

## АННОТАЦИЯ

Настоящая часть руководства оператора содержит описание создания проекта в ПП «СКАДА А-СОФТ» (далее по тексту СКАДА) и организации работы с источниками данных в данной системе.

## СОДЕРЖАНИЕ

<b>1 РАЗРАБОТКА ПРОЕКТА .....</b>	<b>5</b>
1.1 ОБЩАЯ ЧАСТЬ.....	5
1.2 СОЗДАНИЕ БД ДЛЯ БАЗОВОГО ПРОЕКТА.....	6
1.2.1 Создание новой БД в PostgreSQL.....	6
1.2.2 Добавление новой БД в СКАДА.....	8
1.2.3 Сохранение БД.....	11
1.3 СОЗДАНИЕ ИСТОЧНИКОВ ДАННЫХ .....	12
1.4 СОЗДАНИЕ БИБЛИОТЕКИ БАЗОВЫХ ВИЗУАЛЬНЫХ ЭЛЕМЕНТОВ.....	15
1.5 СОЗДАНИЕ ЭЛЕМЕНТА В БИБЛИОТЕКЕ «ПРОЕКТ (ИНТЕРФЕЙС)».....	18
1.5.1 Создание виджета на основе базовых шаблонов.....	18
1.5.2 Создание элемента видеокadra копированием.....	19
1.5.3 Создание базовых видеокadров.....	21
1.6 СОЗДАНИЕ ЭЛЕМЕНТОВ ВИДЕОКАДРОВ.....	22
1.6.1 Добавление элементов отображения аналогового сигнала.....	22
1.6.2 Создание комплексного элемента.....	23
1.7 СОЗДАНИЕ ПРОЕКТА .....	35
1.7.1 Настройка окна визуального представления.....	36
1.7.2 Добавление элемента в проект.....	37
1.7.3 Подключение мнемосхем и источников данных к проекту.....	38
1.7.4 Исполнение проекта .....	40
<b>2 НАСТРОЙКА РАЗЛИЧНЫХ ТИПОВ ИСТОЧНИКОВ ДАННЫХ.....</b>	<b>43</b>
2.1 Модуль источника данных MODBUS.....	43
2.1.1 Конфигурирование сбора данных по протоколу ModBUS.....	43
2.1.2 Конфигурирование обработки данных, полученных по протоколу ModBUS.....	53
2.2 Модуль шлюз источников данных DAQGATE.....	59
2.2.1 Назначение модуля.....	59
2.2.2 Конфигурирование передачи данных.....	60
2.3 Модуль источника данных TANGO.....	65
2.4 Модуль источника данных PCI.....	69
2.5 Модуль источника данных SNMP.....	69
2.6 Модуль источника данных IEC 61850.....	73
2.7 Модуль источника данных IEC 60870-5-104.....	77
2.8 Модуль Клиент диагностики ПТК.....	80
2.8.1 Составление конфигурационного файла dasc.xml.....	82
2.8.2 Составление конфигурационного файла dsc.xml.....	82
2.9 Модуль источника данных OPC UA.....	85
2.10 Модуль источника данных HART.....	89
<b>3 НАСТРОЙКА РЕЗЕРВИРОВАНИЯ.....</b>	<b>96</b>
3.1 РЕЗЕРВИРОВАНИЕ МОДУЛЯ «БАЗЫ ДАННЫХ».....	96
3.2 РЕЗЕРВИРОВАНИЕ МОДУЛЯ «СБОР ДАННЫХ».....	98
<b>4 РЕКОМЕНДАЦИИ ПО РАБОТЕ С ПРОЕКТОМ.....</b>	<b>100</b>
4.1 РЕДАКТИРОВАНИЕ ГРАФИЧЕСКОЙ ЧАСТИ ПРОЕКТА АСУ ТП.....	100

4.2 НАСТРОЙКА ОТОБРАЖЕНИЯ ПРОЕКТА НА УДАЛЕННОМ АРМ.....	100
4.3 НАСТРОЙКА ОТОБРАЖЕНИЯ ПРОЕКТА НА НЕСКОЛЬКИХ МОНИТОРАХ .....	101
4.4 СОХРАНЕНИЕ ПРОЕКТА АСУ ТП И ЕГО ЧАСТЕЙ .....	102
4.4.1 <i>Рекомендации</i> .....	102
4.4.2 <i>Что не рекомендуется делать при сохранении проекта</i> .....	102
4.5 ПЕРЕНОС КОНФИГУРАЦИИ СКАДА ИЗ ОДНОГО ПРОЕКТА В ДРУГОЙ.....	103
<b>ПРИЛОЖЕНИЕ 1.....</b>	<b>105</b>
<b>ПРИЛОЖЕНИЕ 2.....</b>	<b>109</b>
<b>ПРИЛОЖЕНИЕ 3.....</b>	<b>111</b>
<b>ПРИЛОЖЕНИЕ 4.....</b>	<b>112</b>
<b>ПЕРЕЧЕНЬ ПРИНЯТЫХ СОКРАЩЕНИЙ.....</b>	<b>113</b>

## 1 Разработка проекта

### 1.1 Общая часть

Непосредственная конфигурация и свойства конечного интерфейса визуализации содержатся в проекте интерфейса визуализации СВУ. Может быть создано множество проектов интерфейсов визуализации.

Каждый проект включает видеокadres из библиотек видеокadres/виджетов. Видеокادر предоставляет инструмент для привязки динамики к описанным в нём свойствам. Все свойства видеокадра могут быть связаны с динамикой или определены константами, а могут выступать в роли шаблона для формирования производных страниц.

Для создания нового проекта в графике в СКАДА необходимо:

- создать базу данных для хранения проекта в PostgreSQL;
- в системном конфигураторе СКАДА создать БД с обязательной привязкой к БД, сделанной в предыдущем пункте;
- создать в системном конфигураторе источники данных и переменные;
- в редакторе пользовательского интерфейса в подсистеме «Виджеты» создать:
  - библиотеки визуальных элементов проекта путем копирования либо проектирования новых;
  - библиотеки видеокadres;
  - создать и заполнить видеокadres элементами;
- создать проект в подсистеме «Проекты» редактора пользовательского интерфейса с подключением созданных ранее библиотек и элементов.

Источниками данных являются контроллеры с параметрами, созданные в подсистеме «Сбор данных» с использованием шаблонов или спроектированные самостоятельно в зависимости от требований поставленной задачи.

Далее рассмотрим выполнение каждого шага более подробно.

## 1.2 Создание БД для базового проекта

Проект имеет модульную структуру и создается на основе различных библиотек: библиотеки видеокadres, библиотеки динамических и статических элементов видеокadra, библиотеки сервисных элементов (кнопки вызова видеокadres, индикаторы времени и т.д.).

В ПП «СКАДА А-СОФТ» все элементы хранятся в БД (предпочтительнее использовать PostgreSQL). Допустимо произвольное разделение элементов по БД: возможно хранение всех элементов в одной БД, каждого элемента в отдельной БД, хранение каждой группы элементов в своей БД. Для больших проектов удобнее разделение библиотек по группам элементов. Рассмотрим далее создание проекта, данные которого хранятся в одной БД PostgreSQL.

### 1.2.1 Создание новой БД в PostgreSQL

Создать новую БД можно двумя способами: используя приложение pgAdmin или выполнив команду в терминале Fly.

#### 1.2.1.1 Создание БД через приложение pgAdmin

Для создания новой БД в PostgreSQL необходимо зайти на вкладку «Разработка» из

меню «Пуск» и запустить приложение pgAdmin:



В окне «Браузер объектов» выбрать postgres (localhost:5432) → Базы данных и нажать на правую кнопку мыши (рисунок 1). Из всплывающего меню выбрать пункт «Новая база данных...» и в появившемся окне «Новая база данных» ввести имя «NewDB» (рисунок 2).

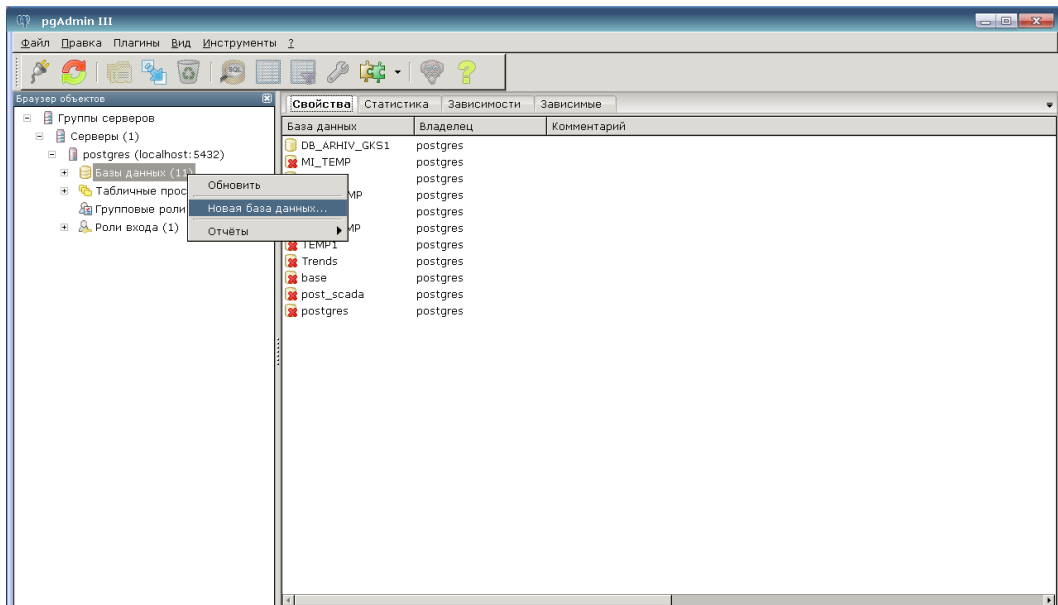


Рисунок 1

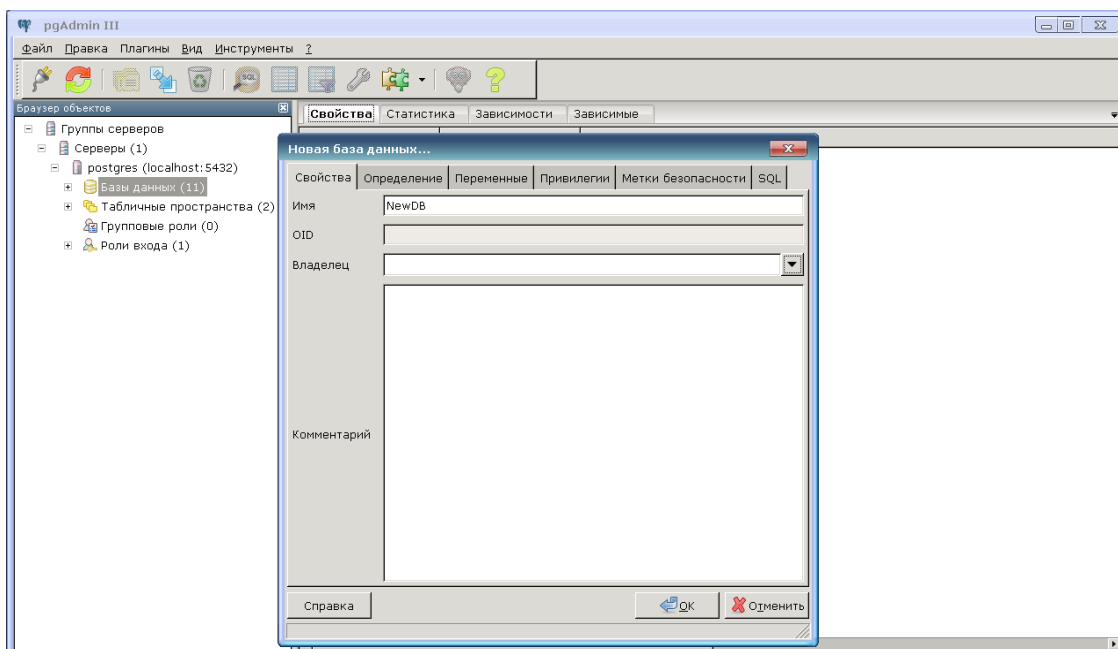


Рисунок 2

Перейдя на вкладку «Определение» можно задать характеристики БД: кодировку, шаблон БД и другие (рисунок 3). Подтвердить создание новой БД нажатием на кнопку ОК и выйти из приложения pgAdmin.

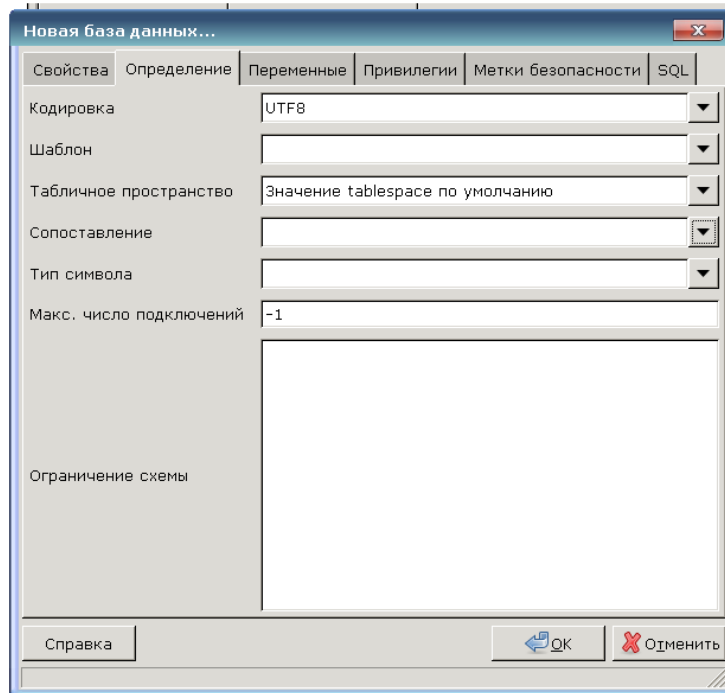


Рисунок 3

#### 1.2.1.2 Создание новой БД командой в терминале Fly

Для создания новой БД необходимо вызвать окно терминала и запустить оболочку `psql` от имени непривилегированного пользователя ОС - `postgres` (администратор БД), выполнив команду:

```
psql -U postgres
```

При этом система попросит ввести пароль для пользователя `postgres`. В результате, в окне терминала высветится приглашение `postgres=#`.

Далее следует создать новую базу данных командой:

```
create database <имя БД>;
```

Выйти из программной оболочки `psql` командой `\q`.

#### 1.2.2 Добавление новой БД в СКАДА.

Для создания новой БД в левой части окна configurатора ПП «СКАДА А-СОФТ» (рисунок 4) раскрыть вкладку «Базы данных». В появившемся списке выбрать вкладку «БД PostgreSQL» и нажатием правой клавиши «мыши» на ней вызвать контекстное меню, в котором выбрать пункт «Добавить».

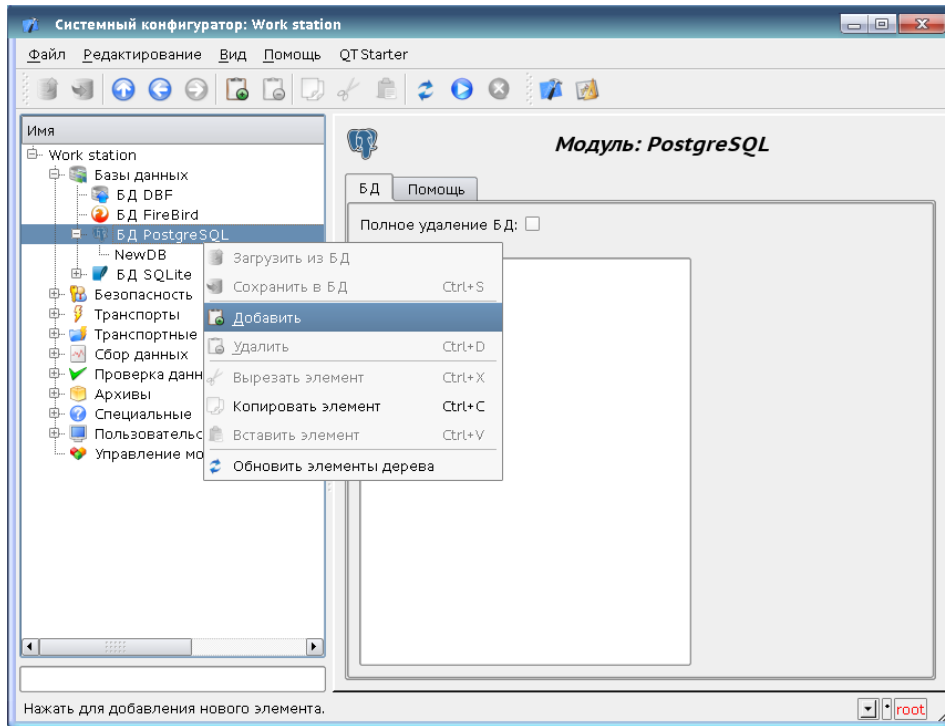


Рисунок 4

В появившемся окне (рисунок 5) необходимо задать имя и «ID» (идентификатор) БД, в которой будет храниться проект. Имя используется для отображения пользователю. Имена любых объектов СКАДА ограничены размером в 50 символов и могут содержать любые символы (цифры, буквы любого алфавита, знаки и т.д.). Если имя объекта остается пустым, то вместо него для отображения будет использоваться идентификатор объекта. Идентификатор (ID) используется для обращений к БД внутри ПП «СКАДА А-СОФТ». Идентификатор может содержать только буквы латинского алфавита, цифры, символы «-» и «\_», кроме того начинать идентификатор рекомендуется с буквы. Длина «ID» ограничена 20 символами. После задания имени и идентификатора БД закрыть окно нажатием кнопки «ОК» (рисунок 5).

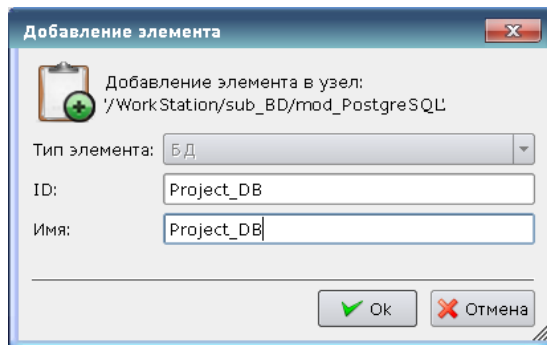


Рисунок 5



В появившемся окне (рисунок 6) в поле ввода «Адрес» необходимо задать полный путь к файлу БД, в котором будет храниться проект, и установить отметку около пункта «Включать» (указывает на состояние БД при загрузке).

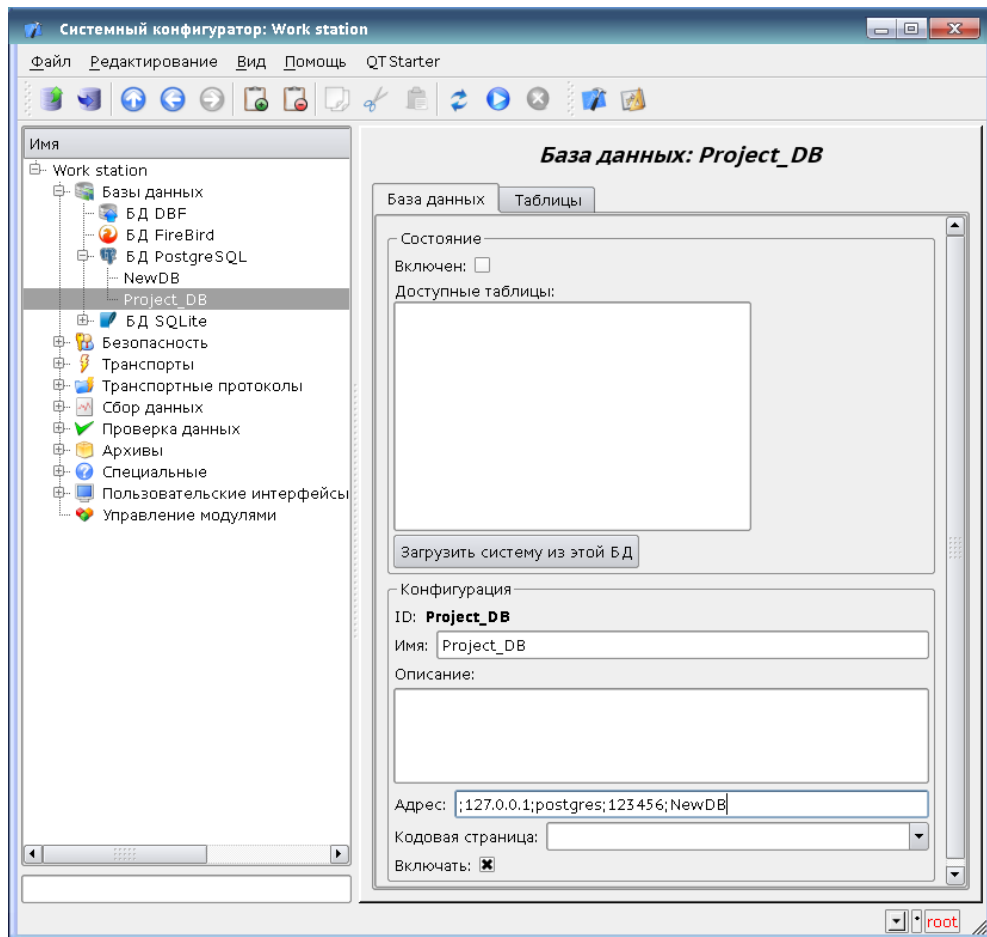


Рисунок 6

Включить БД, установив отметку возле пункта «Включен» (рисунок 7).

