирования фильтра. Для просмотра результата работы фильтра необходимо выбрать временной интервал и нажать кнопку «Поиск».

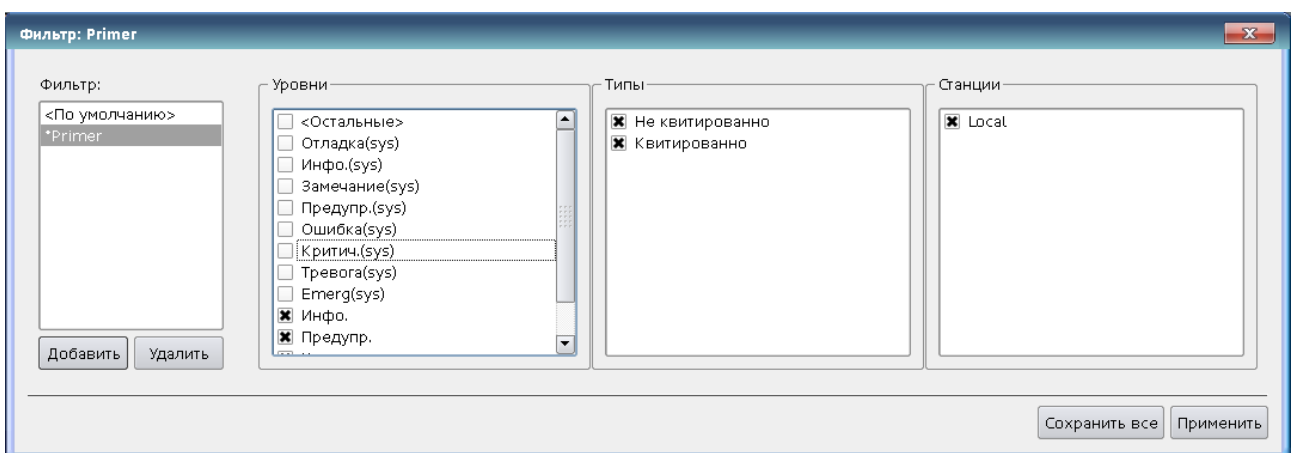


Рисунок 114

Журнал активных тревог служит для просмотра активных тревог и квитирования любой из них. Тревоги от одного источника отображаются вложенной строкой, которую при необходимости можно развернуть.

В строке тревоги отображаются:

- дата и время;
- уровень (ошибка, критическое сообщение);
- оборудование;
- текст сообщения о тревоге.

Тревоги отображаются цветом следующим образом:

тревога активна и не квитирована	– красный фон с черной мигающей рамкой;
тревога активна и квитирована	– красный фон;
тревога вошла в норму, но не квитирована	– белый фон с черной мигающей рамкой.

Квитированная тревога, вошедшая в норму, удаляется из журнала активных тревог.

Квитирование тревоги производится из контекстного меню квитированной строки.

Для квитирования сообщения необходимо нажать на него правой кнопкой мыши и в появившемся меню выбрать пункт «Квитировать» (рисунок 115). Квитированное сообщение либо изменит цвет, если значение атрибута не вернулось к моменту квитирования в норму, либо удалится из таблицы в противном случае. Если строка имеет вложенные тревоги, то можно в контекстном меню выбрать строку «квитировать все» и будут квитированы все вложенные тревоги.

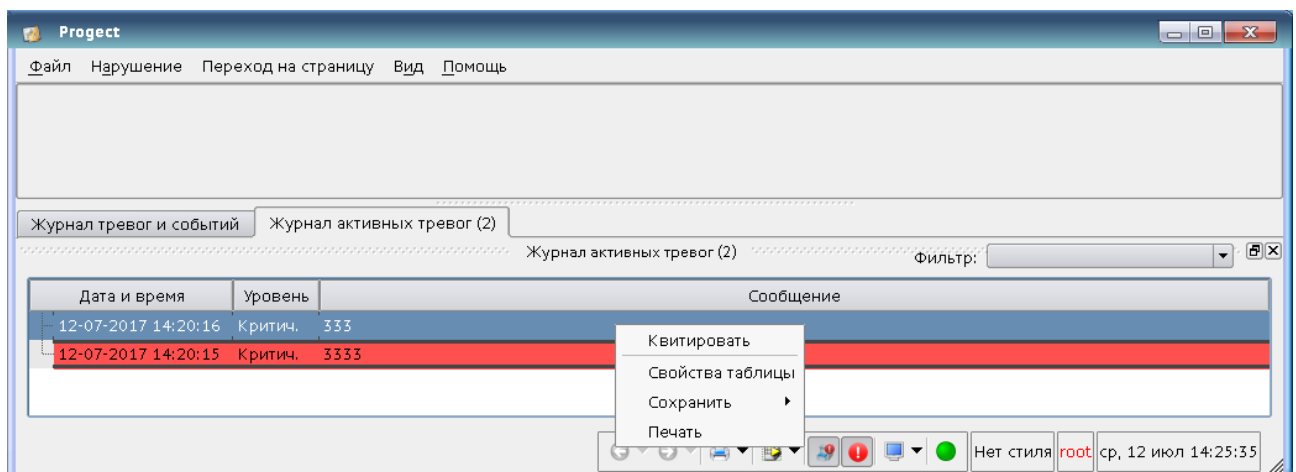


Рисунок 115

Кроме того, обеспечена возможность сохранения данных о тревоге в log-файл, при выборе соответствующего пункта в контекстном меню.