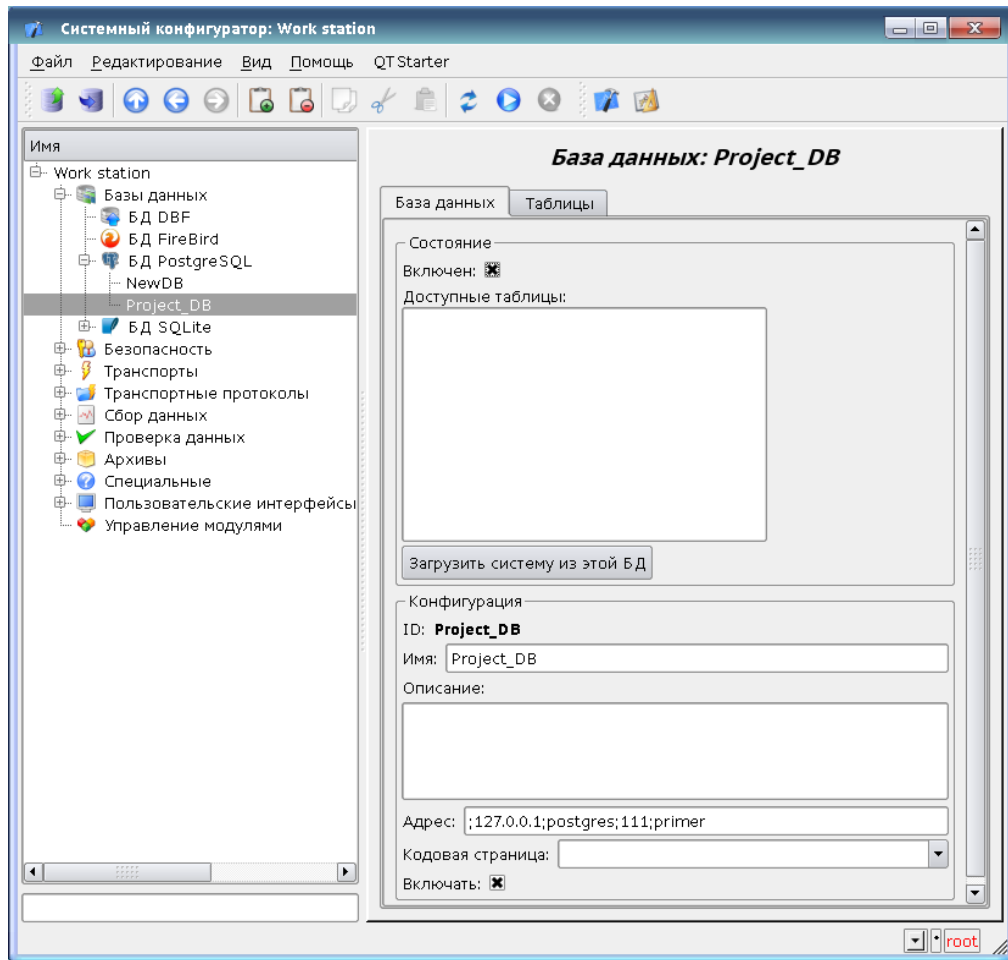



Рисунок 7

### 1.2.3 Сохранение БД.

Сохранить изменения можно, нажав на пиктограмму  окна configurатора, либо выбрав правой кнопкой мыши «Сохранить в БД», либо нажав на клавиатуре сочетание клавиш «Ctrl+S» (сохранить) и подтвердить сохранение.

### 1.3 Создание источников данных

Для примера создадим контроллер «primer» для отображения параметров на Видеокадре 1.

Для этого в Системном конфигураторе СКАДА выберем «Сбор данных» → «Логический уровень» и, выбрав в контекстном меню строку «Добавить», создадим контроллер с ID и именем «primer» (рисунок 8).

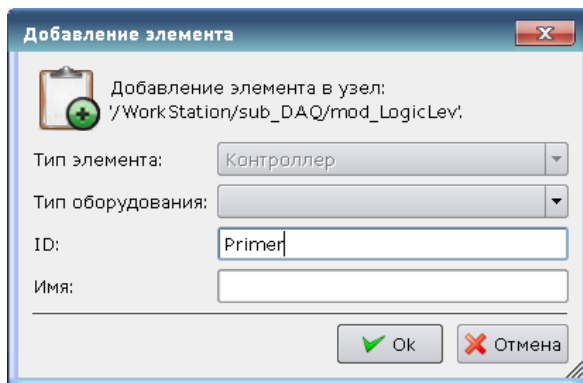


Рисунок 8

Далее необходимо сконфигурировать контроллер. Для этого на вкладке «Контроллер» установить БД контроллера, выбрав из всплывающего списка «БД контроллера». Элемент списка должен иметь имя вида «PostgreSQL.<Имя БД>». Где <Имя БД> соответствует «ID» ранее созданной БД - «PROJECT\_DB» (см. 1.2.2). После чего, поставить галочки «включен» и «запущен» (рисунок 9).

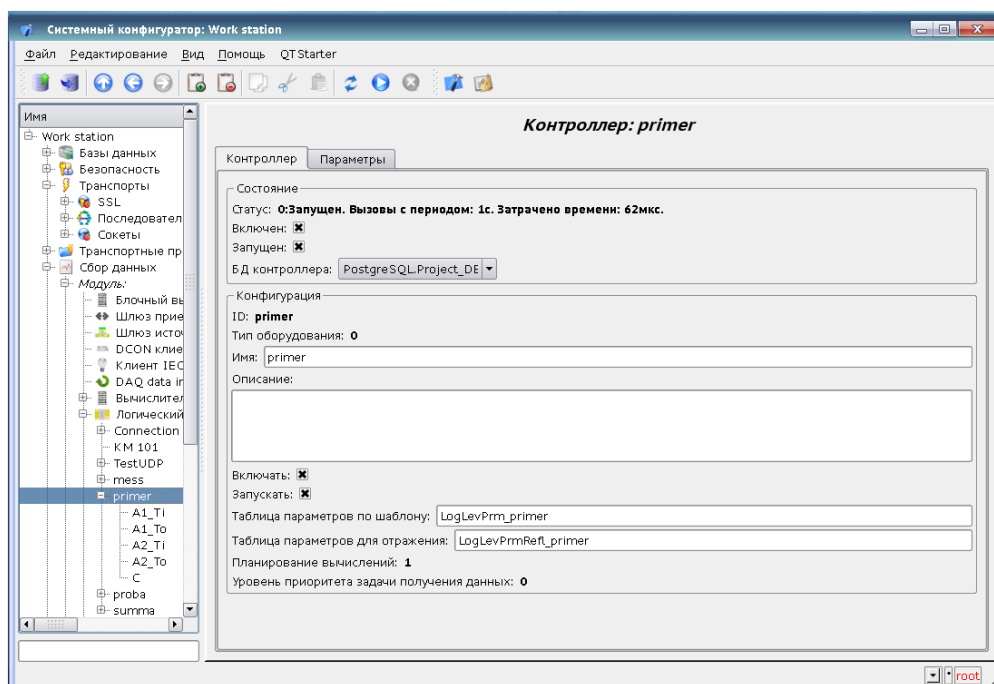


Рисунок 9

Открыв вкладку «Параметры» контроллера «primer», добавить 5 параметров: A1\_Ti, A1\_To, A1\_Ti, A2\_To, C (для отображения температуры на входе, на выходе и суммы). Для всех параметров необходимо указать состояние «включен» и задать шаблоны: для параметров температур - «base.rand1», для параметра сумма- «base.summa».

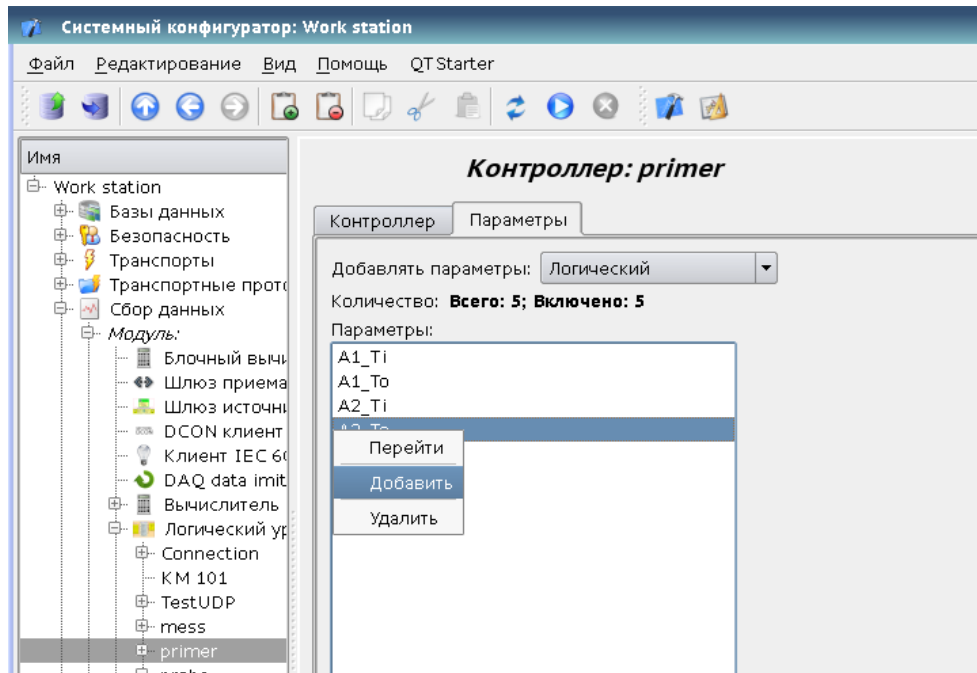


Рисунок 10

На рисунке 11 показана установка логических связей для атрибута C – сумма. Конфигурация параметра «C» изменяется путем последовательного указания в соответствующих полях значений слагаемых – атрибутов, определенных ранее (выбор из выпадающего списка).

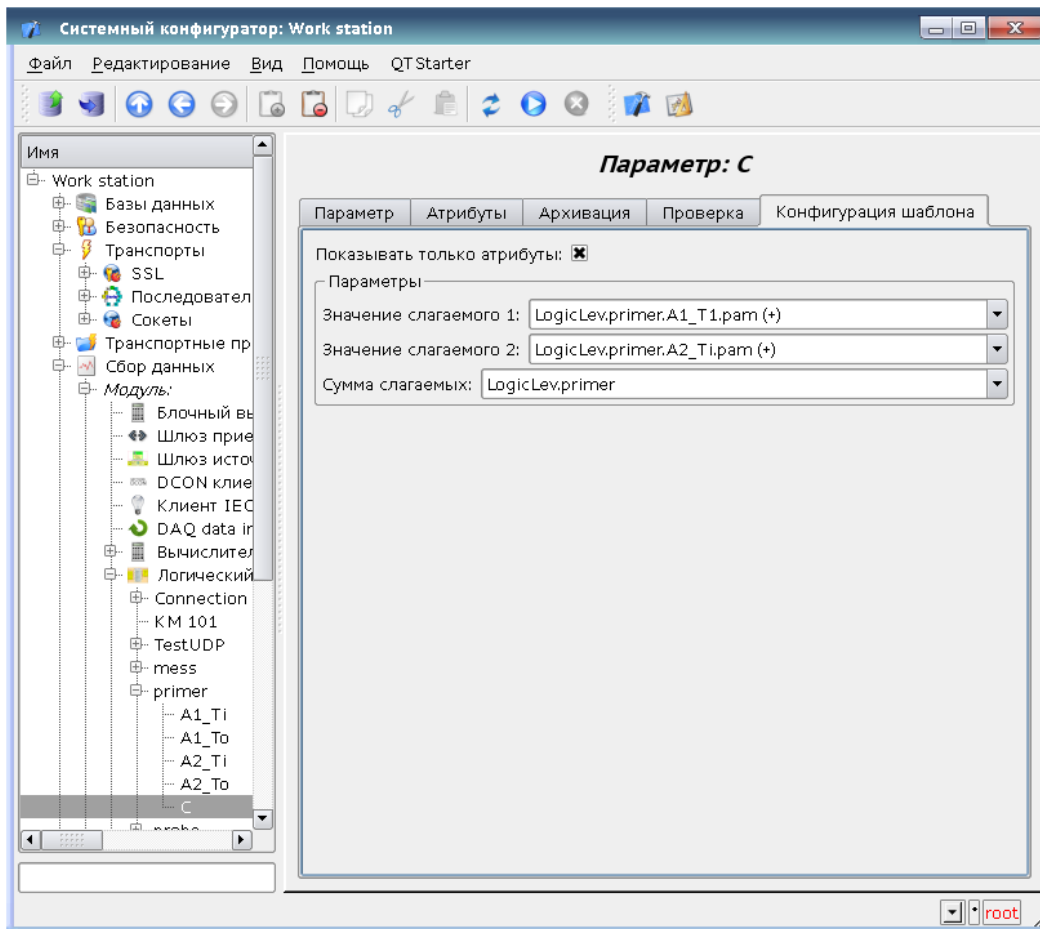


Рисунок 11

На этом создание контроллера Primer считается окончанным, поэтому необходимо сохранить сделанные изменения в соответствии с 1.2.3.

## 1.4 Создание библиотеки базовых визуальных элементов

Базовыми визуальными элементами для проекта являются такие элементы как: главное окно, кнопки перехода на видеокadres, индикатор времени, базовые элементы отображения значений параметров контроллеров и т.д. Новые видеокadres, предназначенные впоследствии для помещения в проект, принято создавать в библиотеке виджетов.

Для открытия окна интерфейса "Vision" нажать крайнюю правую иконку на панели инструментов конфигуратора «Рабочий пользовательский интерфейс (Qt)» (рисунок 12).

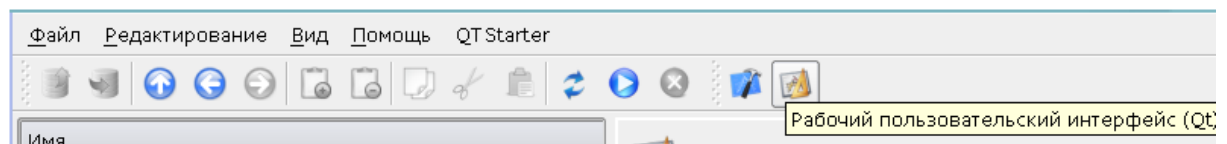


Рисунок 12

В открывшемся окне интерфейса визуализации выбрать в главном меню пункт «Виджет» и в открывшемся контекстном меню выбрать пункт «Новая библиотека» (рисунок 13).

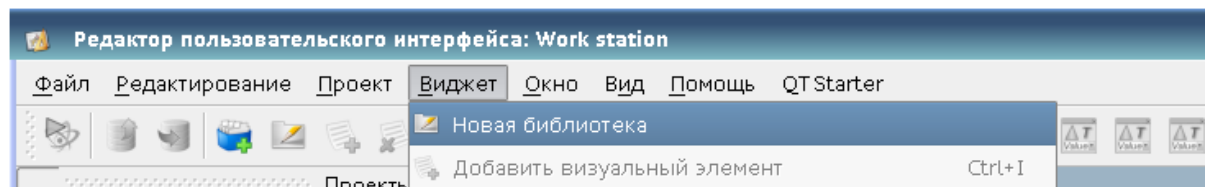


Рисунок 13

В появившемся окне (рисунок 14) задать идентификатор (ID) и имя библиотеки. Нажать кнопку «Принять». Идентификатор используется для обращений к библиотеке внутри системы. Имя библиотеки используется для отображения в списке библиотек виджетов.

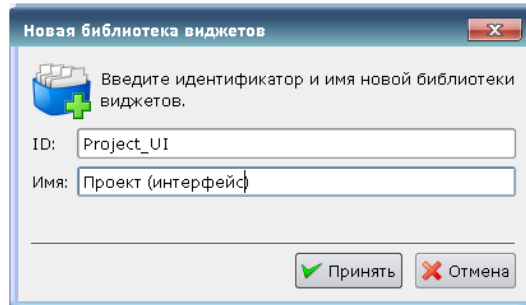


Рисунок 14

Выбрать в левой части окна интерфейса визуализации вертикальную вкладку «Виджет» (рисунок 15) и в появившемся списке выбрать созданный элемент «Проект (интерфейс)».

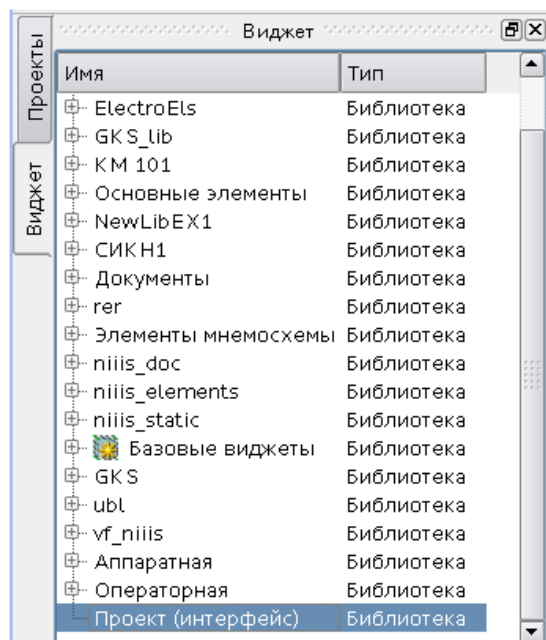


Рисунок 15

Выбрать «Свойства визуального элемента» из всплывающего списка или нажать сочетание клавиш «Ctrl+P» и в появившемся окне (рисунок 16) раскрыть список «БД контейнера». Выбрать базу данных (БД) для хранения библиотеки визуальных элементов. Элемент списка должен иметь имя вида

PostgreSQL.<Имя БД>.<Имя библиотеки>

где <Имя БД> соответствует «ID» ранее созданной БД «PROJECT\_DB»;

<Имя библиотеки> соответствует «ID» настраиваемой библиотеки «PROJECT\_UI».

После выбора элемента нажать кнопку «Закреть».

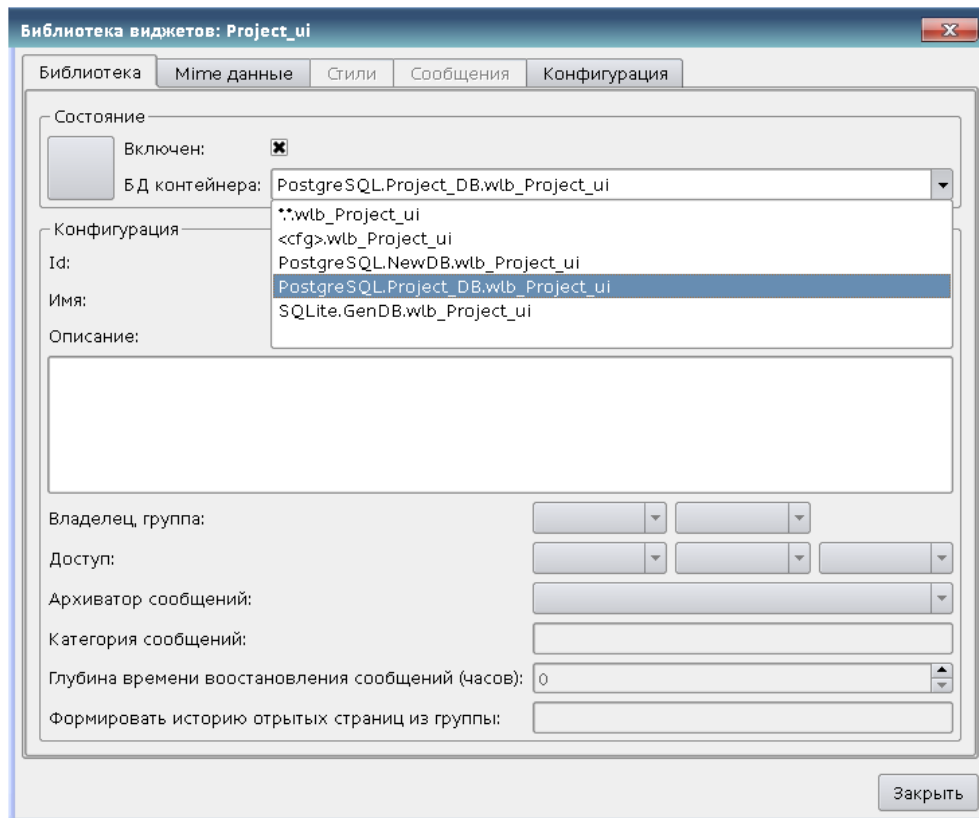



Рисунок 16

Выбрать библиотеку в списке виджетов и сохранить, нажав на пиктограмму  окна конфигуратора, либо выбрав правой кнопкой мыши «Сохранить в БД», либо нажав на клавиатуре сочетание клавиш «Ctrl+S». В появившемся диалоговом окне подтверждения сохранения (рисунок 17) нажать кнопку «Принять».

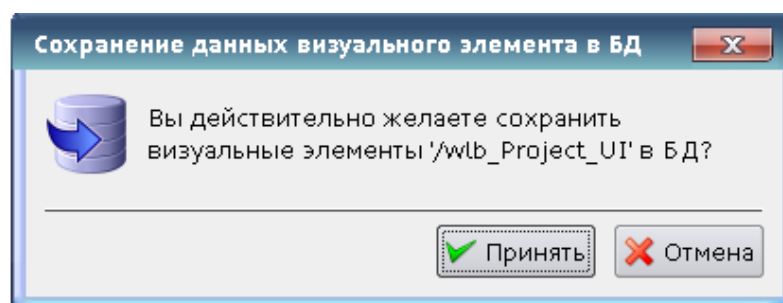


Рисунок 17

Для разделения по функциональным группам аналогичным образом необходимо создать библиотеку элементов с ID «Project\_EL» и именем «Проект (элементы)» и библиотеку видеок кадров с ID «Project\_VF» и именем «Проект (ВК)». В первую библиотеку поместим элементы видеок кадров для использования в проекте (линии, элементы отображения аналоговых значений и т.п.), во втором – создадим видеок кадры (мнемосхемы, предназначенные для отображения технологического процесса).



## 1.5 Создание элемента в библиотеке «Проект (интерфейс)»

ПП «СКАДА А-СОФТ» позволяет создавать новые элементы на основе базовых шаблонов или копированием уже существующих элементов. В библиотеке «Проект (интерфейс)» создадим элемент «Главное окно», в котором в дальнейшем будет осуществляться переключение по видеокадрам и отображаться текущая дата.

### 1.5.1 Создание виджета на основе базовых шаблонов.

Для создания нового видеокadra необходимо, выделив библиотеку «Проект (интерфейс)», выбрать пункт «Библиотека: originals» → «Группа элементов» в контекстном меню созданной библиотеки. В диалоге ввода имени указать идентификатор «Main» и имя «Главное окно», а затем подтвердить изменения. В основе любого видеокadra и страницы должен лежать элемент «Группа элементов».