Конфигурирование журнала активных тревог и журнала тревог и событий осуществляется средствами редактора пользовательского интерфейса в окне редактирования свойств визуального элемента «Проект» – «Журнал сообщений».

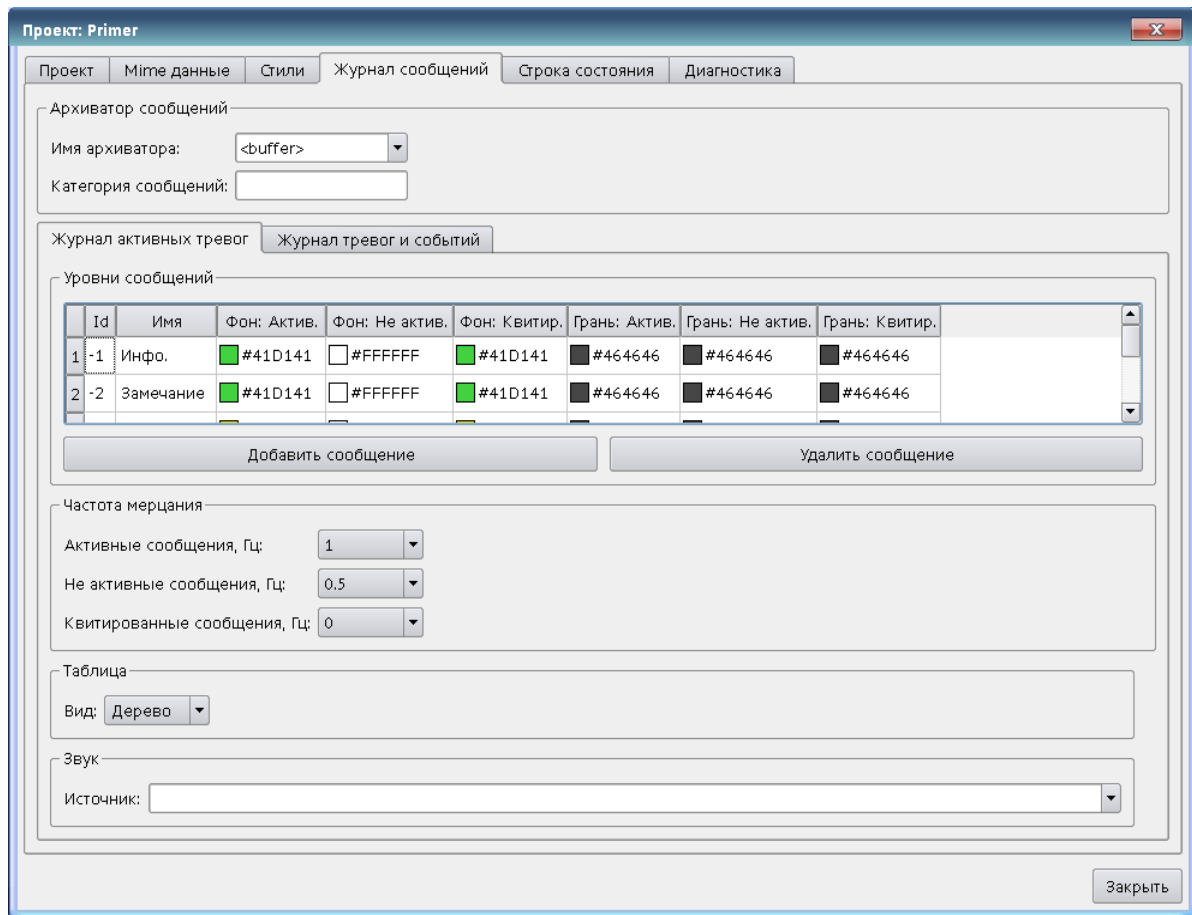


Рисунок 116

Для этого необходимо выбрать вкладку «Журнал сообщений» и настроить атрибуты «Журнала активных тревог» и «Журнала тревог и событий» в зависимости от поставленной задачи, например, добавить сообщения, изменить частоту и цвет мерцания сообщения (рисунок 116).

4.2 Описание быстрого перехода по видеокадрам

4.2.1 Функциональное назначение

Разработанное ПО обеспечивает выполнение следующих функций:

- создание списка истории посещенных видеокадров;
- предоставление возможности пользователю перехода на выбранный из ранее сформированного списка видеокадр;
- разделение доступа к спискам истории для разных пользователей.

4.2.2 Элементы интерфейса для быстрого перехода по видеокадрам

Для обеспечения возможности предоставления пользователю функций перехода по видеокадрам в строке состояния интерфейса пользователя находятся кнопки перехода (рисунок 117) на предыдущий или на следующий видеокадр (если ранее был осуществлен переход на предыдущий видеокадр), а также возможность выбора конкретного видеокадра из списка посещенных видеокадров.