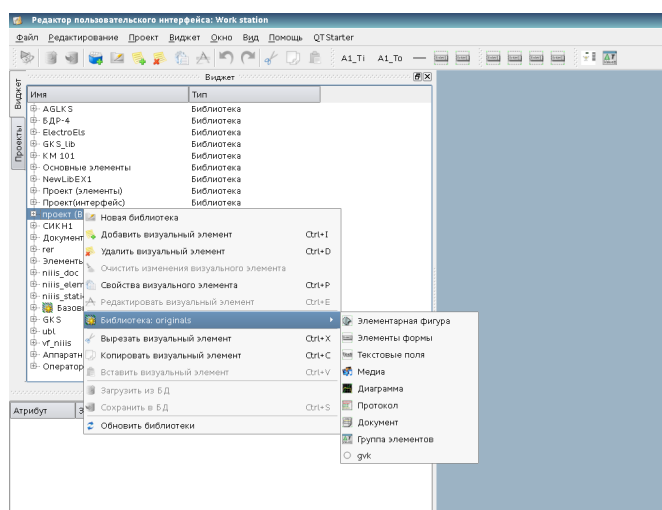


Рисунок 18

Сразу после создания элемента нового видеокadra можно установить его базовые свойства. Свойства или атрибуты любого визуального элемента можно указать в панели инструментов «Атрибуты», предварительно выбрав нужный визуальный элемент или нажав «горячие клавиши» на клавиатуре «Ctrl+P». Выберем созданный Элемент «Главное окно» и установим для него следующие свойства:

- Геометрия: ширина - 500;
- Геометрия: высота - 500;
- Граница: ширина - 1;
- Граница: цвет - «black».

В результате получим пустой видеокادر (рисунок 19), готовый для добавления элементов на него. Для редактирования или просмотра вида видеокадра необходимо в контекстном меню видеокадра выбрать пункт «Редактировать визуальный элемент» или нажать сочетание клавиш «Ctrl+E».

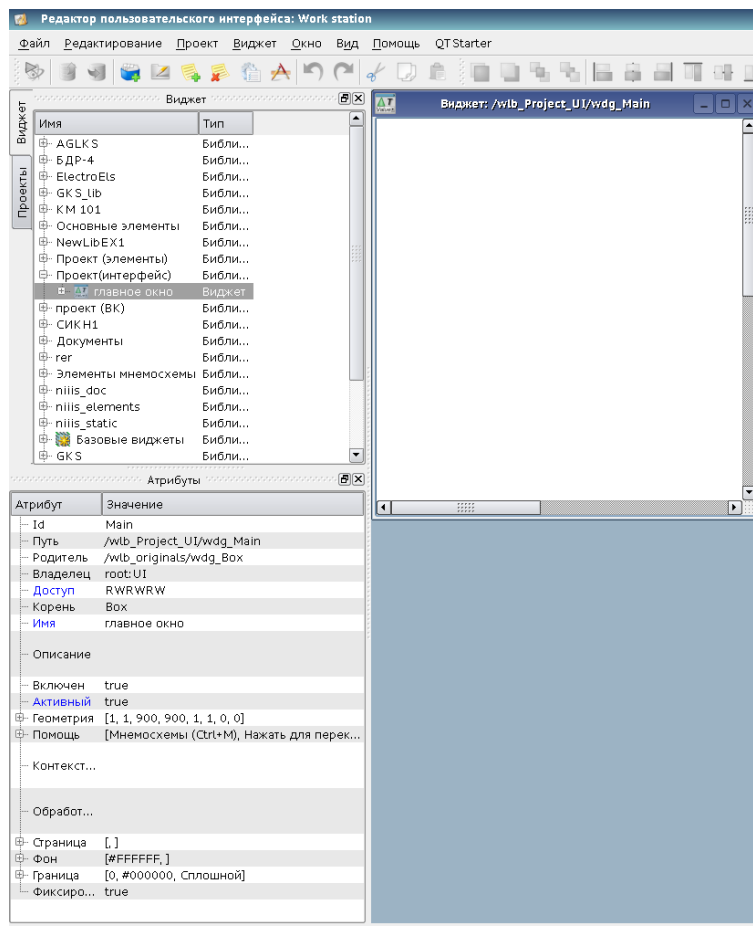


Рисунок 19

Сохраним сделанные изменения в соответствии с 1.2.3.

### 1.5.2 Создание элемента видеокадра копированием.

Для создания элемента копированием необходимо скопировать библиотеку из уже созданных библиотек. Например, можно выбрать в библиотеке виджетов «Основные элементы» → «Корневая страница» → элемент «go\_mn» (рисунок 20) и выбрать строку «Копировать визуальный элемент» контекстного меню (при нажатии правой кнопкой мыши) или нажать сочетание клавиш «Ctrl+C». Затем выбрать библиотеку «Проект (интерфейс)» и произвести вставку элемента - нажав сочетание клавиш «Ctrl+V».

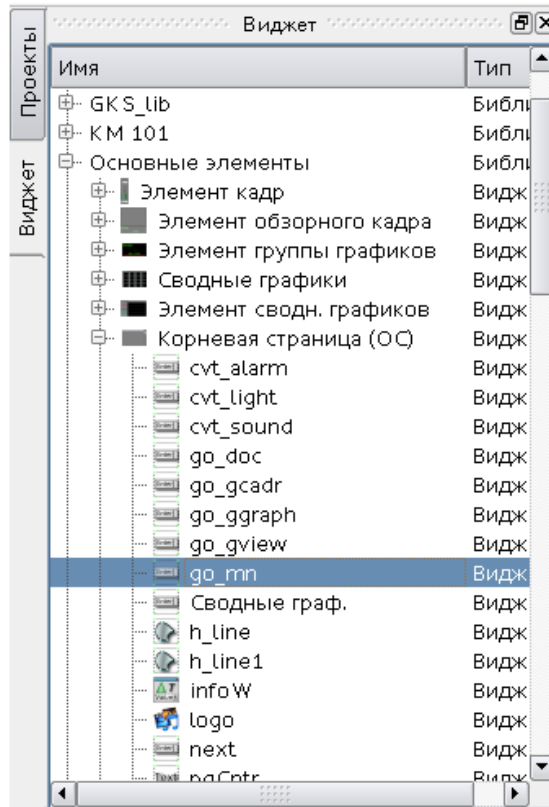


Рисунок 20

В появившемся окне (рисунок 21) изменить «ID» на «MAIN» (обязательно, для всех других элементов при копировании менять «ID» не нужно). Имя, если не будет указано, будет взято с копируемого элемента. Нажать кнопку «Принять».

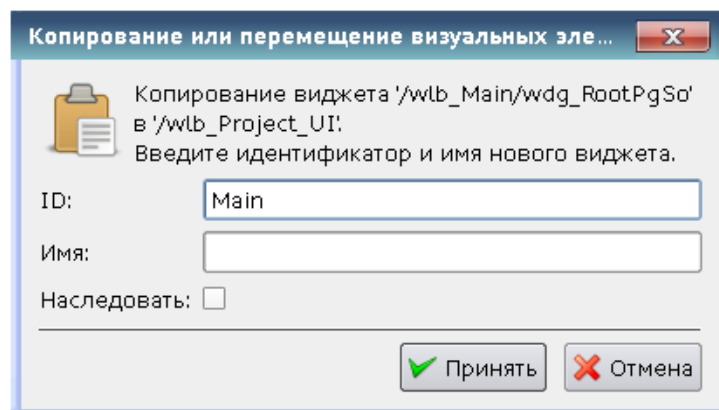


Рисунок 21

### 1.5.3 Создание базовых видеокадров.

В соответствии с 1.4 в библиотеке «Проект (ВК)» создадим пустой видеокадр с ID «VK» и именем «Пустой». Затем для него изменим свойства во вкладке «Атрибуты»: «Геометрия» (ширина, высота) и «Фон». Таким образом, можно настроить базовый размер видеокадра и цвет фона. Обратите внимание, что измененные поля у унаследованных виджетов подсвечиваются синим цветом для удобства отслеживания изменений и последующей их «очистки» (отката) нажатием правой кнопкой «мыши» по измененному атрибуту.

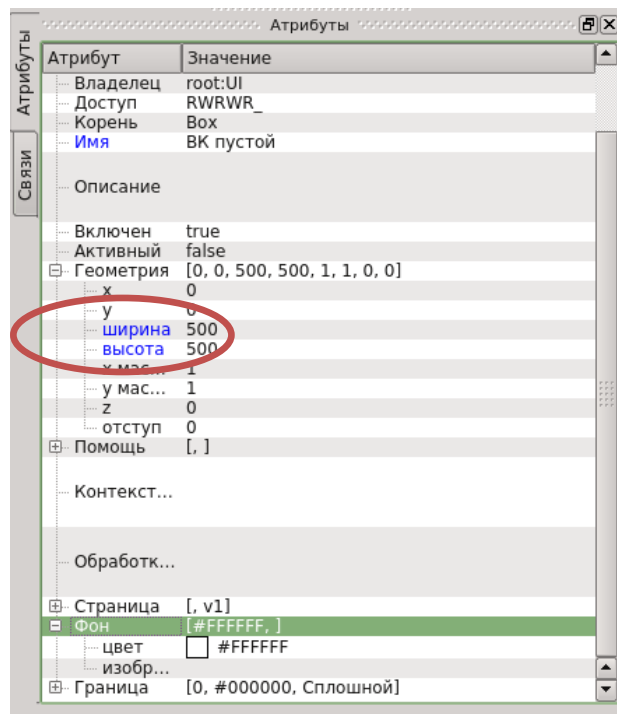


Рисунок 22

Скопировав элемент «VK пустой» способом, аналогичным описанному в 1.5.2, в библиотеку «Проект (ВК)», указав при этом в качестве ID - «VK1», а имени - «Видеокадр1» получаем видеокадры с одинаковыми характеристиками. Создадим еще один видеокадр с ID «VK2» и именем - «Видеокадр 2». Обязательно сохранить изменения (см. 1.2.3).

## 1.6 Создание элементов видеокадров.

Элементами видеокадра (мнемосхемы) являются:

- статические элементы (надписи, рисунки, линии и прочие фигуры), т.е. элементы, пиктограмма которых остается неизменной;
- динамические элементы представляют собой пиктограммы, отображающие состояние механизмов, процессов, аналоговых и дискретных параметров, т.е. их состояние меняется с течением времени;
- сервисные элементы, такие как поля перехода.

Описание всех элементов СКАДА приведено в части 2 настоящего руководства оператора.

Далее рассмотрим заполнение видеокадров различными элементами.

### 1.6.1 Добавление элементов отображения аналогового сигнала.

Добавим на «Видеокадр1» библиотеки «Проект (ВК)» элементы отображения значения аналогового параметра для четырёх сигналов. Для помещения на видеокадр элемента отображения аналогового сигнала необходимо выбрать «Видеокадр 1», а затем в меню окна выбрать пункт «Виджет» → «Библиотека: Main» → «Отобр аналог»; после чего появится курсор с образом этого элемента, который нужно подвести в желаемую область видеокадра и нажать левую кнопку мыши. В момент добавления появится диалог с запросом имени нового элемента. Добавим подобным образом пять элементов, которые назовём: «A1\_Ti», «A1\_To», «A2\_Ti», «A2\_To», «С». Добавленные элементы можно впоследствии расположить как нужно, просто выделяя и перетаскивая «мышью», либо указать координаты элемента (заполнить поля атрибута «Геометрия»). Масштабирование элемента производится с помощью одновременного нажатия левой кнопки «мыши» и клавиши «Ctrl». Выравнивание элементов по горизонтали или вертикали, а также перемещение на более низкий уровень мнемосхемы осуществляется путем выделения элементов и нажатия на одну из кнопок панели инструментов



или из меню «Виджет» → «Вид»).

После выполнения подобных манипуляций у нас должен получиться видеокадр с видом, похожим на представленный на рисунке 23.

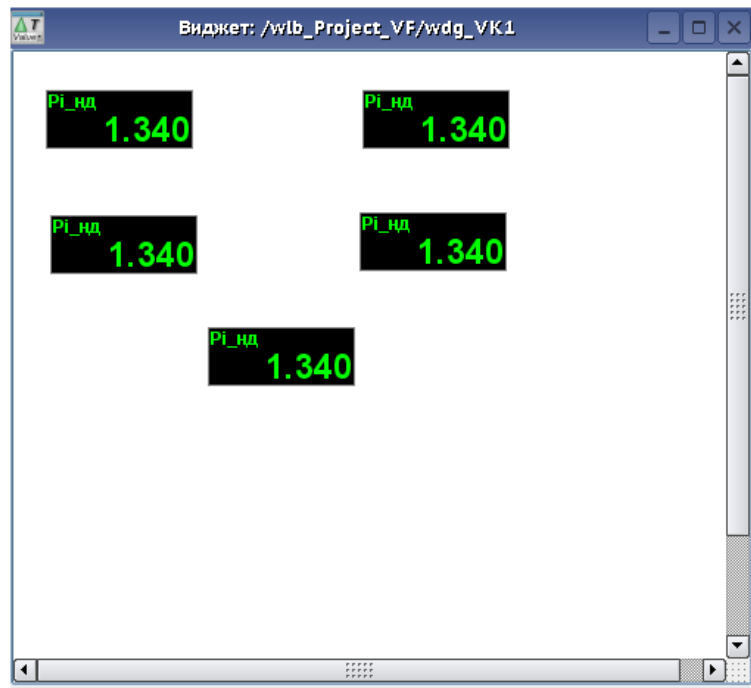


Рисунок 23

На этом процедуру создания мнемосхемы будем считать законченной. Сохраним изменения в соответствии с 1.2.3

#### 1.6.2 *Создание комплексного элемента.*

Создание нового комплексного элемента, включающего в себя комбинацию различных базовых примитивов, может осуществляться в несколько этапов. В качестве примера рассмотрим задачу, создания виджета «Кран с заглушкой», изменяющего цвет в зависимости от своего состояния.

##### 1.6.2.1 *Создание видеокадра на основе примитива «Элементарная фигура».*

Элемент будем создавать в созданном ранее базовом виджете «Видеокадр2». Для этого кликаем правой кнопкой «мыши» по пункту названия библиотеки «Видеокадр 2» и выбираем пункт «Библиотека: originals»→ «Элементарная фигура», как это показано на рисунке 24.

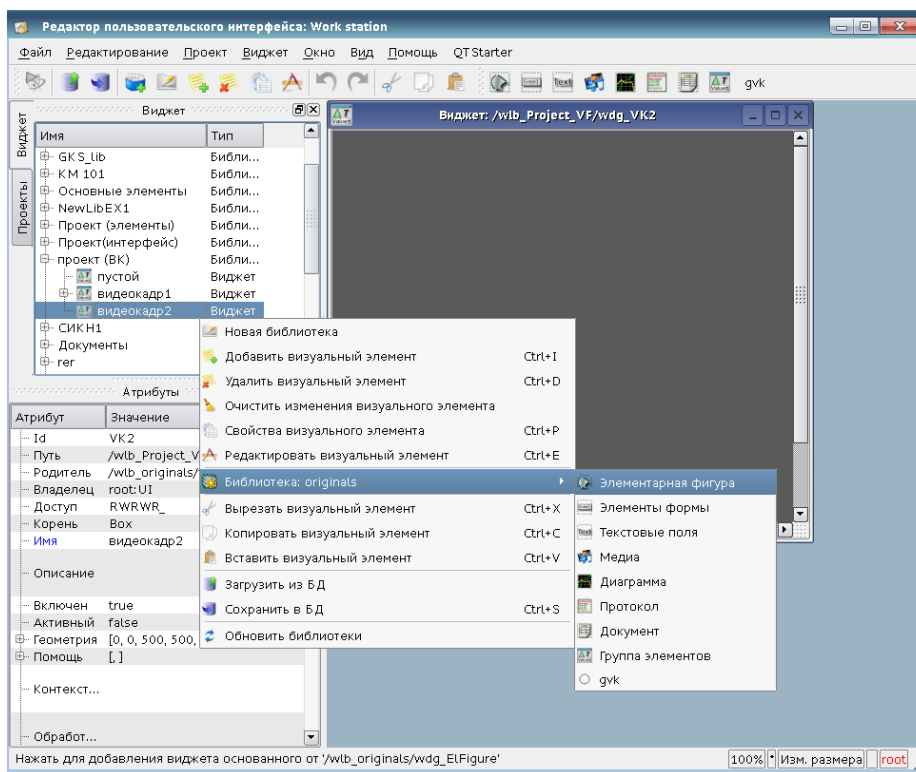


Рисунок 24

Для нового элемента указываем идентификатор «left», имя наследуется автоматически.

После подтверждения ввода появится объект нового виджета с именем «left». Выберем его в списке виджетов библиотеки «Проект (VK)» и откроем для редактирования посредством контекстного меню нового элемента или нажав сочетание клавиш «Ctrl+P».

Теперь нарисуем визуальное представление виджета «left» (левая сторона крана). Эту процедуру можно проделать двумя нижеописанными способами:

- нарисовать желаемое изображение «мышью», используя «Линию», «Дугу», «Кривую Безье» и «Заливку». Соответствующая панель («Панель элементарных фигур») появится после входа в режим редактирования (рисования). Вход в этот режим осуществляется, как показано на рисунке 25, либо двойным нажатием левой кнопки «мыши» на теле виджета;
- вручную заполнить поле «Список элементов», введя перечень необходимых элементов и координат точек.

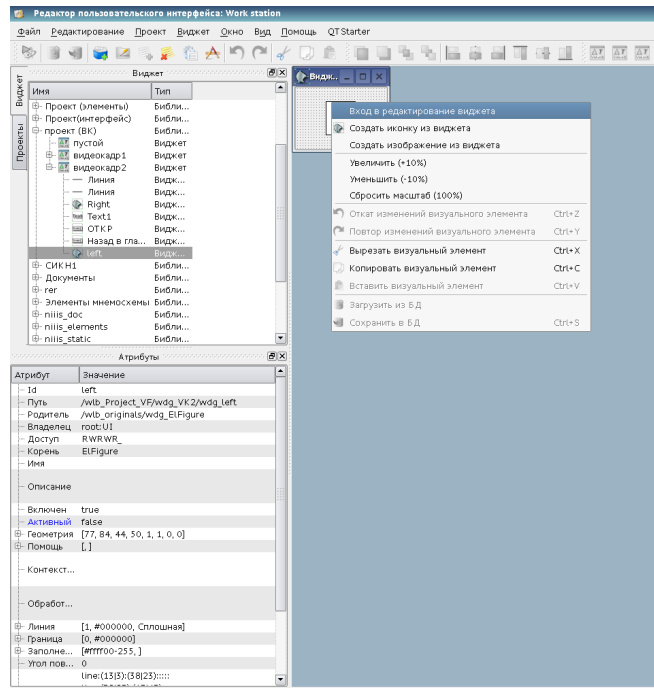


Рисунок 25

В нашем примере воспользуемся вторым способом. Для этого в поле значений атрибута «Список элементов» атрибутов введем нижеприведенный перечень и нажмем «Ctrl»+ «Enter»:

```
line: (20|10) : (45|30)
line: (45|30) : (20|50)
line: (20|50) : (20|10)
fill: (20|10) : (45|30) : (20|50) ::
```

где line – отрисовка линии;

fill – заполнение цветом фигуры.



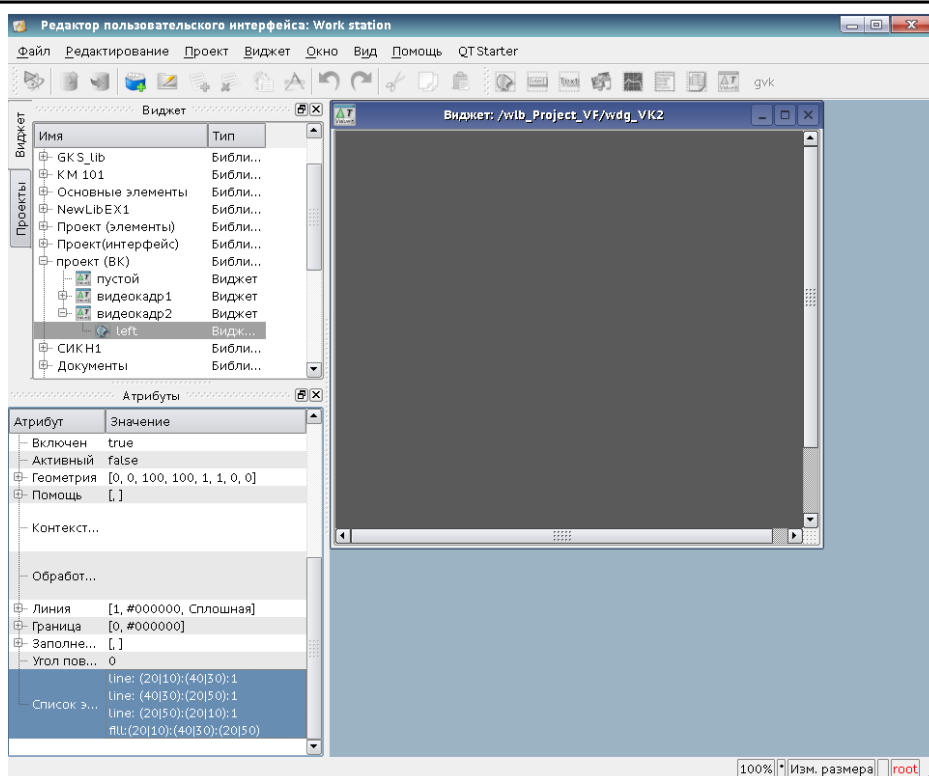


Рисунок 26

После чего установим значение атрибута «Цвет» заполнения "#ffff00". Цвет можно задавать, как с помощью названий цветов, так и выбрав из палитры в формате #RRGGBB (#RRGGBB-AAA, где AAA - прозрачность).

Все точки, в нашем случае, указаны в статическом виде, так как не предусматривается динамизация и смена координат в режиме исполнения, а все остальные параметры оставлены по умолчанию. Вследствие этого наш виджет примет вид, изображенный на рисунке 27.

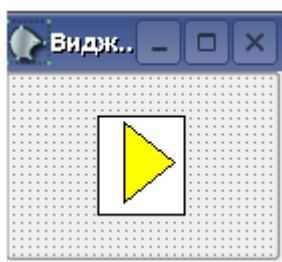


Рисунок 27

Создадим иконку для нашего виджета, которая будет видна в дереве виджетов. Для этого в поле редактирования элемента необходимо кликнуть правой кнопкой мыши и из всплывающего меню выбрать строку «Создать иконку из виджета» (рисунок 28).

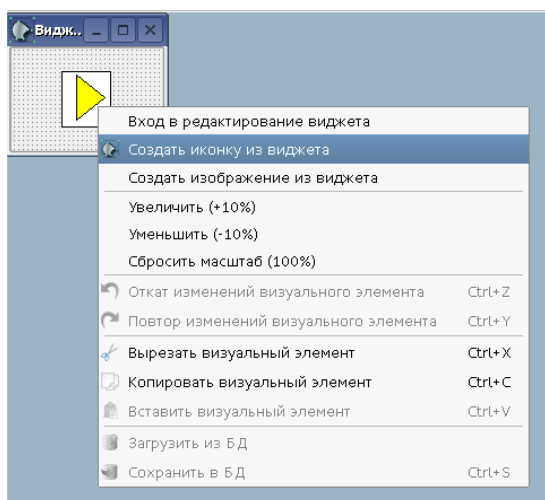


Рисунок 28

Сохраним сделанные изменения.

Аналогичным образом создадим виджет правой стороны крана. Для этого откроем в библиотеке «Проект(ВК)» - «Видеокадр 2» и создадим элемент как описано ранее или путем копирования элемента «left».

Рассмотрим второй вариант. Для этого выберем элемент «left» нашей библиотеки и нажмем сочетание клавиш «Ctrl+C», далее вставим скопированный элемент, нажав сочетание клавиш «Ctrl+V», и в поле ID зададим «right». Далее необходимо исправить значение атрибута «Угол поворота» на 180. В результате получится зеркальное отражение левой части крана (рисунок 29).

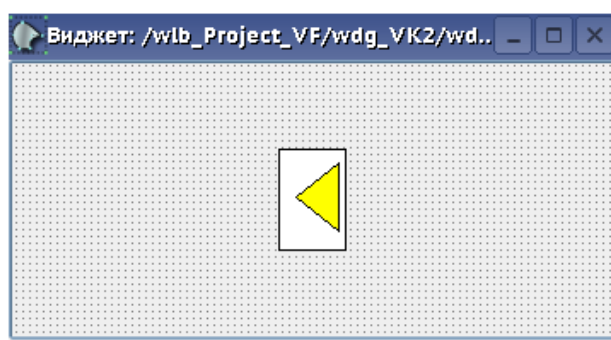


Рисунок 29

Сохраним сделанные изменения.

Далее необходимо совместить два элемента в окне виджета «VK2» (рисунок 30).

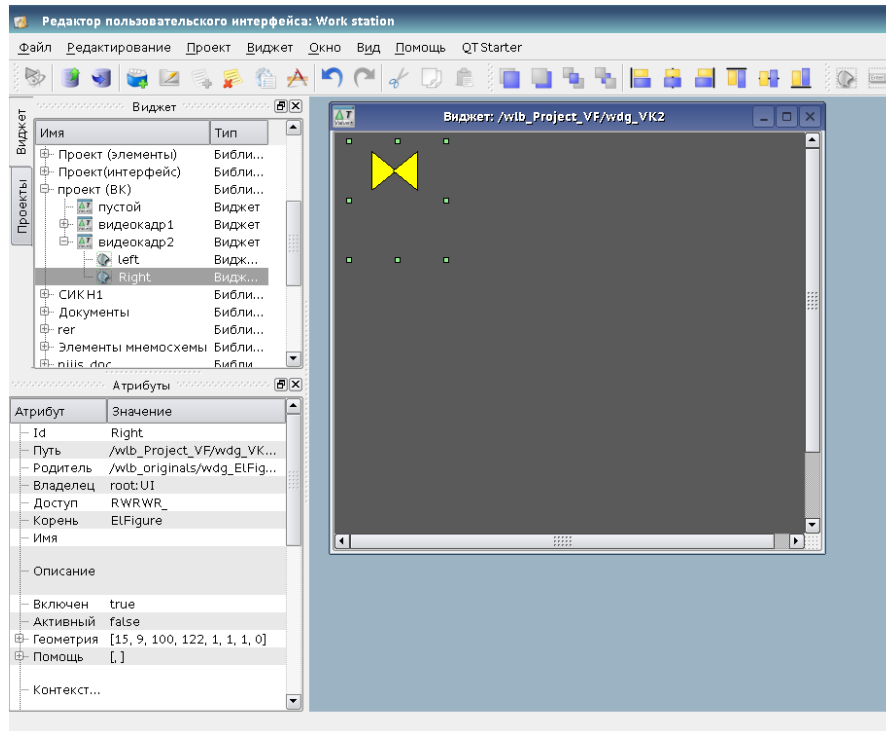


Рисунок 30

Теперь добавим на наш виджет линии. Для этого перетащим мышкой из библиотеки «Проект (элементы)» элемент Линия на наш видеокадр. Имя и ID виджета можно оставить без изменений. Установим линию на свое усмотрение. Сохраним изменения.

Далее можно добавить текстовое поле в наш виджет-контейнер «VK2», основанное на примитиве «Текст», с целью добавления названия крана. Для этого в библиотеке «Проект (VK)» выделим виджет «Видеокадр 2» и нажмем на панели визуальных элементов на иконку примитива «Текст», как это показано на рисунке 31. В результате появится диалог ввода идентификатора и имени вновь создаваемого элемента. Согласимся с предложенным вариантом.

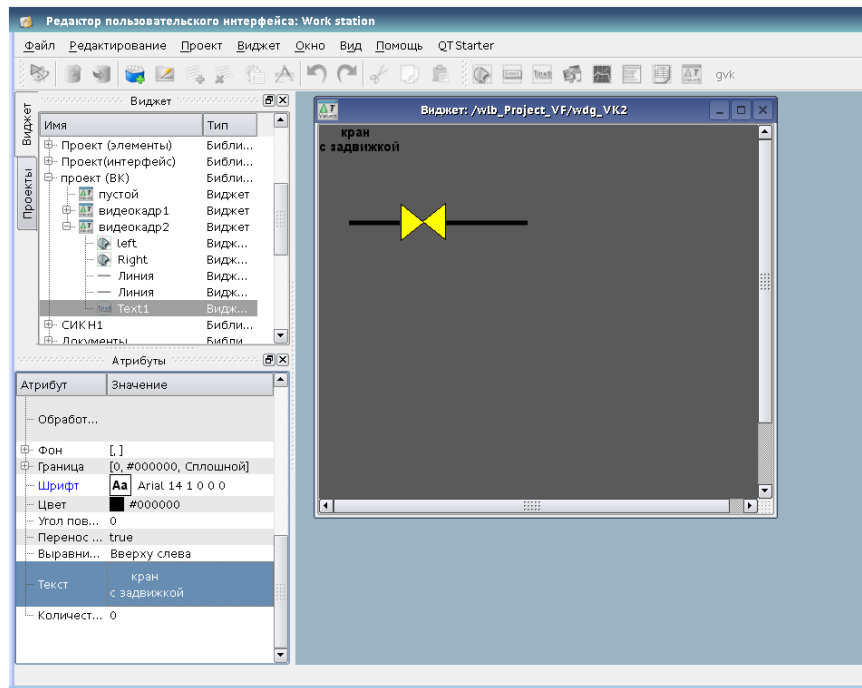


Рисунок 31

Изменим размер и установим усиление шрифта для этого элемента, нажав левой кнопкой мыши на значок ключа в поле «Шрифт» (рисунок 32). В поле значения атрибута «Текст» введем название «Кран с задвижкой».

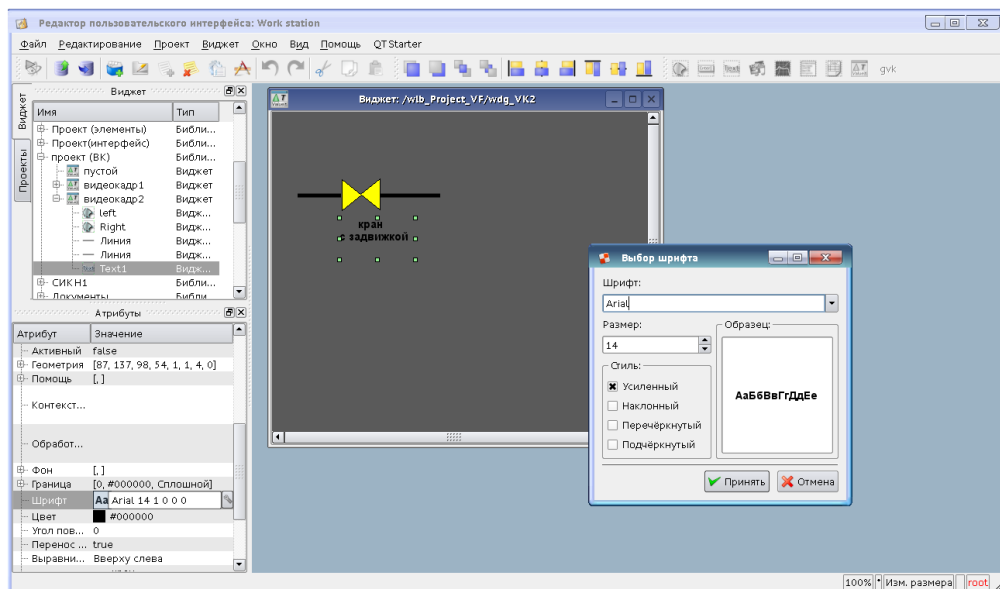


Рисунок 32

Сохраним видеокادر.

### 1.6.2.2 Добавление логики обработки виджета.

Для добавления логики обработки виджета «Видеокадр 2» (VK2) откроем диалог редактирования свойств этого визуального элемента, нажав сочетание клавиш «Ctrl+P», и перейдём на вкладку «Обработка». На этой вкладке мы увидим дерево атрибутов виджета и поле текста программы, для обработки атрибутов. В качестве примера будем изменять цвет заливки крана с желтого на зеленый в зависимости от его состояния: открыт/закрыт.

Для решения нашей задачи нужно добавить два атрибута: ON и OFF (рисунок 33). Для добавления атрибута необходимо развернуть корневой элемент «+», выбрать любой элемент внутри и нажать кнопку «Добавить атрибут». После добавления каждого атрибута задать для него все поля, как показано на рисунке.

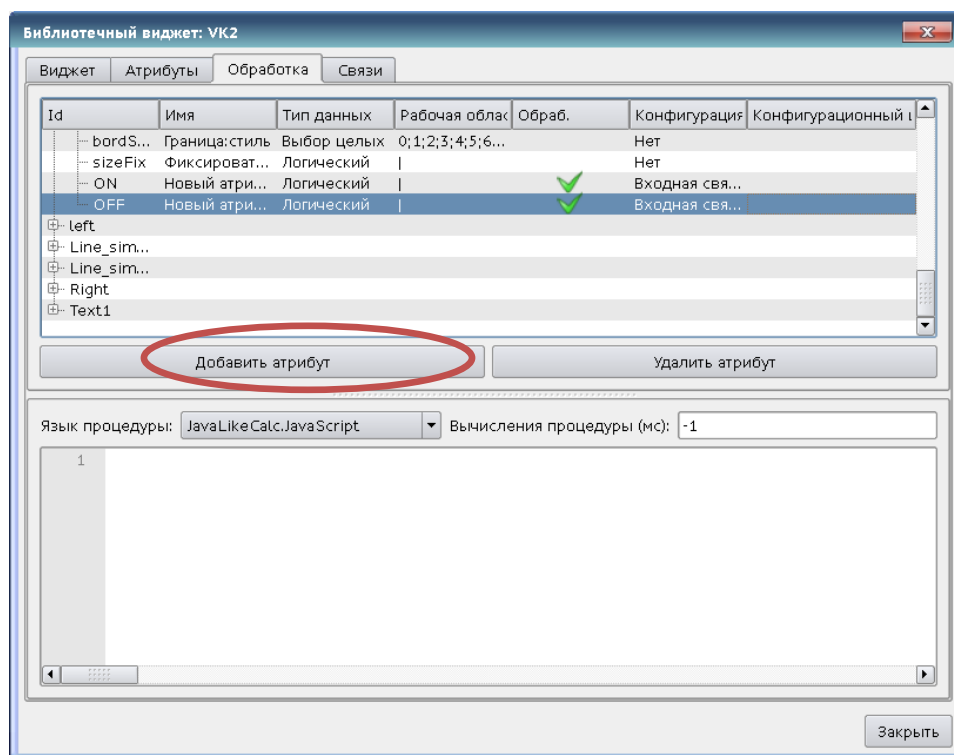


Рисунок 33

Далее для элементов «left» и «right» изменить конфигурацию для поля «FillColor» на «Постоянная» и установить флаг «true» в поле «Обработка» (рисунок 34).

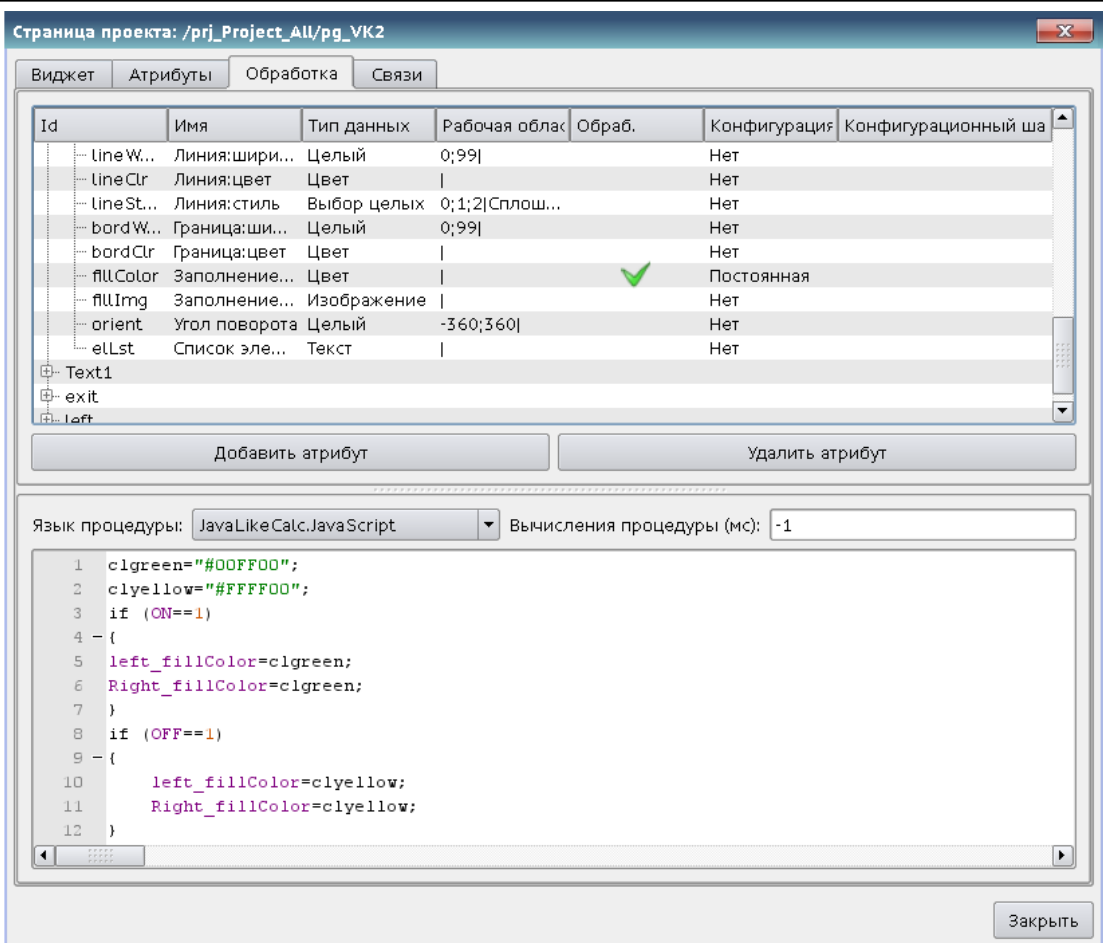


Рисунок 34

В завершение установим язык пользовательского программирования для процедуры в «JavaLikeCalc.JavaScript» и напишем программу обработки этого виджета:

```
nSingSpesial.FLibSys;
clgreen="#00FF00";
clyellow="#FFFF00";
if (ON==1)
{
    left_fillColor=clgreen;
    right_fillColor=clgreen;
}
If (OFF==1)
{
    left_fillColor=clyellow;
    right_fillColor=clyellow;
}
```

Нажмем на кнопку «Принять», после чего программа будет сохранена.

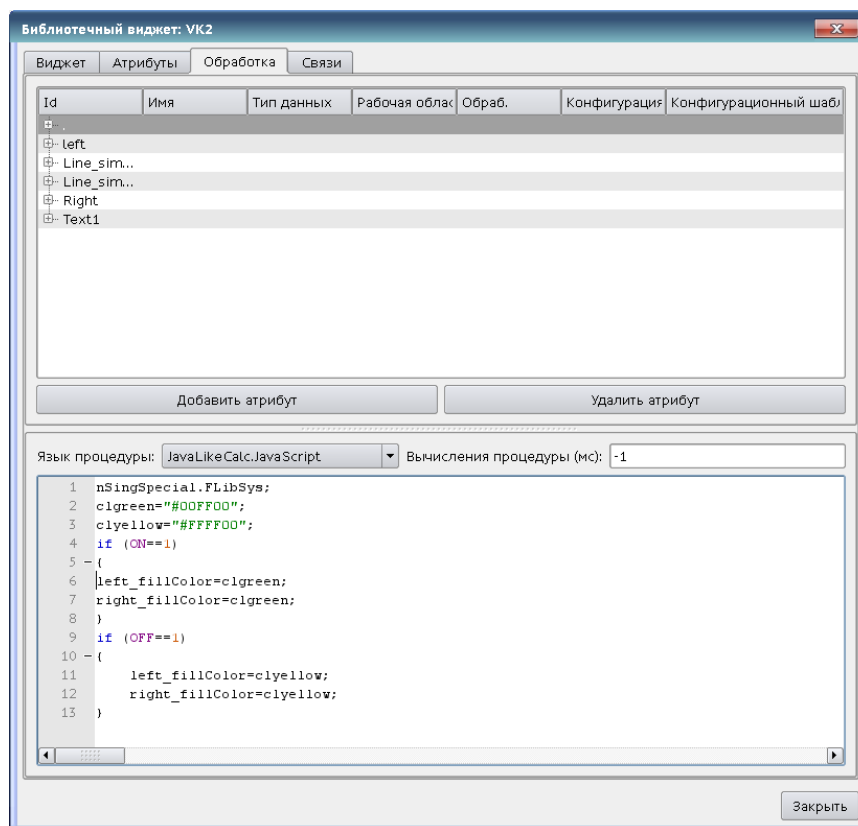


Рисунок 35

### 1.6.2.3 Создание сервисных кнопок

В Главном окне интерфейса пользователя поместим сервисные кнопки для перехода по мнемосхемам, созданным ранее, и поле редактирования текущей даты.

Для редактирования визуального представления «Главного окна» выберем библиотеку «Проект (интерфейс)» → «Главное окно» и нажмем сочетание клавиш «Ctrl+E». В появившемся окне создадим кнопку из примитива библиотеки originals. В качестве ID укажем «VK1» (соответствует ID виджета «Видеокадр 1»), имя укажем «Видеокадр 1». Размер и положение кнопки определим произвольно. Изменим атрибуты кнопки:

тип элемента - кнопка;

активный - true;

шрифт - усиленный, 14;

обработка событий: ws\_BtPress::open://pg\_VK1

Сохраним произведенные изменения.

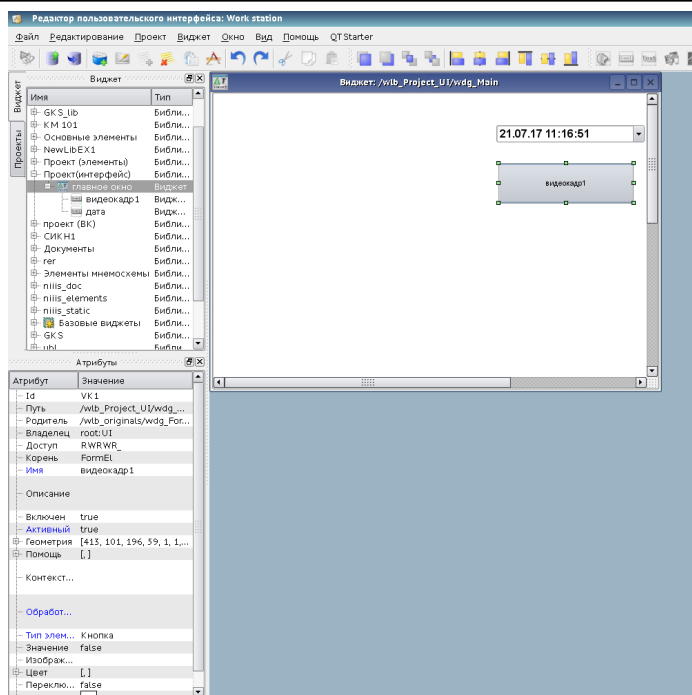


Рисунок 36

Скопируем виджет кнопки «Видеокадр 1» в библиотеку «Главное окно». Зададим ID – «VK2», имя – «Видеокадр 2». Все значения атрибутов у кнопки будут установлены как у копируемой. Необходимо будет только внести изменения в поле Обработка событий: `ws_BtPress::open://pg_vk2`. Сохраним изменения.

В «Главном окне» создадим еще одно поле «Текущая дата». Поле скопировано из примитива «Основные элементы», ID и имя оставлены без изменений.

На этом будем считать работу над виджетом «Главное окно» законченной, поэтому сохраним виджет и закроем окно редактирования.

Теперь создадим сервисную кнопку возврата в главное окно для мнемосхем, созданных ранее. Для этого откроем для редактирования библиотеку «Проект (VK)» → «Видеокадр 1» и создадим кнопку с ID «Exit» и названием «Возврат в главное окно». Установим для нее атрибуты:

тип элемента – кнопка;

активный – true;

шрифт – усиленный, 13.

Сохраним изменения.