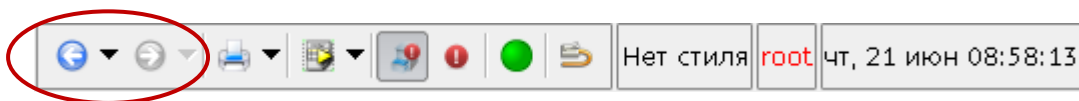


Рисунок 117

4.2.3 Основные особенности быстрого перехода между видеокадрами

История посещения видеокадров обновляется в момент открытия новой мнемосхемы и хранится до завершения сеанса. Для каждого подключенного пользователя (с разных станций оператора) к серверу формируется своя история посещенных видеокадров. Для того чтобы один пользователь не мог получить доступ к истории посещения видеокадров другого пользователя (на той же самой станции оператора) происходит очищение списка истории при каждой смене пользователя.

ПРИЛОЖЕНИЕ 1

Типовой конфигурационный файл СКАДА

```
<?xml version="1.0" encoding="UTF-8" ?>
<SCADA>
<!-- This is the SCADA configuration file. -->
<station id="DemoStation">
<!-- Discribe internal parameter for station. Station this only SCADA
programm. -->
<prm id="StName">Demo station</prm>
<prm id="StName_ru">Демо станция</prm>
<prm id="WorkDB">SQLite.GenDB</prm>
<prm id="Workdir">~/SCADA</prm>
<prm id="IcoDir">./icons</prm>
<prm id="ModDir">/usr/lib/SCADA</prm>
<prm id="LogTarget">10</prm>
<prm id="MessLev">0</prm>
<prm id="Lang2CodeBase">en</prm>
<prm id="SaveAtExit">0</prm>
<prm id="SavePeriod">0</prm>
<node id="sub_BD">
<prm id="SYSStPref">0</prm>
<tbl id="DB">
<fld ID="GenDB" TYPE="SQLite" NAME="Generic DB" NAME_ru="Основная БД"
ADDR="./DEMO/DemoSt.db" CODEPAGE="UTF-8"/>
</tbl>
</node>
<node id="sub_Security">
<!--
<tbl id="Security_user">
<fld NAME="root" DESCR="Super user" DESCR_ru="Супер пользователь"
PASS="SCADA"/>
<fld NAME="user" DESCR="System user" DESCR_ru="Системный пользователь"
PASS=""/>
</tbl>
<tbl id="Security_grp">
<fld NAME="root" DESCR="Super users groups" DESCR_ru="Группа
суперпользователей" USERS="root;user"/>
</tbl>
</node>
<node id="sub_ModSched">
```

```
<prm id="ModAllow">*</prm>
<prm id="ModDeny"></prm>
<prm id="ChkPer">0</prm>
</node>
<node id="sub_Transport">
  <!--
  <tbl id="Transport_in">
    <fld ID="WEB_1" MODULE="Sockets" NAME="Generic WEB interface"
NAME_ru="Основной WEB интерфейс" DESCRIPT="Generic transport for WEB interface."
DESCRIPT_ru="Основной транспорт для WEB интерфейса." ADDR="TCP::10002:0"
PROT="HTTP" START="1"/>
    <fld ID="WEB_2" MODULE="Sockets" NAME="Reserve WEB interface"
NAME_ru="Резервный WEB интерфейс" DESCRIPT="Reserve transport for WEB
interface." DESCRIPT_ru="Резервный транспорт для WEB интерфейса."
ADDR="TCP::10004:0" PROT="HTTP" START="1"/>
  </tbl>
  <tbl id="Transport_out">
    <fld ID="testModBus" MODULE="Sockets" NAME="Test ModBus" NAME_ru="Тест
ModBus" DESCRIPT="Data exchange by protocol ModBus test." DESCRIPT_ru="Тест
обмена по протоколу ModBus." ADDR="TCP:localhost:10502" START="1"/>
  </tbl>
</node>
<node id="sub_DAQ">
  <!--
  <tbl id="tmplib">
    <fld ID="test2" NAME="Test 2" NAME_ru="Тест 2" NAME_DESCR="" DESCR_ru=""
DESCR_DB="tmplib_test2"/>
  </tbl>
  <tbl id="tmplib_test2">
    <fld ID="test2" NAME="Test 2" NAME_ru="Тест 2" DESCR="" DESCR_ru=""
DB="test2" PROGRAM="JavaLikeCalc.JavaScript&#010;cnt=5*i"/>
  </tbl>
  <tbl id="tmplib_test2_io">
    <fld TMPL_ID="test2" ID="i" NAME="I" NAME_ru="I" TYPE="4" FLAGS="160"
VALUE="" POS="0"/>