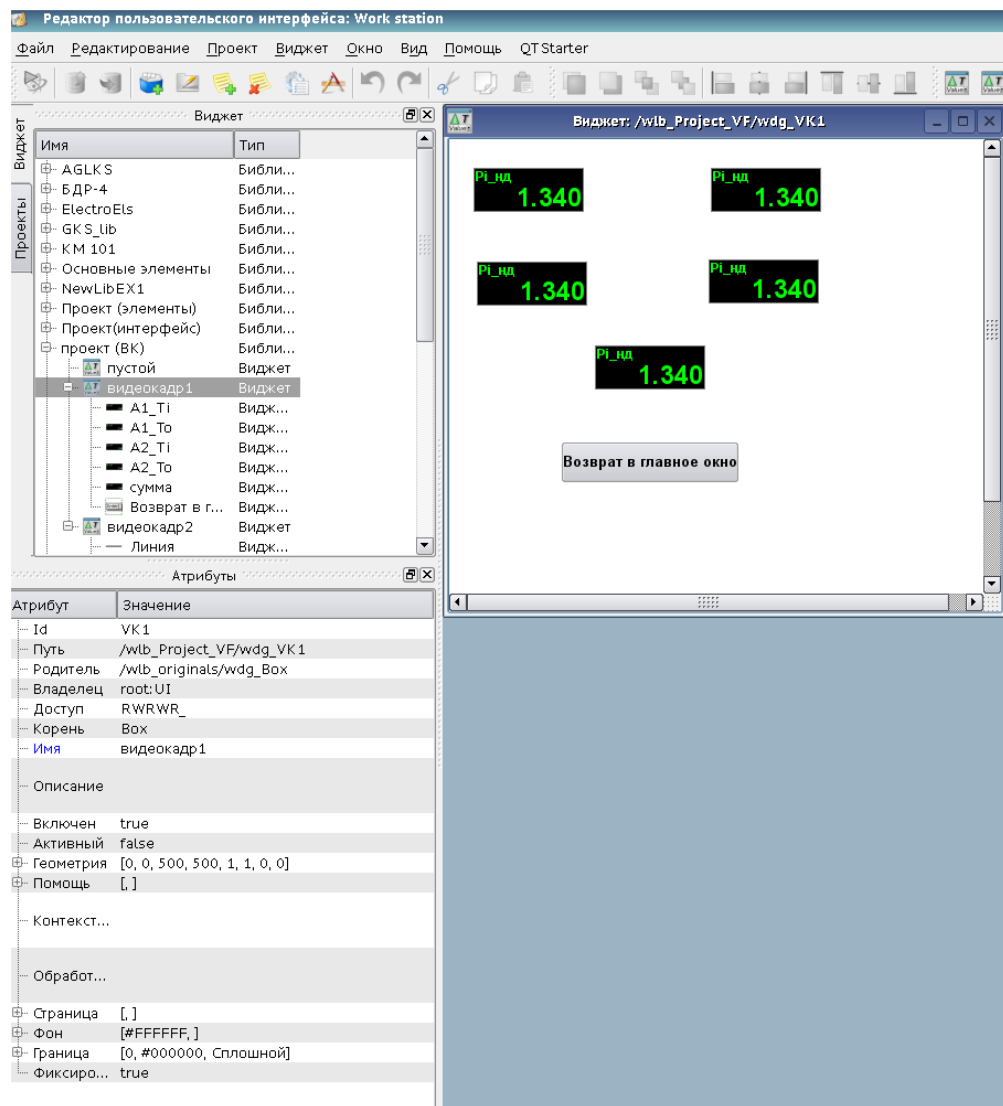


Рисунок 37

И скопируем эту кнопку на «Видеокадр 2», ID укажем такой же как и для «Видеокадр1» – «exit». Сохраним сделанные изменения.

## 1.7 Создание проекта

Для проверки работоспособности созданных нами видеокладов в рамках СКАДА создадим проект на базе наших библиотек элементов.

В визуальной компоненте выбрать в верхнем меню пункт «Проект» и в появившемся контекстном меню выбрать пункт «Новый проект» (рисунок 38).

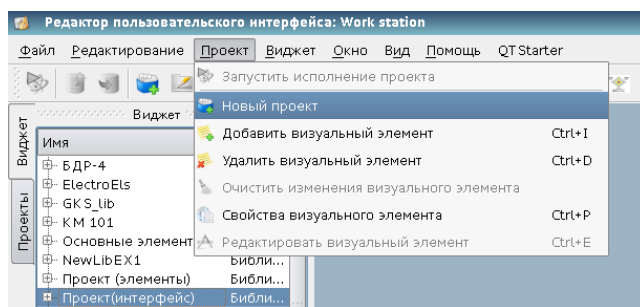


Рисунок 38

В появившемся окне указать ID (идентификатор) и имя библиотеки в соответствии с рисунком 39 и нажать кнопку «Принять».

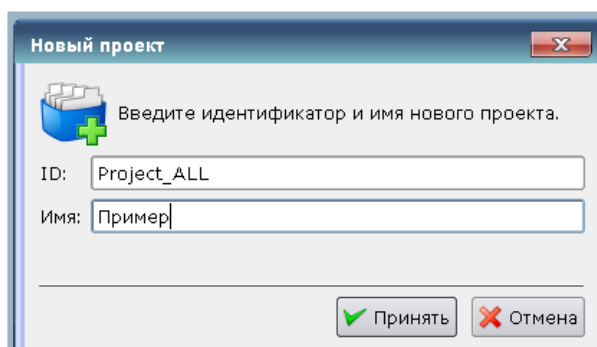


Рисунок 39

Выбрать левую вертикальную вкладку «Проекты» и в появившемся списке выбрать элемент «Проект». Открыть для редактирования свойства элемента, нажав сочетание клавиш «Ctrl+P», или выбрав из списка всплывающего меню и в появившемся окне раскрыть список «БД контейнера». В этом списке необходимо выбрать БД для хранения библиотеки визуальных элементов. Элемент списка должен иметь имя вида «PostgreSQL.<Имя БД>.<Имя библиотеки>». Где <Имя БД> соответствует ID ранее созданной БД - «PROJECT\_DB», <Имя библиотеки> соответствует ID настраиваемой

библиотеки «PROJECT\_ALL». После выбора элемента нажать кнопку «Заккрыть» (рисунок 40).

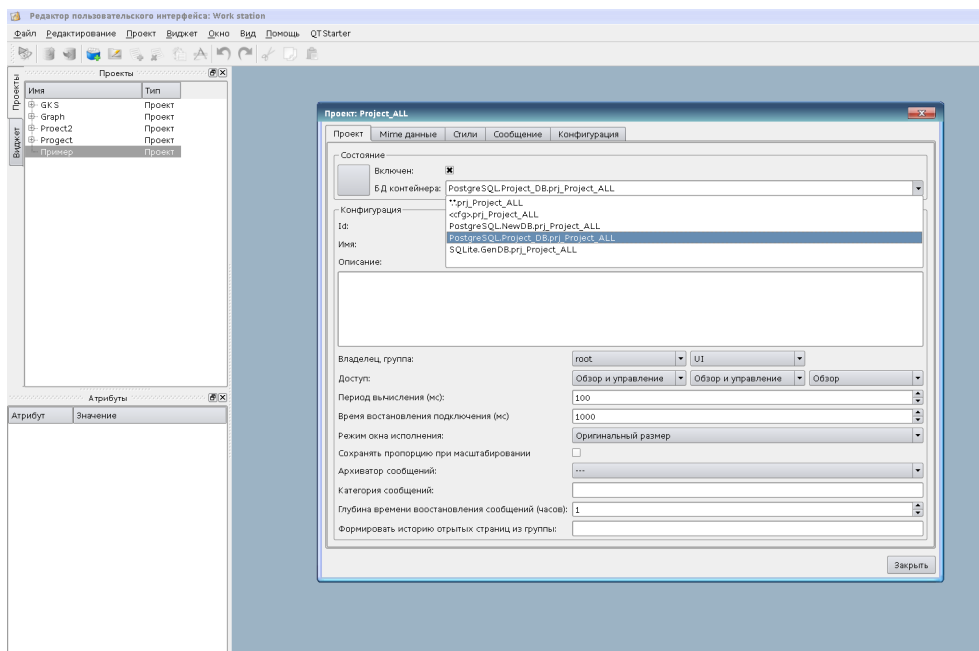


Рисунок 40

Сохранить проект в соответствии с 1.2.3.

### 1.7.1 *Настройка окна визуального представления*

Для настройки окна визуального интерфейса выбрать в меню пункт «Вид» и в появившемся списке (рисунок 41) оставить выбранными созданные нами библиотеки: «PROJECT\_UI», «PROJECT\_EL», «PROJECT\_VF».

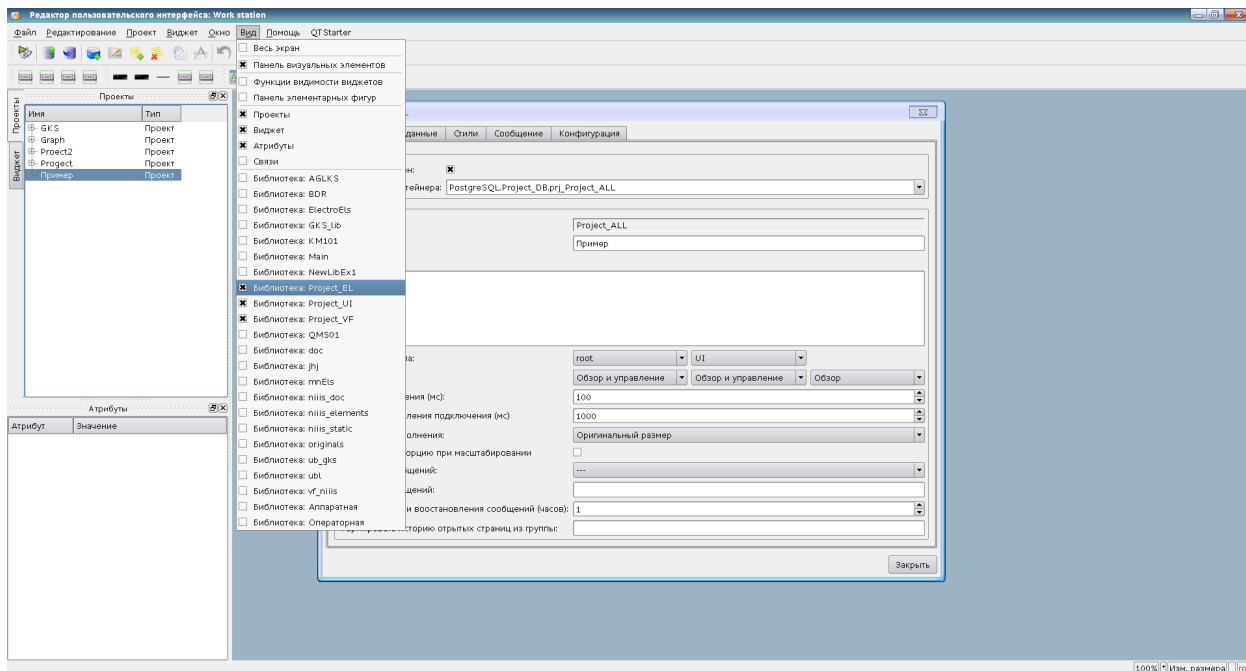


Рисунок 41

Выбрать левую вертикальную вкладку «Проекты» и в появившемся списке выбрать элемент «Проект».

### 1.7.2 Добавление элемента в проект.

На панели инструментов окна визуальной компоненты (рисунок 42) нажать на кнопку, добавляющую в выбранный проект элемент «Главное окно». Всплывающая подсказка данной кнопки содержит текст вида: «Добавление виджета основанного от '/wlb\_<ID\_библиотеки>/wdg\_<ID\_элемента>'. Так как «Главное окно» имеет ID - «MAIN», а содержащая его библиотека ID - «Project\_UI», то следует искать кнопку с описанием «Добавление виджета основанного от '/wlb\_Project\_UI/wdg\_Main'».

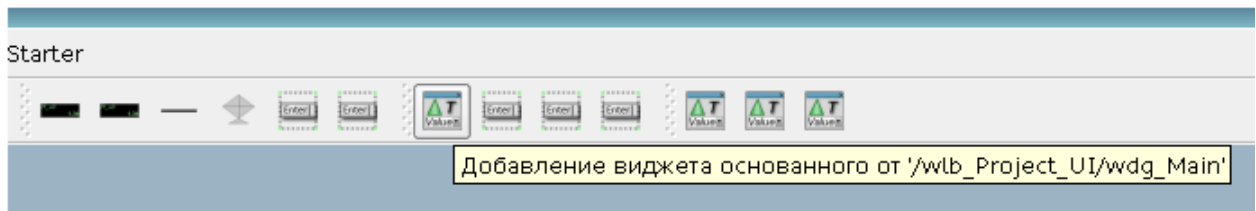


Рисунок 42

В появившемся окне задать в поле ввода ID - «MAIN», в поле «Имя» - «Главное окно».

Выбрать левую вертикальную вкладку «Проекты» и в появившемся списке выбрать элемент «Проект», затем содержащийся в нём элемент «Главное окно». Открыть свойства виджета, нажав сочетание клавиш «Ctrl+P», или из всплывающего меню. В появившемся окне поставить галочку для состояния страницы – «включен», затем раскрыть список «Тип страницы» и выбрать «Контейнер и шаблон». Тип страницы «Контейнер и шаблон» совмещает в себе функции шаблона и контейнера. После чего закрыть окно свойств.

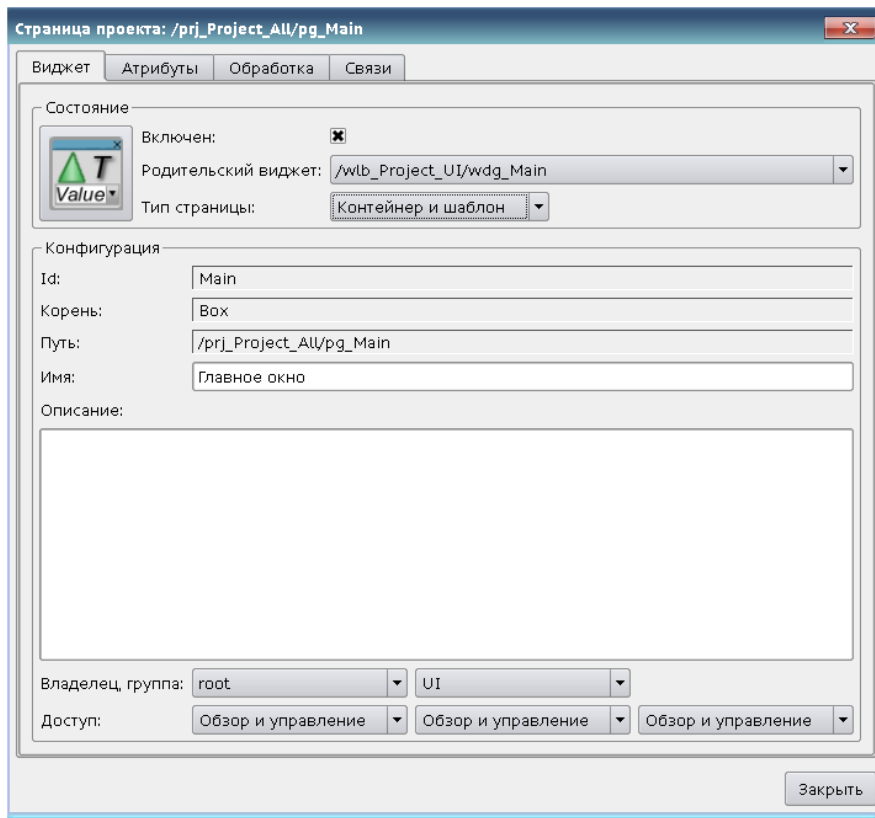


Рисунок 43

Если открыть «Главное окно» для редактирования, то мы увидим окно уже созданного нами виджета с идентичным названием в библиотеке «Проект(интерфейс)».

### 1.7.3 Подключение мнемосхем и источников данных к проекту.

Для подключения к Проекту наших мнемосхем необходимо в окружении Проекты выделить наш Проект и выбрать на верхней панели инструментов добавление визуального элемента, основанного от '/wlb\_Project\_VF/wdg\_VK1'. Создаваемому элементу присвоим ID «VK1» и имя «Видеокадр1». Точно так же добавим «Видеокадр 2», основанный на '/wlb\_Project\_VF/wdg\_VK2' с ID «VK2». Сохраним все видеокадры в проекте.

Далее поочередно открываем Видеокадры для редактирования («Ctrl+E») и выбрав кнопку «Возврат в главное окно» напишем обработку событий: `ws_BtPress::open://pg_Main`.

После чего сохраняем видеокадры в проекте. В результате наших действий дерево проектов имеет следующий вид (рисунок 44).

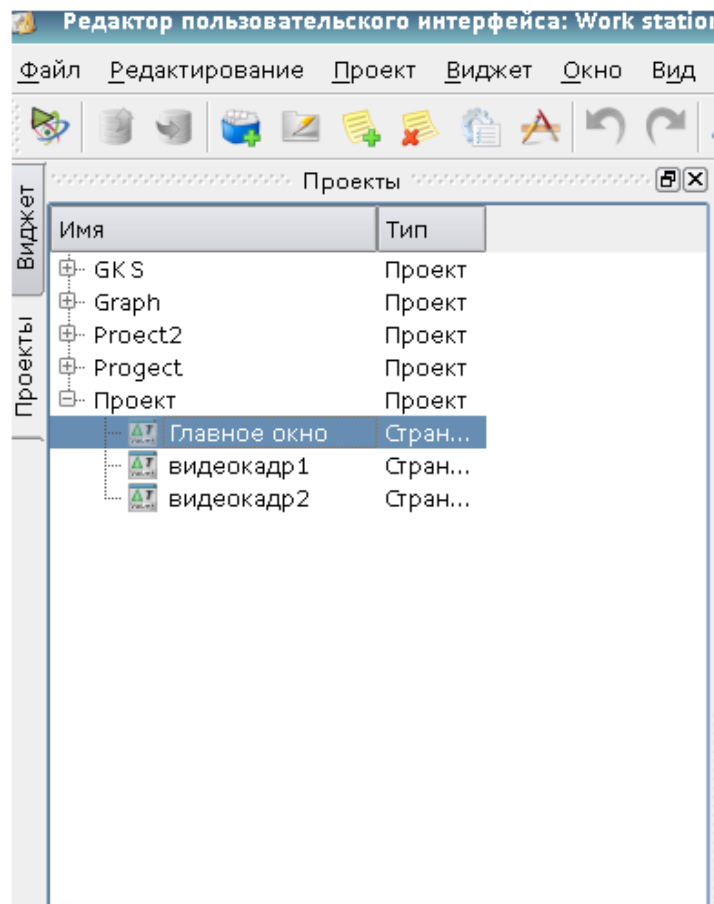


Рисунок 44

Далее для «Видеокадр1» зададим адреса параметров для отображения в виде аналогового значения. Для этого, находясь в проекте, откроем свойства визуального элемента «Видеокадр 1» (Ctrl+P) и в окне Связи установим значения Параметров pName и pVal. Имена параметров (pName) определим как имена отображаемых аналоговых сигналов: A1\_1, A1\_2, A2\_1, A2\_2, C. А в полях значений параметров (pVal) укажем адреса переменных созданных в 1.3 на Логическом уровне в контроллере Primer. Например: «prm:/LogicLev/primer/A1\_T1/const» (при этом строка значений составляется путем последовательного выбора из доступных элементов всплывающего меню). Установленные параметры показаны на рисунке 45.

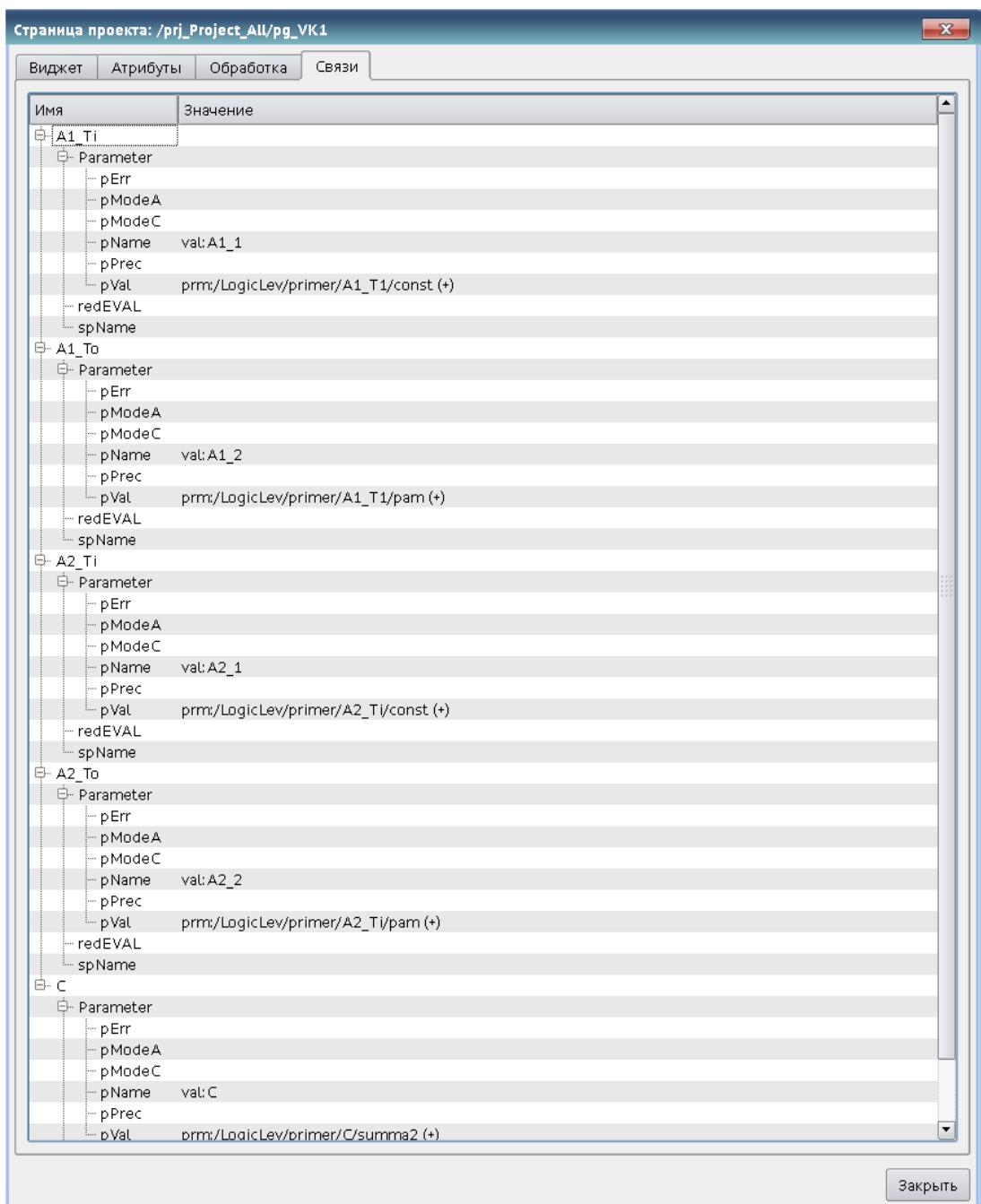


Рисунок 45

Закроем окно диалога свойств, сохраним нашу мнемосхему и проверим, что получилось.

#### 1.7.4 Исполнение проекта

Для проверки работы нашего проекта запустим его на исполнение, нажав на иконку



При безошибочной конфигурации на экране появится окно, в котором можно установить дату, и две кнопки-перехода, при нажатии на которые отображаются 2

мнемосхемы (рисунок 46). Значения, отображаемые на Видеокадре 1 соответствуют значениям параметров контроллера «Primer» Логического уровня (рисунок 47).