    <fld TMPL_ID="test2" ID="cnt" NAME="Cnt" NAME_ru="Cnt" TYPE="4"
FLAGS="32" VALUE="" POS="0"/>
  </tbl>
<node id="mod_LogicLev">
  <!--
  <tbl id="DAQ">
```

```
<fld ID="test2" NAME="Test 2" NAME_ru="Тест 2" DESCR="" DESCR_ru=""
ENABLE="1" START="1" PRM_BD="test2prm" PERIOD="1000" PRIOR="0"/>
</tbl>
<tbl id="test2prm">
<fld SHIFR="test2" NAME="Test 2" NAME_ru="Тест 2" DESCR="" DESCR_ru=""
EN="1" MODE="2" PRM="test2.test2"/>
</tbl>
</node>
<node id="mod_System">
<!--
<tbl id="DAQ">
<fld ID="DataOS" NAME="Data OS" NAME_ru="Данные ОС" DESCR="Data of
services and subsystems OS." DESCR_ru="Данные сервисов и подсистем ОС."
ENABLE="1" START="1" AUTO_FILL="0" PRM_BD="DataOSprm" PERIOD="1000" PRIOR="0"/>
</tbl>
<tbl id="DataOSprm">
<fld SHIFR="CPU" NAME="CPU load" NAME_ru="Нагрузка CPU" DESCR=""
DESCR_ru="" EN="1" TYPE="CPU" SUBT="gen"/>
<fld SHIFR="MEM" NAME="Memory" NAME_ru="Память" DESCR="" DESCR_ru=""
EN="1" TYPE="MEM"/>
</tbl>
</node>
<node id="mod_DiamondBoards">
<!--
<tbl id="DAQ">
<fld ID="Athena" NAME="Athena board" NAME_ru="Плата Athena" DESCR=""
DESCR_ru="" ENABLE="1" START="0" BOARD="25" PRM_BD_A="AthenaAnPrm"
PRM_BD_D="AthenaDigPrm" ADDR="640" INT="5" DIO_CFG="0" ADMODE="0" ADRANGE="0"
ADPOLAR="0" ADGAIN="0" ADCONVRATE="1000"/>
</tbl>
<tbl id="AthenaAnPrm">
<fld SHIFR="ai0" NAME="AI 0" NAME_ru="AI 0" DESCR="" DESCR_ru="" EN="0"
TYPE="0" CNL="0" GAIN="0"/>
</tbl>
<tbl id="AthenaDigPrm">
<fld SHIFR="di0" NAME="DI 0" NAME_ru="DI 0" DESCR="" DESCR_ru="" EN="0"
TYPE="0" PORT="0" CNL="0"/>
</tbl>
</node>
<node id="mod_BlockCalc">
<!--
<tbl id="DAQ">
```

```
<fld ID="Model" NAME="Model" NAME_ru="Модель" DESCR="" DESCR_ru=""
ENABLE="1" START="1" PRM_BD="Model_prm" BLOCK_SH="Model_blcks" PERIOD="1000"
PRIOR="0" PER_DB="0" ITER="1"/>
</tbl>
<tbl id="Model_blcks">
<fld ID="Klap" NAME="Klapan" NAME_ru="Клапан" DESCR="" DESCR_ru=""
FUNC="DAQ.JavaLikeCalc.lib_techApp.klap" EN="1" PROC="1"/>
</tbl>
<tbl id="Model_blcks_io">
<fld BLK_ID="Klap" ID="1_kl1" TLNK="0" LNK="" VAL="50"/>
<fld BLK_ID="Klap" ID="1_kl2" TLNK="0" LNK="" VAL="20"/>
</tbl>
<tbl id="Model_prm">
<fld SHIFR="1_kl" NAME="Klap lev" NAME_ru="Полож. клапана" DESCR=""
DESCR_ru="" EN="1" BLK="Klap" IO="1_kl1"/>
</tbl>
</node>
<node id="mod_JavaLikeCalc">
<!--
<tbl id="DAQ">
<fld ID="CalcTest" NAME="Calc Test" NAME_ru="Тест вычисл." DESCR=""
DESCR_ru="" ENABLE="1" START="1" PRM_BD="Cal FUNC="TemplFunc.d_alarm"
PERIOD="1000" PRIOR="0" PER_DB="0" ITER="1"/>
</tbl>
<tbl id="CalcTest_val">
<fld ID="in" VAL="0"/>
<fld ID="alm" VAL=""/>
<fld ID="alm_md" VAL="1"/>
<fld ID="alm_mess" VAL="Error present."/>
</tbl>
<tbl id="CalcTest_prm">
<fld SHIFR="alm" NAME="Alarm" NAME_ru="Авария" DESCR="" DESCR_ru=""
EN="1" FLD="alm"/>
</tbl>
<tbl id="lib">
<fld ID="TemplFunc" NAME="" NAME_ru="" DESCR="" DESCR_ru=""
DB="lib_TemplFunc"/>
</tbl>
<tbl id="lib_TemplFunc">
<fld ID="d_alarm" NAME="Digit alarm" NAME_ru="Авария по дискр." DESCR=""
FORMULA="alm=(in==alm_md)?&quot;1:&quot;+alm_mess:&quot;0&quot;;"/>
</tbl>
```

```
<tbl id="lib_TemplFunc_io">
  <fld F_ID="d_alarm" ID="in" NAME="Input" NAME_ru="Вход" TYPE="3" MODE="0"