Рисунок 46

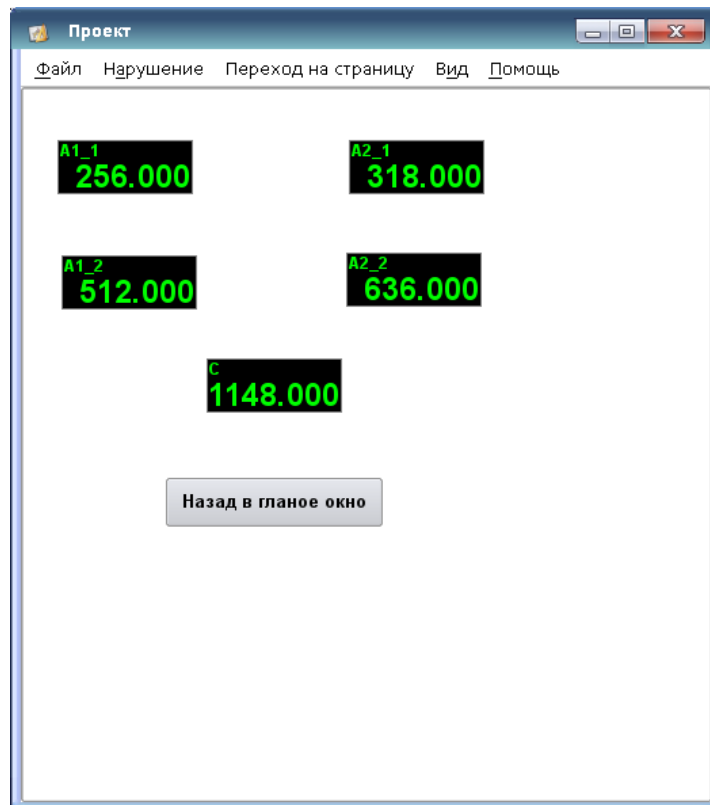


Рисунок 47

Кран на Видеокадре 2 изменяет свой цвет с желтого на зеленый при изменении значения атрибута ON на val:1 в окне «Связи» этого виджета (рисунок 48).



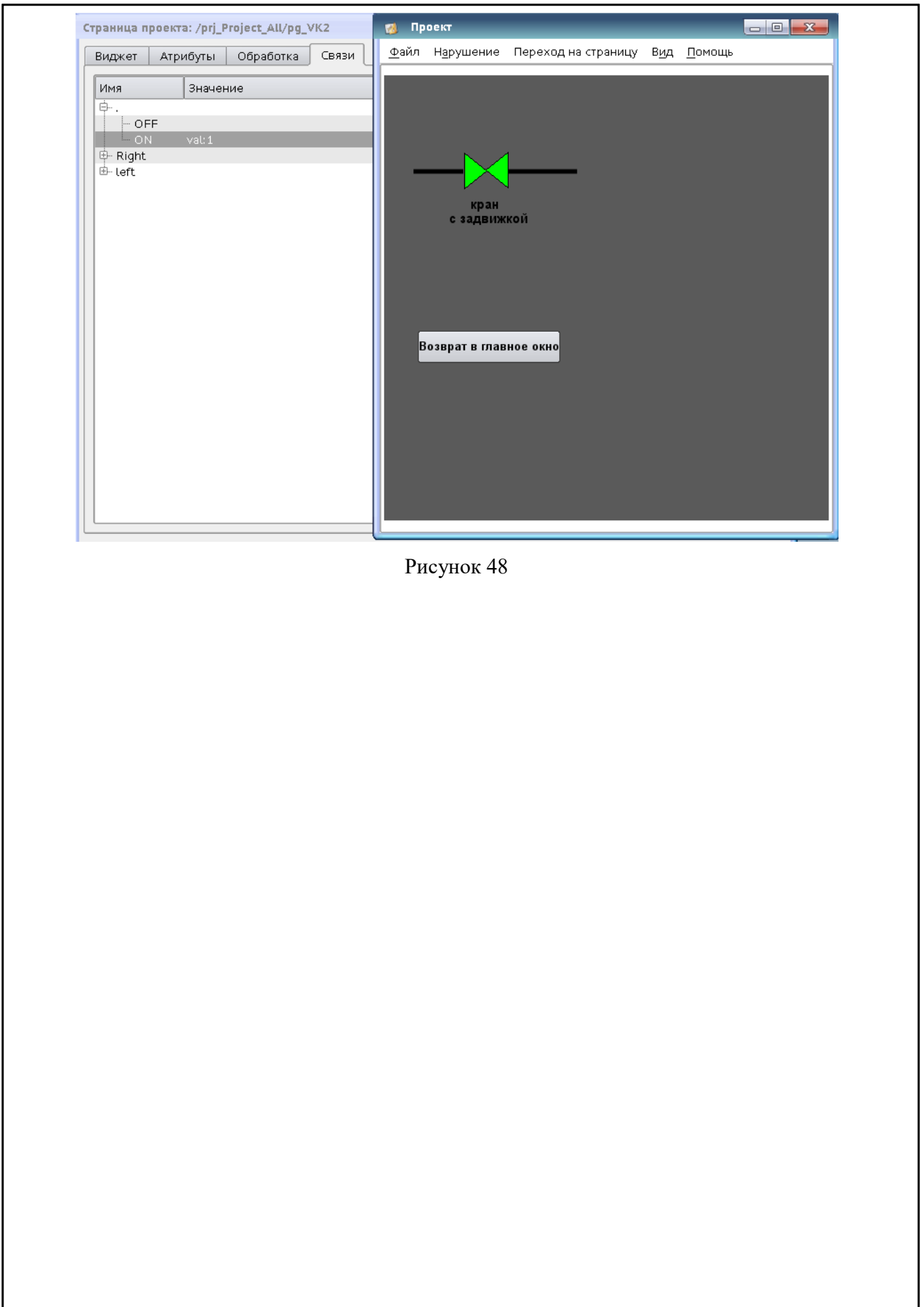


Рисунок 48

## 2 Настройка различных типов источников данных

### 2.1 Модуль источника данных ModBUS

#### 2.1.1 Конфигурирование сбора данных по протоколу ModBUS

Создадим объект контроллера для опроса по протоколу ModBUS и получим эти данные, тем самым фактически реализовав задачу опроса реальных данных (от настоящего внешнего устройства наша конфигурация будет отличаться адресом устройства, адресами регистров ModBUS и возможно интерфейсом взаимодействия).

Для добавления нового контроллера необходимо открыть в конфигураторе страницу модуля «Сбор данных» → «Модуль» → «ModBUS» и в контекстном меню пункта "ModBUS" нажать "Добавить" (рисунок 49).

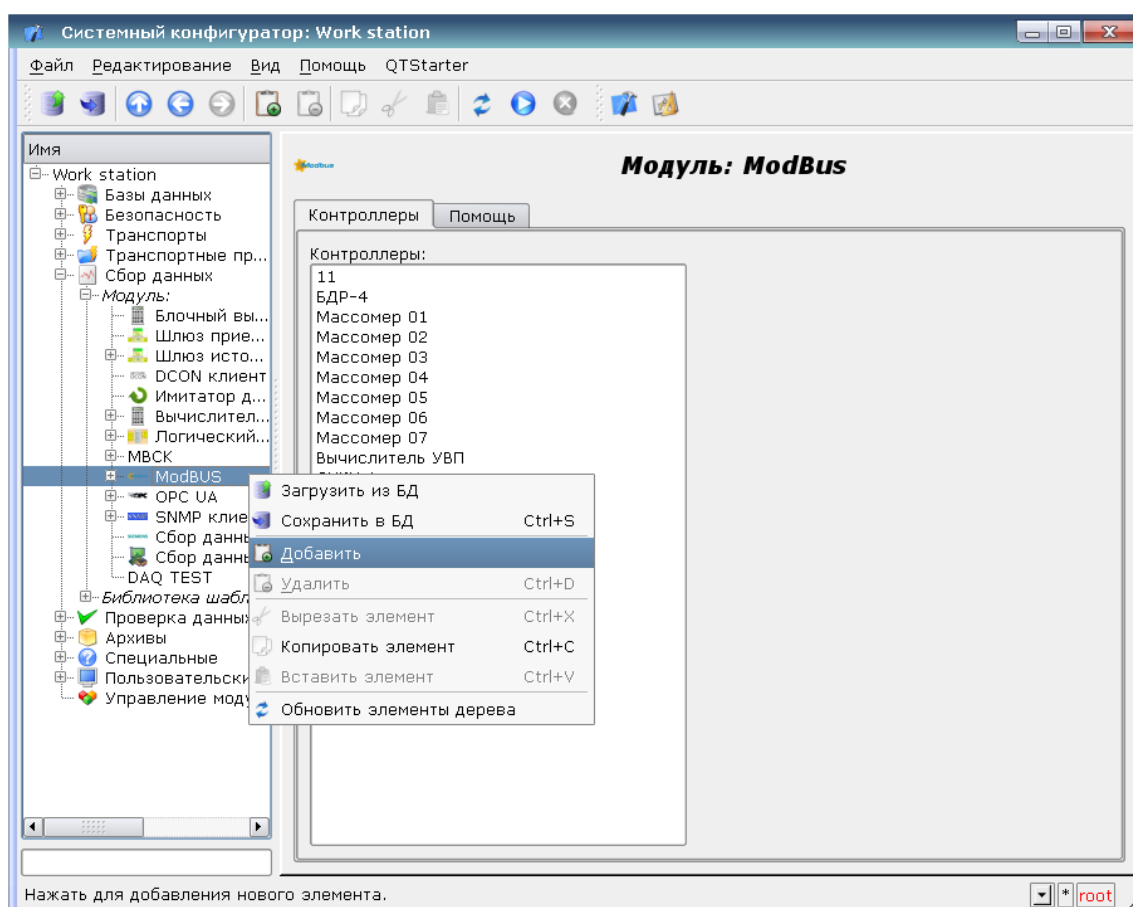


Рисунок 49

В результате появится окно диалога (рисунок 50) с предложением ввести идентификатор и имя нового контроллера (для написания имени и идентификатора действуют ранее описанные правила). Введем идентификатор "primer" и имя "primer".

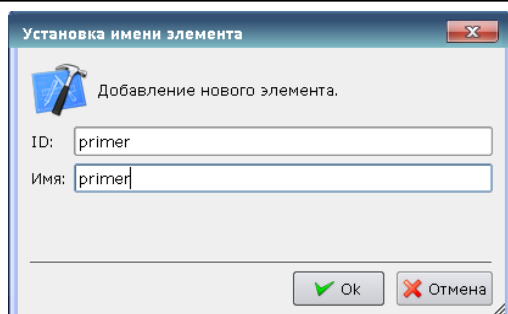


Рисунок 50

После подтверждения появится объект нового контроллера. Выбрав который можно провести его настройку.

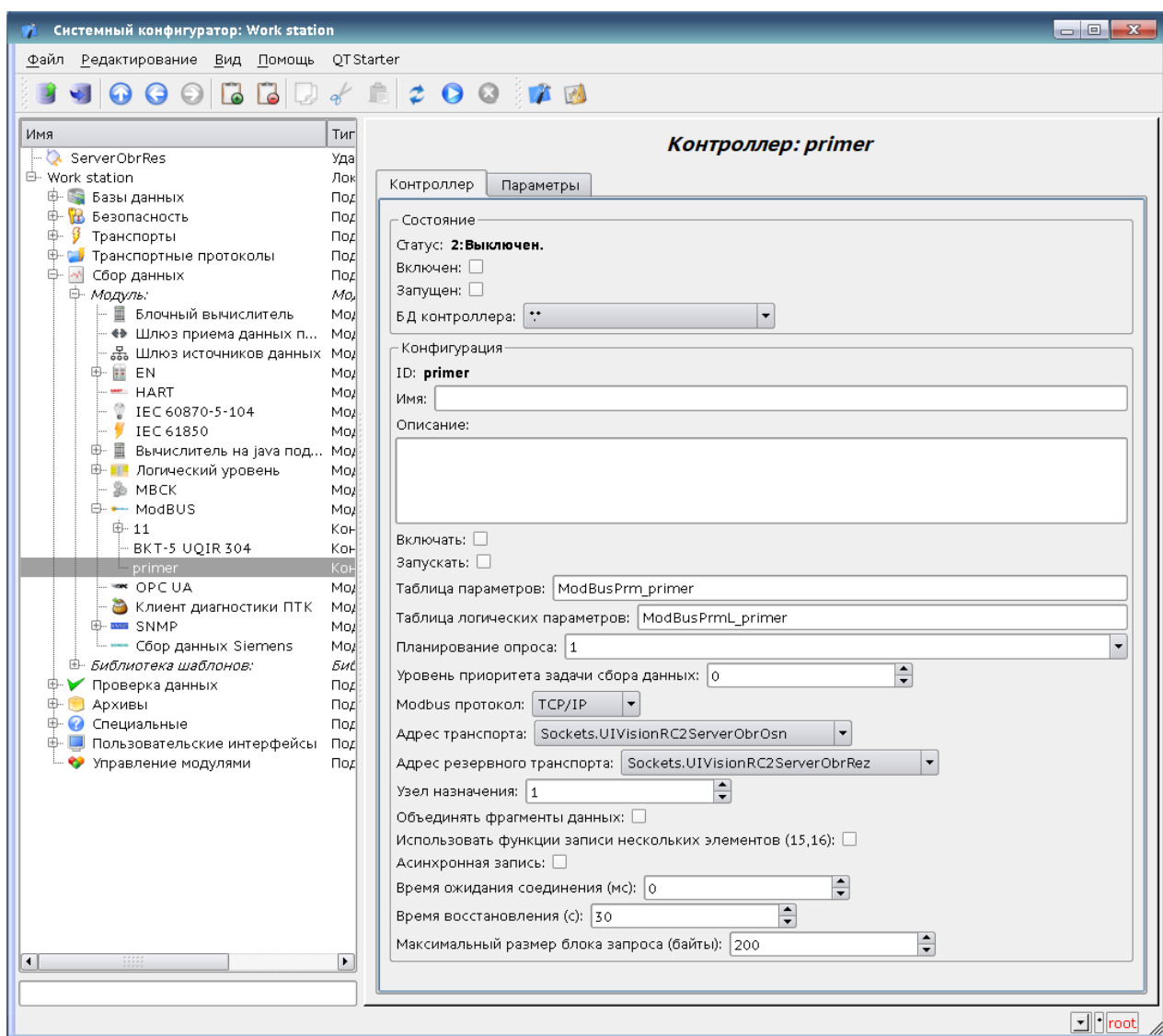


Рисунок 51

Перед конфигурацией связи со своим контроллером предварительно необходимо, из документации на контроллер, выяснить настройки его сетевых интерфейсов и протоколов, а также получить таблицу назначения внешних и внутренних сигналов контроллера на номера регистров "ModBus".

С помощью страницы объекта контроллера в разделе "Состояние" можно в первую очередь оценить текущее состояние объекта контроллера и реальное состояние связи с физическим контроллером, а также оперативно его менять. Так, поле "Статус" содержит код ошибки и текстовое описание текущего состояния связи с контроллером, в нашем случае объект контроллера выключен. Мы его можем включить и запустить, установив флажки напротив соответствующих полей. Включенный объект контроллера инициализирует объекты параметров, запущенный же запускает задачу опроса и предоставляет возможность передавать данные в контроллер через атрибуты параметров. Поле БД указывает на то, в какой БД хранится конфигурация данного объекта. Нам устроит хранение в главной БД, т.е. оставим по умолчанию.

В разделе "Конфигурация" непосредственно содержится конфигурация объекта контроллера:

- "Идентификатор" и "Имя" содержат названия, которые мы вводили при создании объекта. Имя контроллера можно изменить в данном окне, а изменение идентификатора возможно только путем вырезания (Ctrl+X) и вставки (Ctrl+V) объекта с его переименованием.
- "Описание" может содержать развёрнутую характеристику и назначение объекта контроллера.
- "Включать" и "Запускать" указывают на состояние в которое следует переводить объект контроллера при запуске СКАДА. Установим оба поля.
- "Таблица параметров" — содержит имя таблицы БД, в которой будет храниться конфигурация параметров данного контроллера. Оставим по умолчанию.
- «Таблица логических параметров» - содержит имя таблицы БД, в которой будет храниться конфигурация логических параметров данного контроллера. Оставим по умолчанию.
- "Планирование опроса" — содержит конфигурацию планировщика для запуска задачи опроса. Получить описание формата конфигурации данного поля можно из всплывающей подсказки. Одиночная цифра указывает на периодичность запуска в секундах.

- "Уровень приоритета задачи сбора данных" — указывает насколько приоритетная данная задача (от -1 до 99). Приоритеты выше нуля имеют смысл только при запуске СКАДА от привилегированного пользователя. Оставим это поле без изменений.
- "ModBUS протокол" — указывает на вариант протокола ModBUS. Возможны варианты протокола "TCP/IP", "RTU" и "ASCII". Выберем "TCP/IP". Варианты протоколов "RTU" и "ASCII" нужно устанавливать в случае связи с контроллером посредством последовательных интерфейсов (например, "RS-485").
- "Адрес транспорта" — указывает на исходящий транспорт подсистемы "Транспорты", который используется для соединения с контроллером. При выборе "TCP/IP" будет необходим транспорт в модуле Sockets, а при выборе "RTU", "ASCII" и последовательного интерфейса – транспорт в модуле Serial. Создание исходящего транспорта в "Sockets" описано далее. Если транспорт уже создан, то достаточно выбрать его из выпадающего списка.
- "Адрес резервного транспорта" — позволяет выбрать резервный исходящий транспорт подсистемы "Транспорты", который будет использоваться в случае отсутствия ответа от основного транспорта.
- "Узел назначения" — указывает узел источника данных или контроллера в сети ModBUS. В нашем случае - "1".
- "Объединять фрагменты данных" — включает объединение несмежных фрагментов регистров в один блок запроса, до 100 регистров, вместо генерации отдельных запросов. Позволяет уменьшить общее время опроса. Установим эту опцию.
- "Использовать функции записи нескольких элементов (15,16)" — вместо функций одноэлементной записи будут использованы многоэлементные. Оставим неизменным.
- "Время ожидания соединения" — указывает в течение какого времени ожидать ответа от контроллера и по истечению которого сообщать об ошибке связи. Ноль указывает на использование времени транспорта. Оставим без изменений.
- "Асинхронная запись" — установленный флаг реализует асинхронный способ передачи данных;
- "Время восстановления" — указывает на время в секундах, через которое, в случае отсутствия связи, повторять попытку восстановить соединение.

– "Максимальный размер блока запроса (байты)" — устанавливает максимальный размер блока групповых запросов регистров и битов, в байтах. Полезен для некоторых контроллеров с подобным ограничением. Оставим неизменным.

Сохраним наши изменения в БД (см. 1.2.3).

Далее необходимо создать Выходной транспорт в модуле "Sockets" «Транспорты» → «Сокеты» посредством контекстного меню (рисунок 52) с названием контроллера: "primer" и именем "primer".

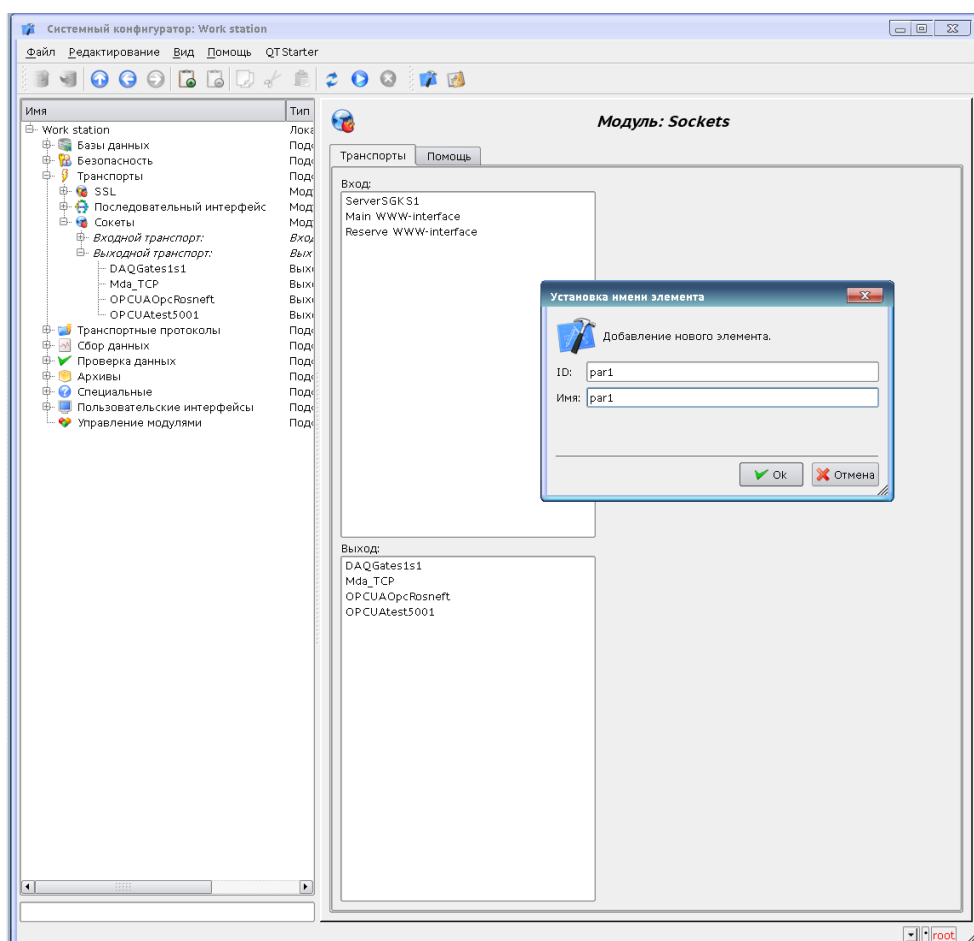


Рисунок 52

Страница конфигурации полученного исходящего транспорта приведена на рисунке 53. Эта страница также содержит раздел состояния и оперативного управления. В поле "Статус" содержится текстовое описание текущего состояния транспорта. Мы его можем запустить на исполнение, установив флажок напротив соответствующего поля. Выполняющийся объект транспорта инициирует соединение с внешним узлом. Поле БД указывает на то, в какой БД хранится конфигурация данного объекта. Нас устроит хранение в главной БД.