DEF="" HIDE="0" POS="0"/>
  <fld F_ID="d_alarm" ID="alarm" NAME="Alarm" NAME_ru="Авария" TYPE="0"
MODE="1" DEF="" HIDE="0" POS="1"/>
  <fld F_ID="d_alarm" ID="alarm_md" NAME="Alarm mode" NAME_ru="Режим аварии"
TYPE="3" MODE="0" DEF="" HIDE="0" POS="2"/>
  <fld F_ID="d_alarm" ID="alarm_mess" NAME="Alarm message" NAME_ru="Сообщ.
аварии" TYPE="0" MODE="0" DEF="" HIDE="0" POS="3"/>
</tbl>
</node>
<node id="mod_Siemens">
  <!--
  <tbl id="DAQ">
    <fld ID="test2" NAME="Test 2" NAME_ru="Тест 2" DESCR="" DESCR_ru=""
ENABLE="1" START="1" PRM_BD="test2prm" PERIOD="1000" PRIOR="0" CIF_DEV="0"
ADDR="5" ASINC_WR="0"/>
  </tbl>
  <tbl id="test2prm">
    <fld SHIFR="test2" NAME="Test 2" NAME_ru="Тест 2" DESCR="" DESCR_ru=""
EN="1" TMPL="S7.ai_man"/>
  </tbl>
</node>
<node id="mod_SNMP">
  <!--
  <tbl id="DAQ">
    <fld ID="test2" NAME="Test 2" NAME_ru="Тест 2" DESCR="" DESCR_ru=""
ENABLE="1" START="1" PRM_BD="test2prm" PERIOD="1000" PRIOR="0" ADDR="localhost"
COMM="public" PATTR_LIM="20"/>
  </tbl>
  <tbl id="test2prm">
    <fld SHIFR="test2" NAME="Test 2" NAME_ru="Тест 2" DESCR="" DESCR_ru=""
EN="1" OID_LS="system"/>
  </tbl>
</node>
<node id="mod_ModBus">
  <!--
  <tbl id="DAQ">
    <fld ID="test2" NAME="Test 2" NAME_ru="Тест 2" DESCR="" DESCR_ru=""
ENABLE="1" START="1" PRM_BD="test2prm" PERIOD="1000" PRIOR="0" TRANSP="Sockets"
ADDR="exlar.diya.org" NODE="1"/>
  </tbl>
```

```
<tbl id="test2prm">
  <fld SHIFR="test2" NAME="Test 2" NAME_ru="Тест 2" DESCR="" DESCR_ru=""
EN="1" ATTR_LS="321:0:tst:Test"/>
</tbl>
</node>
<node id="mod_Transporter">
  <!--
  <tbl id="DAQ">
    <fld ID="test2" NAME="Test 2" NAME_ru="Тест 2" DESCR="" DESCR_ru=""
ENABLE="1" START="1" PRM_BD="test2prm" PERIOD="1000" PRIOR="0" SYNCPER="60"
STATIONS="loop" CNTRPRM="System.AutoDA"/>
    </tbl>
  </node>
</node>
<node id="sub_Archive">
  <prm id="MessBufSize">1000</prm>
  <prm id="MessPeriod">5</prm>
  <prm id="ValPeriod">1000</prm>
  <prm id="ValPriority">10</prm>
  <!--
  <tbl id="Archive_mess_proc">
    <fld ID="StatErrors" MODUL="FSArch" NAME="Errors" NAME_ru="Ошибки"
DESCR="Local errors\' archive" DESCR_ru="Архив локальных ошибок" START="1"
CATEG="/DemoStation*" LEVEL="4" ADDR="ARCHIVES/MESS/stError/" FSArchMSize="300"
FSArchNFiles="10" FSArchTmSize="30" FSArchXML="1" FSArchPackTm="10"
FSArchTm="60"/>
    <fld ID="NetRequsts" MODUL="FSArch" NAME="Net requests" NAME_ru="Сетевые
запросы" DESCR="Requests to server through transport Sockets." DESCR_ru="Запросы
к серверу через транспорт Sockets." START="1"
CATEG="/DemoStation/Transport/Sockets*" LEVEL="1" ADDR="ARCHIVES/MESS/Net/"
FSArchMSize="300" FSArchNFiles="10" FSArchTmSize="30" FSArchXML="1"
FSArchPackTm="10" FSArchTm="60"/>
  </tbl>
  <tbl id="Archive_val_proc">
    <fld ID="1h" MODUL="FSArch" NAME="1hour" NAME_ru="1час" DESCR="Averaging
for hour" DESCR_ru="Усреднение за час" START="1" ADDR="ARCHIVES/VAL/1h/"
V_PER="360" A_PER="60" FSArchTmSize="8640" FSArchNFiles="10" FSArchRound="0.1"
FSArchPackTm="10" FSArchTm="60"/>
  </tbl>
  <tbl id="Archive_val">
```



```
<fld ID="test1" NAME="Test 1" NAME_ru="Тест 1" DESCR="Test 1"
DESCR_ru="Тест 1" START="1" VTYPE="1" BPER="1" BSIZE="200" BHGRD="1" BHRES="0"
SrcMode="0" Source="" ArchS=""/>
</tbl>
</node>
<node id="sub_Protocol">
</node>
<node id="sub_UI">
<node id="mod_QTStarter">
<prm id="StartMod">QTCfg</prm>
</node>
<node id="mod_WebCfg">
<prm id="SessTimeLife">20</prm>
</node>
<node id="mod_VCAEngine">
<!--
<tbl id="LIB">
<fld ID="test2" NAME="Test 2" NAME_ru="Тест 2" DESCR="" DESCR_ru=""
DB_TBL="wlib_test2" ICO="" USER="root" GRP="UI" PERMIT="436"/>
</tbl>
<tbl id="wlib_test2">
<fld ID="test2" ICO="" PARENT="/wlb_originals/wdg_Box" PROC="" PROC_ru=""
PROC_PER="-1" USER="root" GRP="UI" PERMIT="436"/>
</tbl> <tbl id="wlib_test2_io">
<fld IDW="test2" ID="name" IO_VAL="Test 2" IO_VAL_ru="Тест 2"
SELF_FLG="" CFG_TMPL="" CFG_TMPL_ru="" CFG_VAL=""/>
<fld IDW="test2" ID="dscr" IO_VAL="Test module 2" IO_VAL_ru="Тест модуля
2" SELF_FLG="" CFG_TMPL="" CFG_TMPL_ru="" CFG_VAL=""/>
</tbl>
<tbl id="PRJ">
<fld ID="test2" NAME="Test 2" NAME_ru="Тест 2" DESCR="" DESCR_ru=""
DB_TBL="prj_test2" ICO="" USER="root" GRP="UI" PER </tbl> <tbl id="prj_test2">
<fld OWNER="/test2" ID="pg1" ICO="" PARENT="/wlb_originals/wdg_Box"
PROC="" PROC_ru="" PROC_PER="-1" USER="root" GRP="UI" PERMIT="436" FLGS="1"/>
<fld OWNER="/test2/pg1" ID="pg2" ICO="" PARENT="/wlb_originals/wdg_Box"
PROC="" PROC_ru="" PROC_PER="-1" USER="root" GRP="UI" PERMIT="436" FLGS="0"/>
</tbl>
<tbl id="prj_test2_incl">
<fld ID W="/prj_test2/pg_pg1" ID="wdg1" PARENT="/wlb_originals/wdg_Box"/>
</tbl>
</node>
</node>
```

```
<node id="sub_Special">
  <node id="mod_SystemTests">
    <prm id="PARAM" on="0" per="5" name="LogicLev.experiment.F3"/>
    <prm id="XML" on="0" per="10" file="/etc/oscada.xml"/> <prm id="MESS"
on="0" per="10" categ="" arhtor="DBArch.test3"/>
    <prm id="SOAttDet" on="0" per="20" name="../../../lib/SCADA/daq_LogicLev.so"
full="1"/>
    <prm id="Val" on="0" per="1" name="LogicLev.experiment.F3.var"
arch_len="5" arch_per="1000000"/>
    <prm id="Val" on="0" per="1" name="System.AutoDA.CPULoad.load"
arch_len="10" arch_per="1000000"/>
    <prm id="BD" on="0" per="10" type="MySQL"
bd="server.diya.org;roman;123456;oscadaTest" table="test" size="1000"/>
    <prm id="BD" on="0" per="10" type="DBF" bd="./DATA/DBF"
table="test.dbf"size="1000"/>
    <prm id="BD" on="0" per="10" type="SQLite" bd="./DATA/test.db"
table="test" size="1000"/>
    <prm id="BD" on="0" per="10" type="FireBird"
bd="server.diya.org:/var/tmp/test.fdb;roman;123456" table="test" size="1000"/>
    <prm id="TrOut" on="0" per="1" addr="TCP:127.0.0.1:10001" type="Sockets"
req="time"/>
    <prm id="TrOut" on="0" per="1" addr="UDP:127.0.0.1:10001" type="Sockets"
req="time"/>
    <prm id="TrOut" on="0" per="1" addr="UNIX:./oscada" type="Sockets"
req="time"/>
    <prm id="TrOut" on="0" per="1" addr="UDP:127.0.0.1:daytime"
type="Sockets" req="time"/>
    <prm id="Func" on="0" per="10"/> <prm id="SysContrLang"
on="0"per="10"path="/Archive/FSArch/mess_StatErrors/%2fprm%2fst"/>
    <prm id="ValBuf" on="0" per="5"/> <prm id="Archive" on="0" per="30"
arch="test1" period="1000000"/>
    <prm id="Base64Code" on="0" per="10"/>
  </node>
</node>
</station>
</SCADA>
```