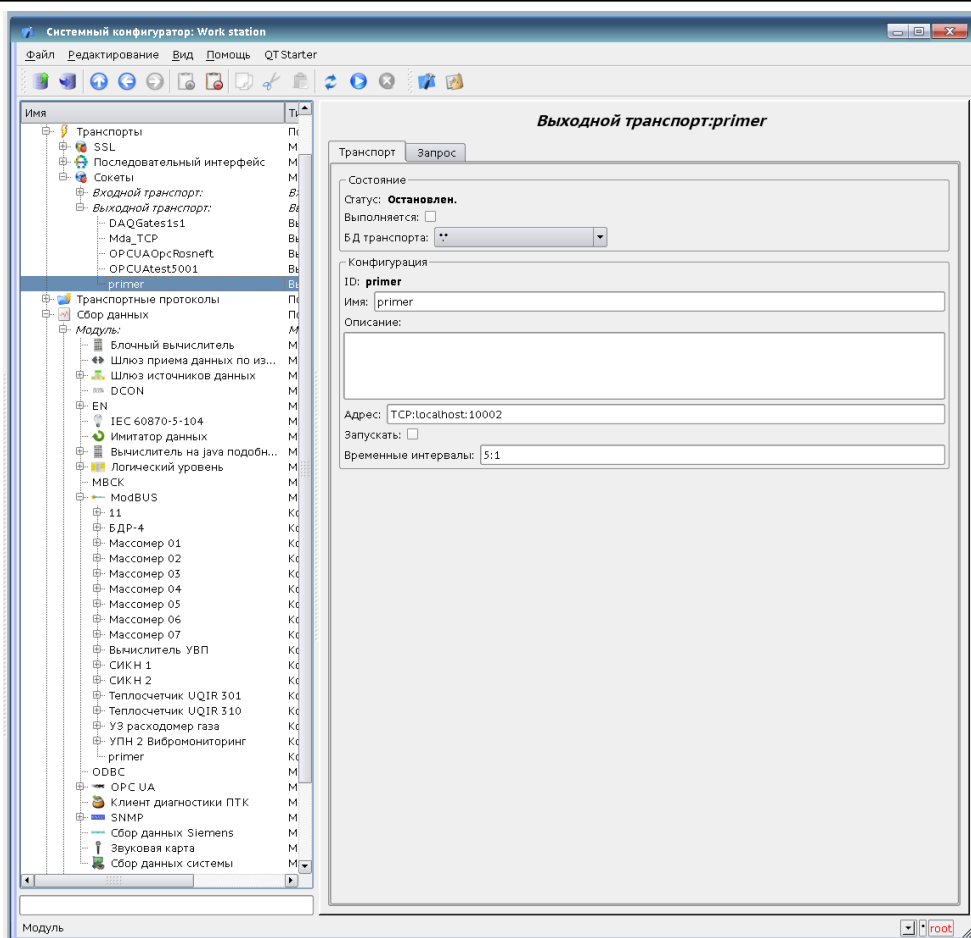


Рисунок 53

В разделе "Конфигурация" непосредственно содержится конфигурация объекта транспорта:

- "Идентификатор" и "Имя" содержат названия, которые мы вводили при создании объекта.
- "Описание" — может содержать развёрнутую характеристику и назначение объекта.
- "Адрес" — указывает тип, адрес и режим соединения с удалённой станцией. Ознакомиться с форматом записи можно из всплывающей подсказки. Установим это поле в значение "TCP:localhost:10502".
- "Запускать" — указывает на то, в какое состояние переводить объект при запуске СКАДА.
- "Временные интервалы" — указывают продолжительность ожидания ответа от удалённой станции. Ознакомиться с форматом записи можно из всплывающей подсказки. Оставим значение неизменным.

Транспорты других типов создаются аналогичным образом, а их конфигурация их отличается обычно только форматом записи адреса и таймаутов. Для транспорта модуля "Serial" в поле адреса указывается путь к последовательному устройству, скорость, и формат. Для переходников *USB →Serial* адрес нужно узнать в операционной системе, например, консольной командой "`$ dmesg`", сразу после подключения переходника.

Сохраним объект транспорта и вернёмся к конфигурационному полю "Адрес транспорта" объекта контроллера, где выберем адрес "Sockets.primer". На этом настройка объекта контроллера закончена, включим его: установив флаг "Включен".

Объект "Параметр" контроллера позволяет описать перечень данных, получаемых у контроллера и передать их в окружение СКАДА.

Для добавления нового объекта параметра контроллера "primer" необходимо открыть в конфигураторе страницу контроллера и нажав правую кнопку мыши выбрать пункт «Добавить». Для нового объекта ввести идентификатор "par1" и имя "par1".

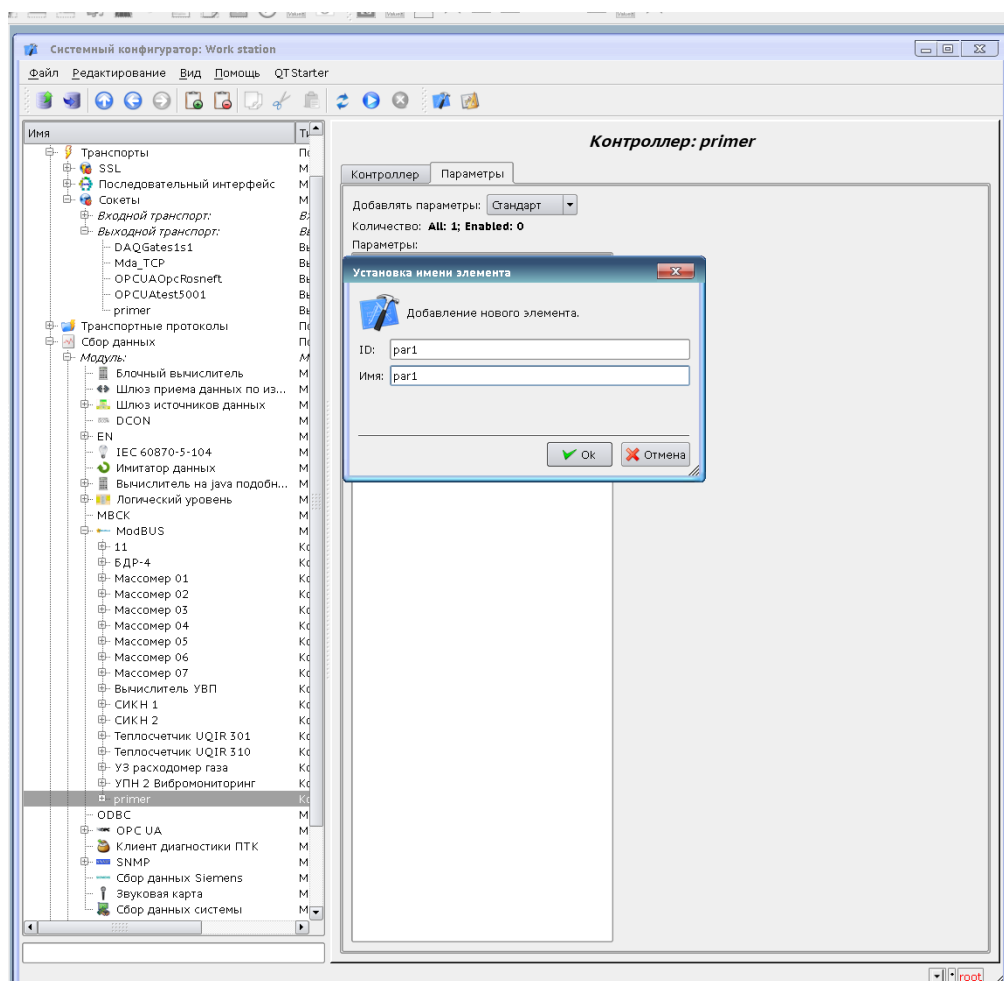


Рисунок 54



Страница конфигурации полученного параметра приведена на рисунке 55. Эта страница содержит раздел состояния и оперативного управления. В поле "Тип" содержится идентификатор типа параметра, в нашем случае тип "Стандартный" (std). Параметр мы можем включить, установив флажок напротив соответствующего поля. Включенный параметр участвует в процессе обмена с контроллером.

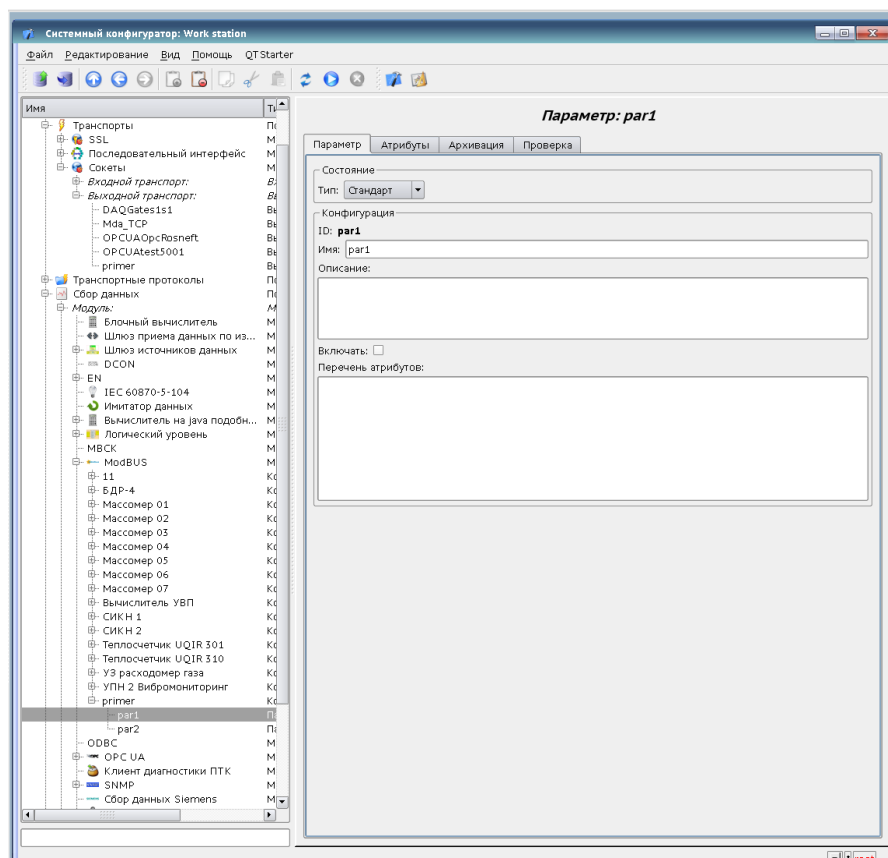


Рисунок 55

В разделе "Конфигурация" непосредственно содержится конфигурация объекта параметра:

- "Идентификатор" и "Имя" содержат названия, введенные при создании объекта.
- "Описание" — может содержать развёрнутую характеристику и назначение объекта.
- "Включать" — указывает на то, в какое состояние переводить объект при запуске СКАДА. Установим поле.
- "Перечень атрибутов" — содержит конфигурацию атрибутов параметров в соотношении их с регистрами и битами "ModBUS". Ознакомиться с форматом записи можно из всплывающей подсказки (рисунок 56).

Список конфигурации атрибутов. Список формируется строками в формате: "{dt}:{numb}:{rw}:{id}:{name}".  
Где:  
dt - ModBus тип данных (R-регистр[3,6(16)], C-бит[1,5(15)], RI-регистр входа[4], CI-бит входа[2]);  
R и RI могут быть расширены суффиксами: i2-Int16, i4-Int32, f-Float, b5-Bit5, s-Строка;  
Начните с символа '#' для комментирования строки;  
numb - адрес ModBus устройства (десять, шестн., или восьмеричн.) [0...65535];  
rw - режим чтения/записи (r-чтение; w-запись, rw-запись и чтение);  
id - идентификатор создаваемого атрибута;  
name - имя создаваемого атрибута.  
Примеры:  
"R:0x300:rw:var:Variable" - доступ к регистру;  
"C:100:rw:var1:Variable 1" - доступ к биту;  
"R\_f:200:r:float:Float" - получить вещественное из регистров 200 и 201;  
"R\_i4:300,400:r:int32:Int32" - получить int32 из регистров 300 и 400;  
"R\_b10:25:r:rBit:Reg bit" - получить бит 10 из регистра 25;  
"R\_s:15,20:r:str:Reg blk" - получить строку, блок регистров, из регистра 15 и размером 20.

### Рисунок 56

Установим содержимое этого текстового поля в:

```
R:100:r:Ti:T вход  
R:101:r:To:T выход  
R:102:rw:Cw:Производ
```

Таким же образом создадим второй параметр: "par2" с именем "par2". Перечень атрибутов для него установим в:

```
R:103:r:Ti:T вход  
R:104:r:To:T выход  
R:105:rw:Cw:Производ
```

Сохраним оба объекта параметра. Теперь мы можем включить и запустить наш контроллер для инициации обмена. Для этого вернёмся на страницу нашего объекта контроллера и в разделе "Состояние" установим флажок "Запущен". В результате в поле "Статус" отразится результат подключенного обмена данными.

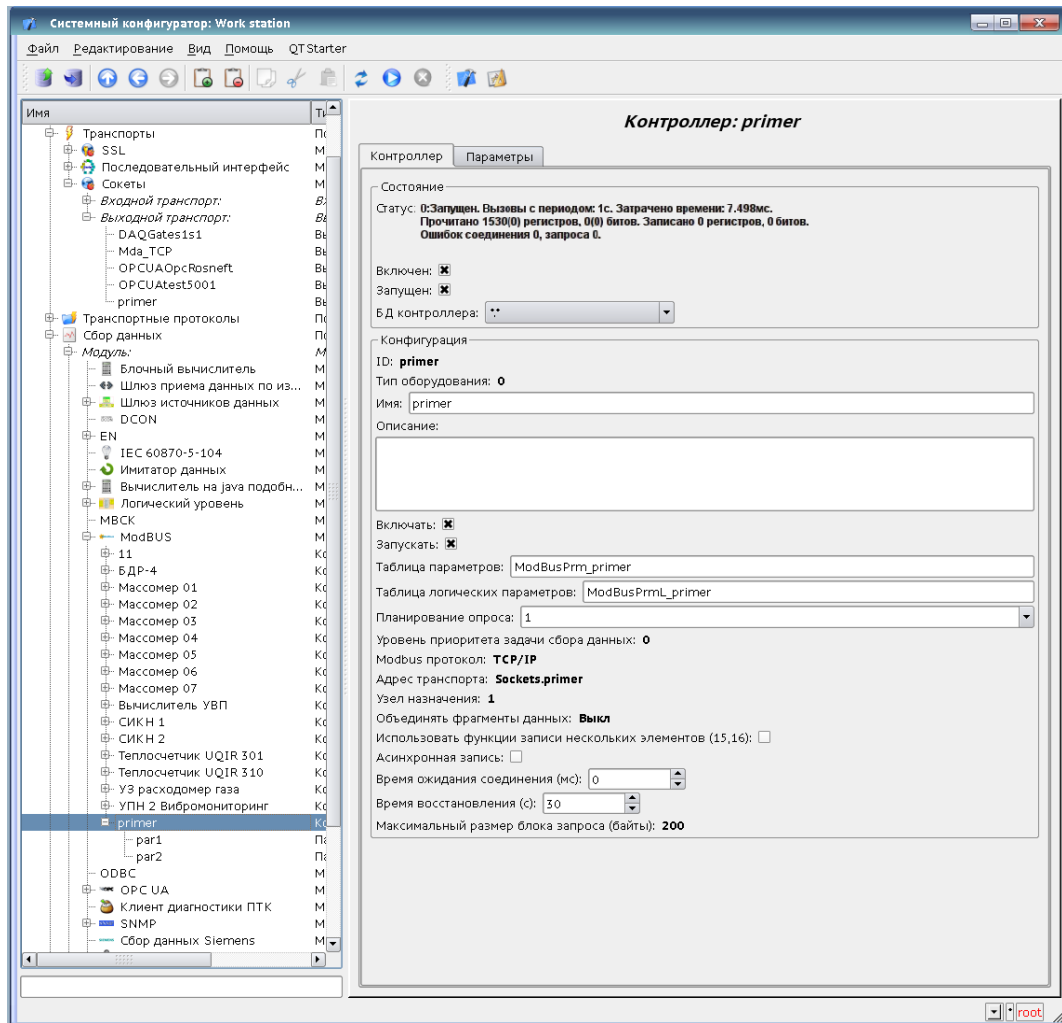


Рисунок 57

В случае успешного обмена с физическим контроллером на вкладке "Атрибуты" наших параметров отобразятся эти данные. Поскольку опрос производится регулярно и с периодичностью в секунду, то мы можем наблюдать их изменение, нажимая кнопку "Обновить текущую страницу" на панели инструментов.

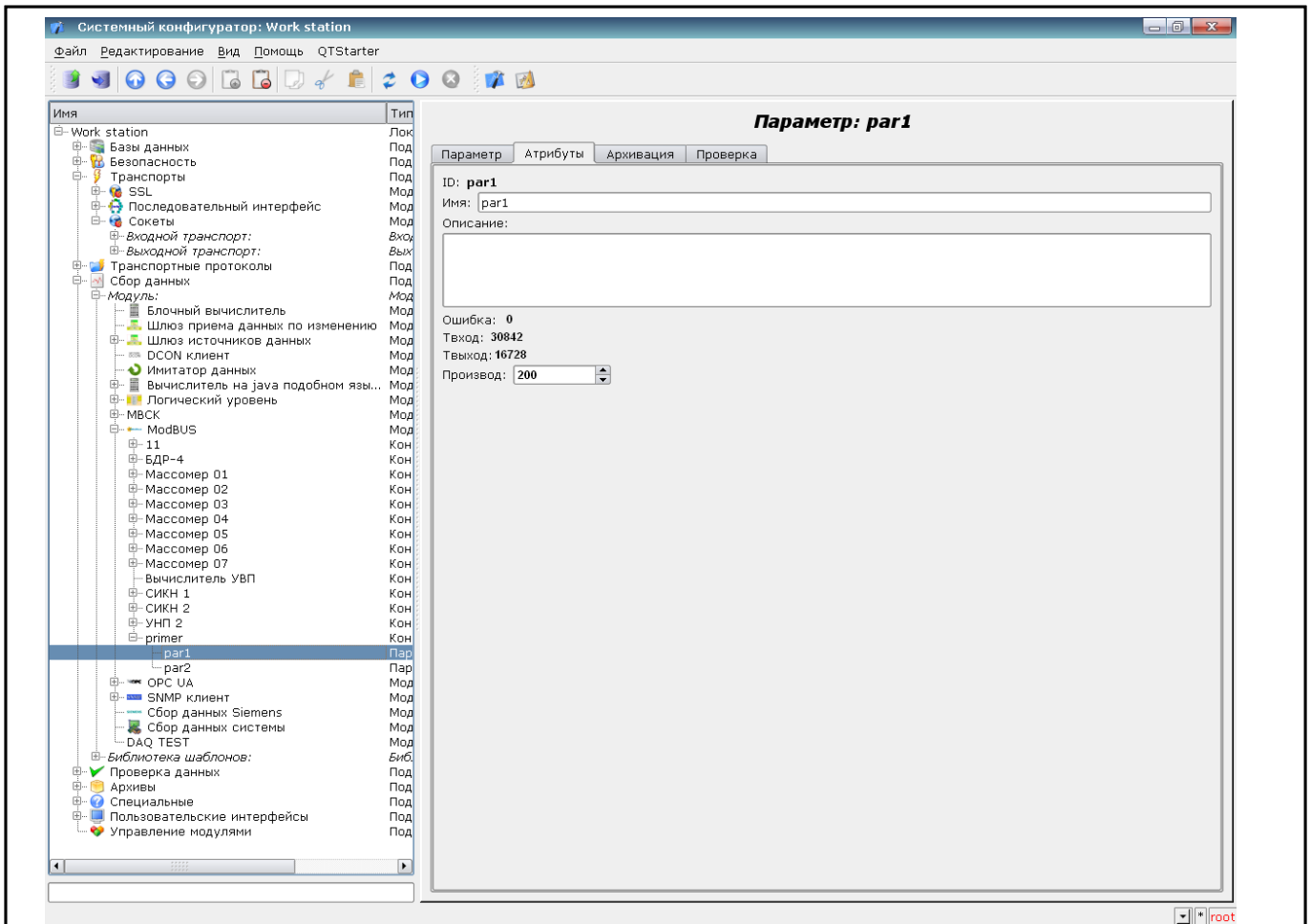


Рисунок 58

На этом конфигурация сбора данных считается законченной.

### 2.1.2 Конфигурирование обработки данных, полученных по протоколу ModBUS

Для возможности использования полученных в предыдущем пункте данных на «Логическом уровне» подсистемы «Сбор данных» необходимо создать объект библиотеки базовых шаблонов ("Сбор данных"→"Библиотека шаблонов"→"Базовые шаблоны") или воспользоваться уже имеющимися шаблонами, указывая адреса ModBus-регистров параметров. Далее описан первый способ.

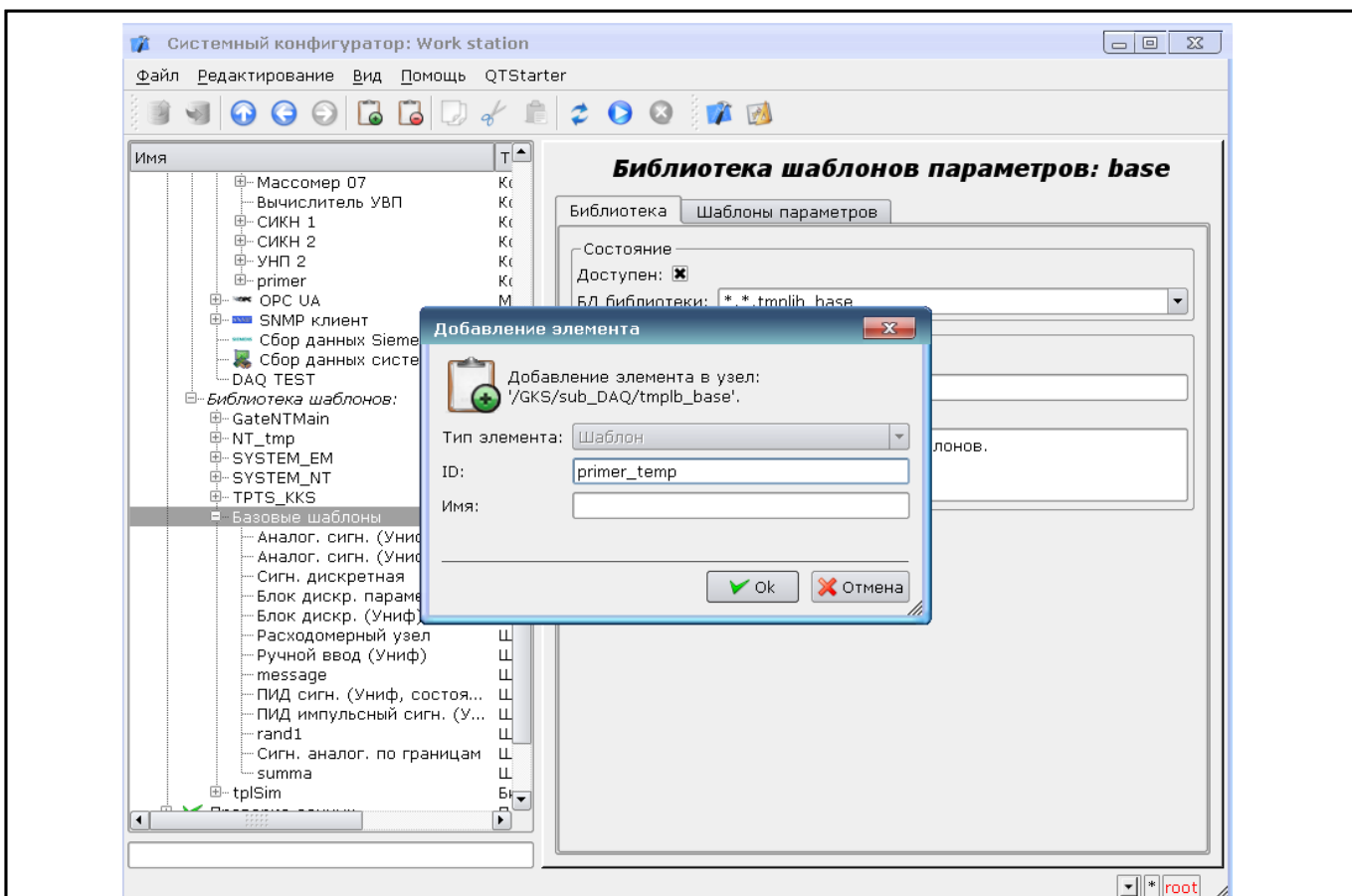


Рисунок 59

Основная конфигурация и формирование шаблона параметра сбора данных осуществляется во вкладке "IO" шаблона (рисунок 60). Создадим в шаблоне два свойства для входов ("TiCod", "ToCod"), два для выходов ("Ti","To") и один прозрачный ("Cw"). Свойствам "TiCod", "ToCod" и "Cw" установим флаг "Конфигурация" в "Связь", что позволит к ним подвязывать "сырой" источник. Параметрам "Ti" и "To" установим флаг "Атрибут" в "Только чтение", "Cw" в "Полный доступ" для формирования трёх атрибутов: два только на чтение и один на полный доступ, у результирующего параметра сбора данных.