ПРИЛОЖЕНИЕ 2

Результат вызова команды: # scada --help

```
*****
***** SCADA v3.19-b1 (Linux-4.2.0-23-generic). *****
*****
```

```
=====
===== Основные опции системы =====
=====
=====
===== The general system options =====
=====
```

```
-h, --help      Info message about system options.
--Config=<path> Config-file path.
--Station=<id>  The station identifier.
--StatName=<name> The station name.
--demon        Start into demon mode.
--CoreDumpAllow Set limits for core dump creation allow on crash.
--MessLev=<level> Process messages <level> (0-7).
--log=<direct> Direct messages to:
                <direct> & 1 - syslogd;
                <direct> & 2 - stdout;
                <direct> & 4 - stderr;
                <direct> & 8 - archive.
```

----- The config-file station '/EmptySt/' parameters -----

```
StName  <nm>  Station name.
WorkDB  <Type.Name> Work DB (type and name).
Workdir <path> Work directory.
IcoDir  <path> Icons directory.
ModDir  <path> Modules directory.
MessLev <level> Messages <level> (0-7).
LogTarget <direction> Direct messages to:
                <direct> & 1 - syslogd;
                <direct> & 2 - stdout;
                <direct> & 4 - stderr;
                <direct> & 8 - archive.
Lang    <lang> Work-internal language, like "en_US.UTF-8".
```

Lang2CodeBase <lang> Base language for variable texts translation, two symbols code.

SaveAtExit <true> Save system at exit.

SavePeriod <sec> Save system period.

===== Подсистема "Управление модулями" =====

--ModPath=<путь> Путь к модулям (/var/os/modules/).

----- Параметры секции '/WorkStation/sub_ModSched/' в конфигурационном файле -----

ModPath <путь> Путь к разделяемым библиотекам (модулям).

ModAllow <список> Список разделяемых библиотек допустимых для автоматической загрузки, подключения и запуска (bd_DBF.so;daq_JavaLikeCalc.so).

Использовать значение '*' для разрешения всех модулей.

ModDeny <список> Список разделяемых библиотек запрещённых для автоматической загрузки, подключения и запуска (bd_DBF.so;daq_JavaLikeCalc.so).

ChkPer <сек> Период поиска новых разделяемых библиотек (модулей).

===== Опции подсистемы "БД" =====

----- Параметры станции '/WorkStation/sub_BD/' в конфигурационном файле -----

SYSStPref <1> Использовать идентификатор станции в общей (SYS) таблице.

===== Опции подсистемы "Безопасности" =====

===== Опции подсистемы "Транспорты" =====

===== Опции подсистемы "Транспортные протоколы" =====

===== Опции модуля <Protocol:HTTP> =====

-----Параметры модульной секции '/WorkStation/sub_Protocol/mod_HTTP/' в конфигурационном файле -----

AuthTime <мин> Время жизни аутентификации, минут (по умолчанию 10).

===== Опции подсистемы "Сбор данных" =====

----- Параметры секции '/WorkStation/sub_DAQ/' в конфигурационном файле -----

RdStLevel <уров> Уровень текущей станции в схеме резервирования.

RdTaskPer <c> Периодичность вызова задачи обслуживания резервирования.

RdRestConnTm <c> Интервал времени восстановления соединения с "мёртвой" резервной станцией.

RdRestDtTm <час> Глубина восстановления данных архива из резервной станции, при включении.

RdStList <список> Список резервных станций, разделённых символом ';' (st1;st2).

=====
----- Parameters of the module section '/WorkStation/sub_DAQ/mod_IEC104/' in config-file -----

=====
----- Parameters of the module section '/WorkStation/sub_DAQ/mod_Imit/' in config-file -----

=====
----- Parameters of the module section '/WorkStation/sub_DAQ/mod_SD/' in config-file -----

=====
Опции подсистемы "Проверка данных" =====

=====
----- Parameters of section '/WorkStation/sub_Archive/' in config-file -----

MessBufSize <items> Messages buffer size.
MessPeriod <sec> Message archiving period.
ValPeriod <msec> Values archiving period.
ValPriority <level> Values task priority level.
MaxReqMess <items> Maximum request messages.
MaxReqVals <items> Maximum request values.

=====
Опции модуля <Archive:FSArch> =====

--noArchLimit Отключить лимит на количество файлов.
Используйте для режима просмотра архивов, не для работы.
--copyErrValFiles Копирование исходных ошибочных файлов архива значений до восстановления.
Используется для отладки ошибок архивов значений и корректности восстановления.

=====
Опции подсистемы "Специальные" =====

=====
Опции модуля <Special:SystemTests> =====

----- Параметры модульной секции '/WorkStation/sub_Special/mod_SystemTests/' в конфигурационном файле -----

Общие опции всех тестов:

id идентификатор теста;
on флаг включения теста;
per период повторения (сек).

*** Опции тестов ***

- 1) Param Тест DAQ параметров. Вычитывает атрибуты и конфигурационные поля параметра.
1:name Адрес DAQ параметра
- 2) XML Тест разбора файла XML. Разбирает и отображает структуру указанного файла.

1:file XML файл

3) Mess Тест архива сообщений. Периодически вычитывает новые сообщения из архива, для указанного архиватора.

1:arhtor Архиватор

2:categ Шаблон категории сообщения

3:depth Глубина сообщения (с)

4) SOAttach Тест подключения/отключения модулей.

1:name Путь к модулю

2:mode Режим (1-подключение;-1-отключение;0-изменение)

3:full Полное подключение(при старте)

5) Val Тест значений атрибута параметра.

Выполняет периодический опрос последнего значения указанного атрибута, а также опрос архива на указанную глубину.

1:name Путь к атрибуту параметра

2:arch_len Глубина запроса к архиву значений (с)

3:arch_per Период запроса к архиву значений (мс)

6) DB Полный тест БД. Выполняет:

- создание/открытие БД;

- создание/открытие таблицы;

- создание множества записей (строк) предопределённой структуры;

- модификация множества записей;

- получение и проверка значений множества записей;

- модификация структуры записи и таблицы;

- удаление записей;

- закрытие/удаление таблицы;

- закрытие/удаление БД.

1:type Тип БД

2:addr Адрес БД

3:table Таблица БД

4:size Количество записей

7) TrOut Тест выходных и/или входных транспортов.

Выполняет тестирование исходящего транспорта путём отправления запроса к указанному входящему транспорту.

1:addr Адрес

2:type Модуль транспорта

3:req Текст запроса

8) SysContrLang Тест языка управления системой.

Производит запрос элементов языка посредством полного пути.

Полный путь к элементу языка имеет вид `</Archive/%2fd%2fm_per>`.

Полный путь состоит из двух вложенных путей.

Первый `</d_Archive/>` это путь к узлу дерева контроля.

Второй `</bd/m_per>` это путь к конкретному элементу узла.

1:path Путь к элементу языка

9) ValBuf Тесты буфера значений. Содержит 13 тестов всех аспектов буфера значений (подсистема "Архивы").

10) Archive Тесты размещения в архиве значений.

Содержит 7(8) тестов архиватора значений на проверку корректности функционирования последовательного механизма упаковки.

1:arch Архив значений

2:period Период значений (мс)

11) Base64Code Тесты кодирования Mime Base64 алгоритмом.

===== Опции подсистемы "Пользовательские интерфейсы" =====

===== Опции модуля `<UI:QTCfg>` =====

----- Параметры модульной секции `'/WorkStation/sub_UI/mod_QTCfg/'` в конфигурационном файле ---

StartPath `<path>` Стартовый путь конфигулятора.

StartUser `<user>` Стартовый, беспарольный, пользователь.

===== Опции модуля `<UI:QTStarter>` =====

----- Отладочные параметры Qt, командной строки --

--sync Переключение в синхронный режим X11 для отладки.

--widgetcount Печать отладочных сообщений при выходе, о количестве виджетов оставшихся неудалёнными и максимальном их количестве.

----- Параметры Qt, командной строки -----

--qws Делает данное приложение сервером с Qt для встраиваемого Linux.

--style=`<имя>` Установить GUI стиль в `<имя>` (windows, platinum, plastique, ...).

--stylesheet=`<путь>` Установить таблицу стилей из файла по `<пути>`.

--session=`<имя>` Восстановить из предыдущего сеанса `<имя>`.

--reverse Установить направление размещения в Qt::RightToLeft.

--graphicssystem=`<имя>` Установить механизм рендеринга для экранных виджетов и QPixmaps (raster, opengl).

--display=`<имя>` Установить X экран (типично в \$DISPLAY).

--geometry=`<геом>` Установить клиентскую геометрию первого отображаемого окна.

----- Параметры модульной секции '/WorkStation/sub_UI/mod_QTStarter/' в конфигурационном файле

StartMod <модули> Список запускаемых модулей (разделитель - ';');

===== Опции модуля <UI:Vision> =====

----- Параметры модульной секции '/WorkStation/sub_UI/mod_Vision/' в конфигурационном файле ---

--

StartUser <польз> Стартовый, беспарольный, пользователь.

UserPass <пароль> Пароль пользователя для нелокального запуска.

RunPrjs <список> Перечень запускаемых при старте проектов.

RunPrjsSt {0;1} Отображать статус для запускаемых проектов (по умолчанию = 1).

ExitLstRunPrjCls {0;1} Выход при закрытии последнего исполняющегося проекта (по умолчанию = 1).

CachePgLife <часы> Время жизни страниц в кеше.

VCAsation <id> Станция с движком СВУ ('!' - локальная).

PlayCom <команда> Команда проигрывания аудио-файлов сигнализации.

===== Опции модуля <UI:WebVision> =====

----- Параметры модульной секции '/WorkStation/sub_UI/mod_WebVision/' в конфигурационном файле -----

SessTimeLife <время> Время жизни сессии, минуты (по умолчанию 10).

SCADA system is correctly exited by cause 10.

ПЕРЕЧЕНЬ ПРИНЯТЫХ СОКРАЩЕНИЙ

- БД - база данных
- ОЗУ - оперативное запоминающее устройство
- ОС - операционная система
- ПО - программное обеспечение
- API - application programming interface (программный интерфейс приложения)
- HTML - язык гипертекстовой разметки (Hypertext Mark-up Language)
- SCADA - диспетчерское управление и сбор данных (Supervisory Control And Data Acquisition)