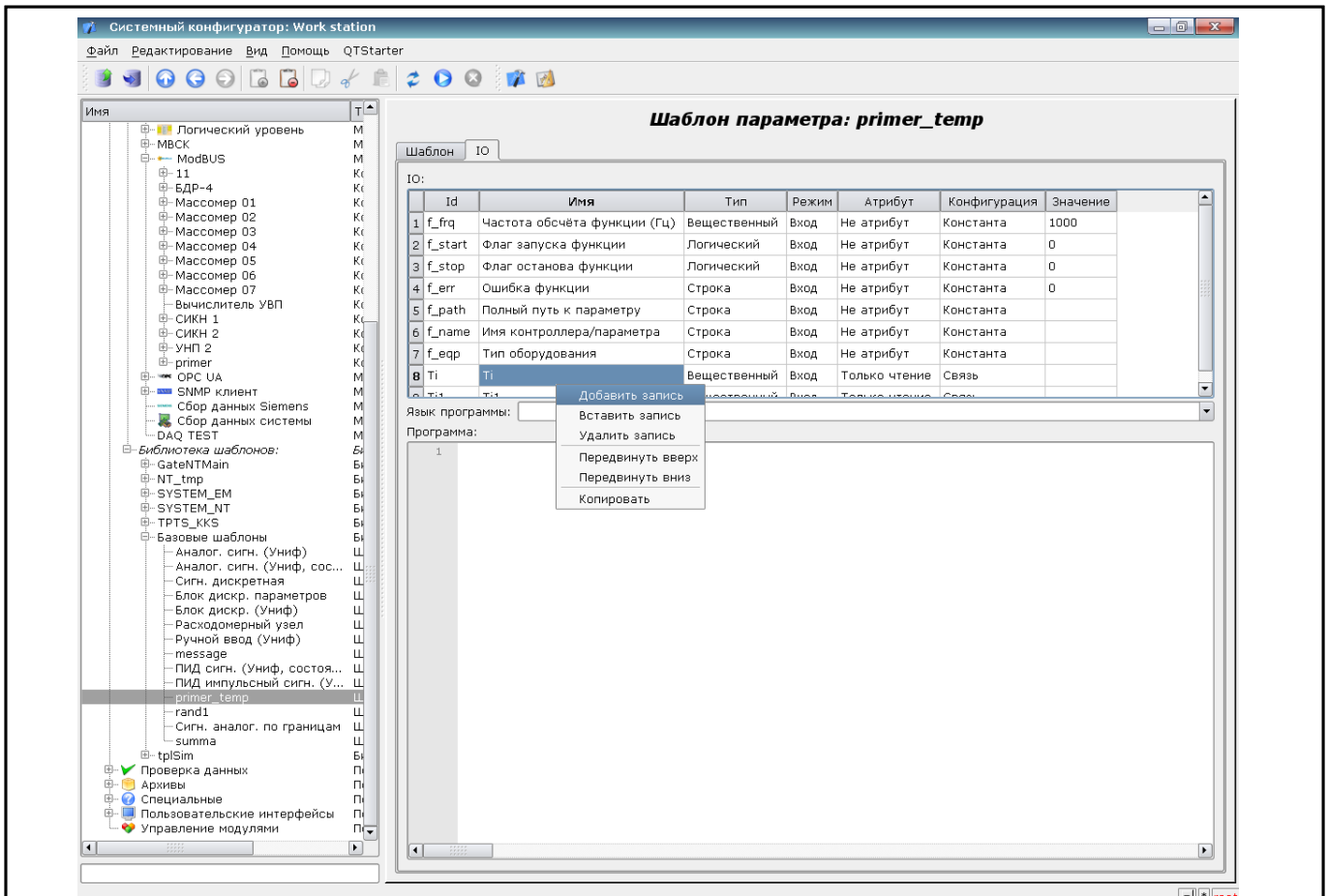


Рисунок 60

Язык программы установим в "JavaLikeCalc.JavaScript", а в поле описания программы введем следующий код:

```
Ti=150*TiCod/65536;  
To=100*ToCod/65536;
```

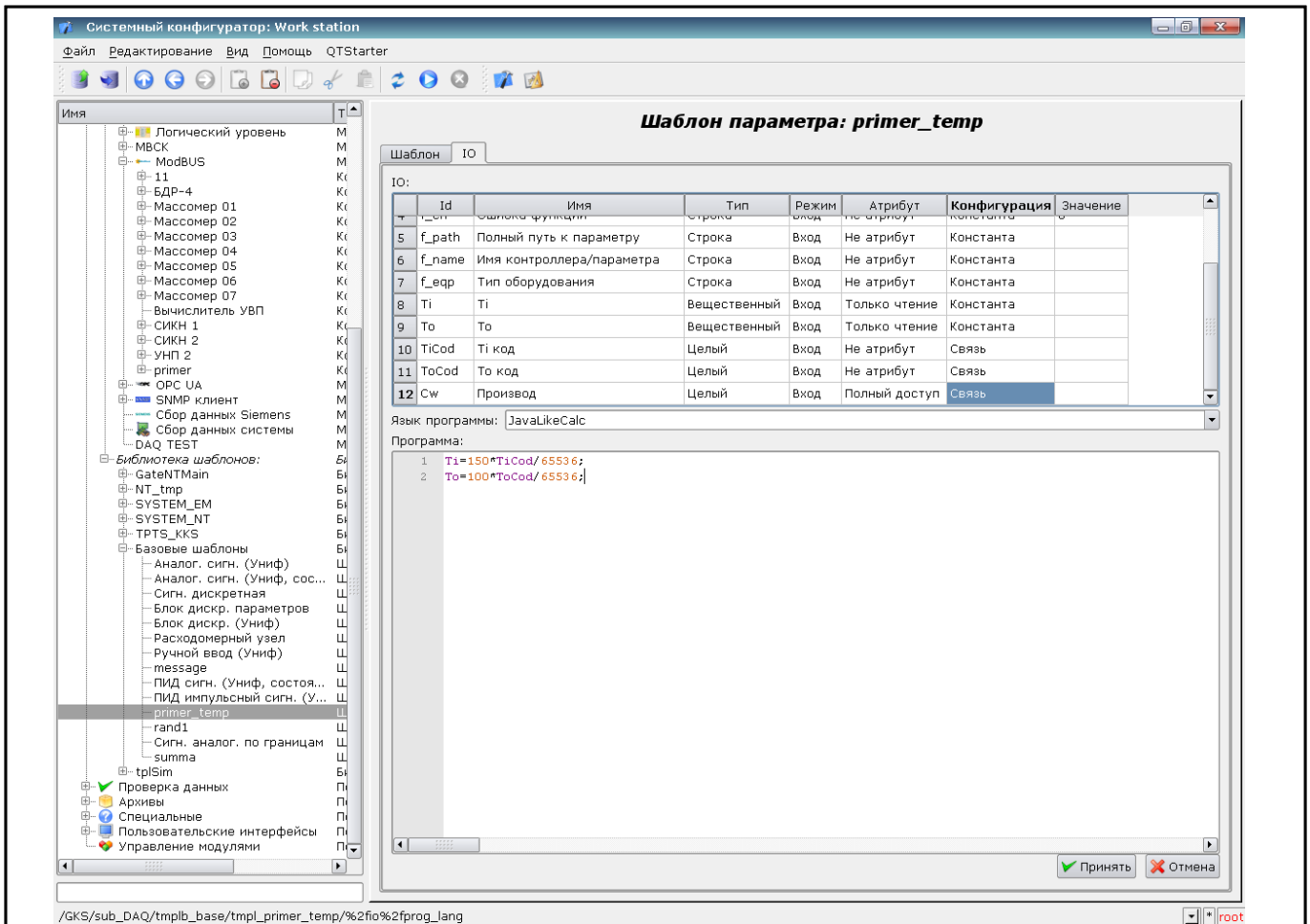


Рисунок 61

Полученный шаблон необходимо сохранить и сделать шаблон доступным, установив флажок напротив соответствующего поля. Доступные шаблоны могут подключаться к параметрам контроллеров сбора данных, а параметры будут выполнять вычисления по этому шаблону. В поле "Использовано" отображается число объектов, которые используют данный шаблон для вычисления образа параметра.

Далее необходимо создать контроллер на «Логическом уровне» и связать его параметры с параметрами шаблона (последовательный выбор типа источника, элемента выбора, параметра, атрибута – алгоритм описан в 1.3).

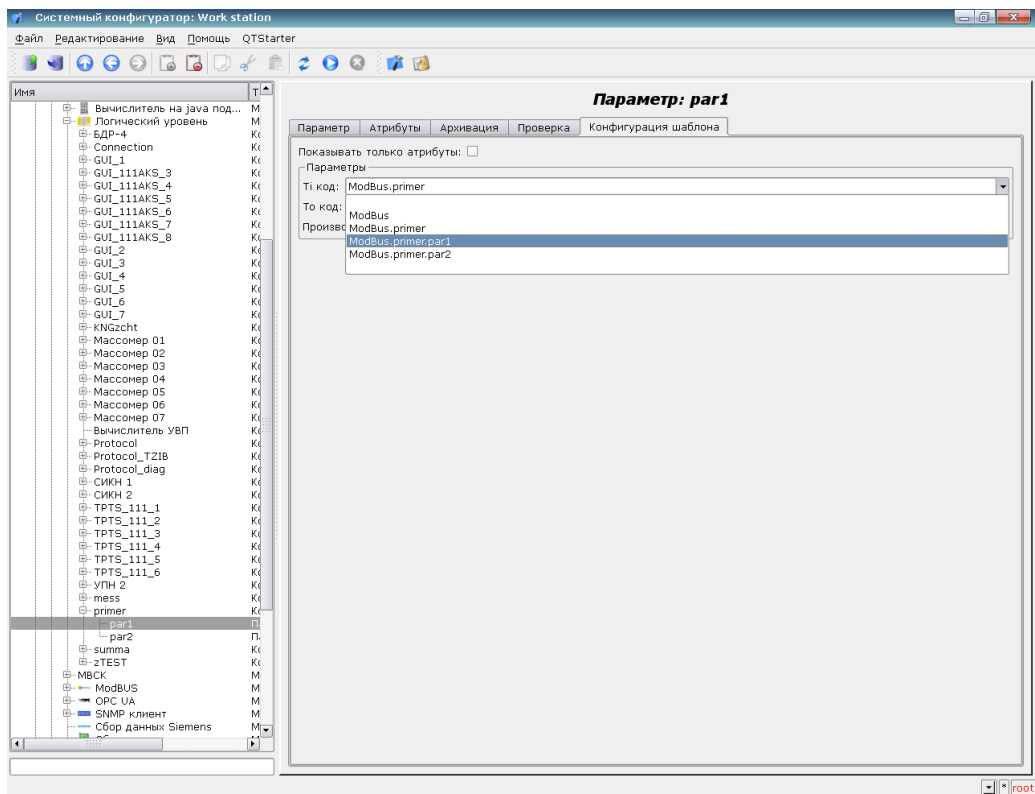


Рисунок 62

После сохранения сделанных изменений, включения и запуска контроллера в поле «Статус» раздела «Состояние» отразится результат обработки данных (рисунок 63), а у параметров контроллера отразятся вычисленные значения атрибутов (рисунок 64).

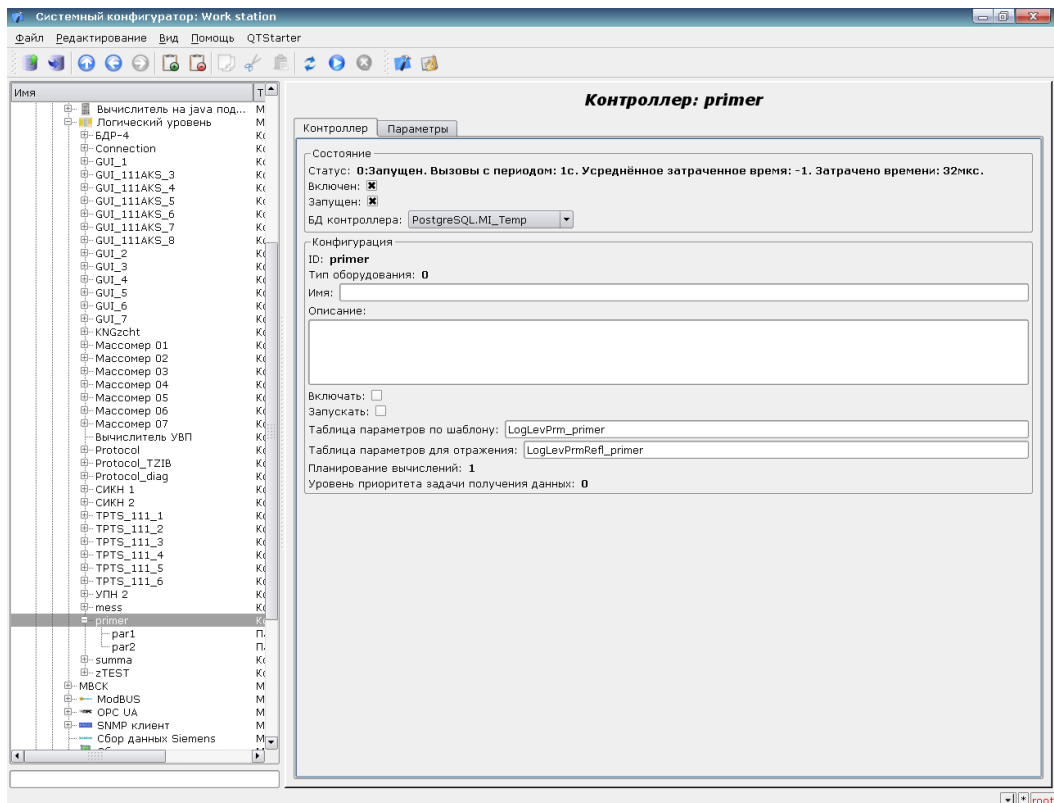


Рисунок 63



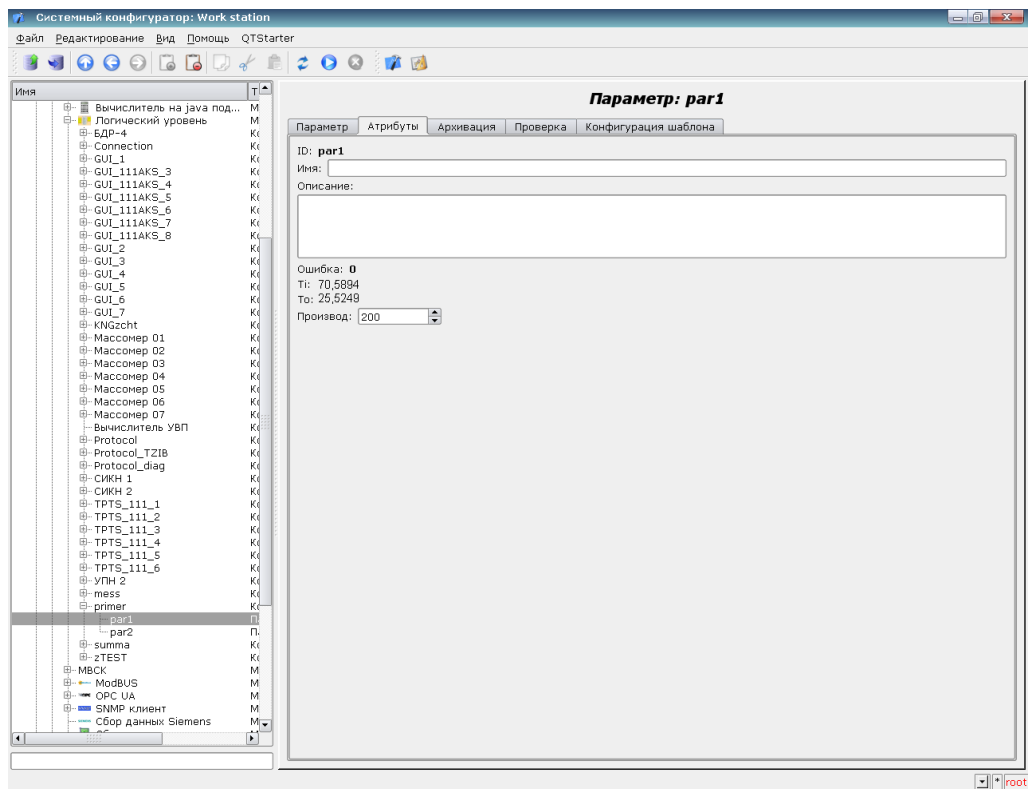


Рисунок 64

На этом конфигурация обработки данных считается законченной.

## 2.2 Модуль шлюз источников данных DAQGate

### 2.2.1 Назначение модуля

Основной функцией данного модуля является отражение данных подсистемы «Сбор данных» с серверов на АРМ. В своей работе модуль использует собственный протокол системы (Self System).

Модулем реализуются следующие функции:

- отражение структуры параметров подсистемы «Сбор данных» удаленного сервера на АРМ. Структура периодически при работе синхронизируется;
- доступ к текущим значениям атрибутов параметров и возможность их модификации. Значения атрибутов параметров обновляются с периодичностью исполнения локального контроллера на АРМ. Запросы на модификацию атрибутов транслируются на сервер;
- предоставление реализации механизма вертикального резервирования, а именно возможность отражения данных с нескольких серверов одного уровня на АРМ.

Использование схемы отражения данных представлено на рисунке 65.

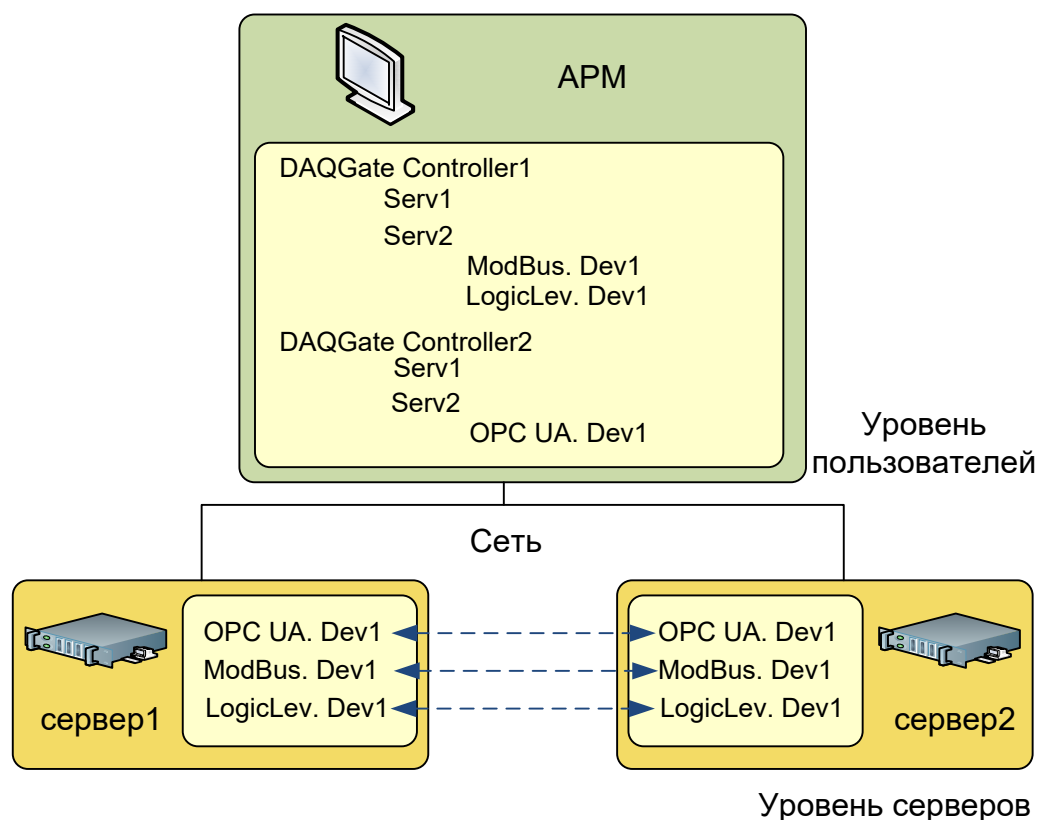


Рисунок 65

### 2.2.2 Конфигурирование передачи данных

Для подключения контроллера «Сбора данных» необходимо в левой части окна системного конфигуратора ПП «СКАДА А-СОФТ» раскрыть вкладку «Сбор данных» нажатием левой клавиши «мыши» на ней. В появившемся меню раскрыть вкладку «Модуль» и выбрать вкладку «Шлюз источников данных» (рисунок 66).

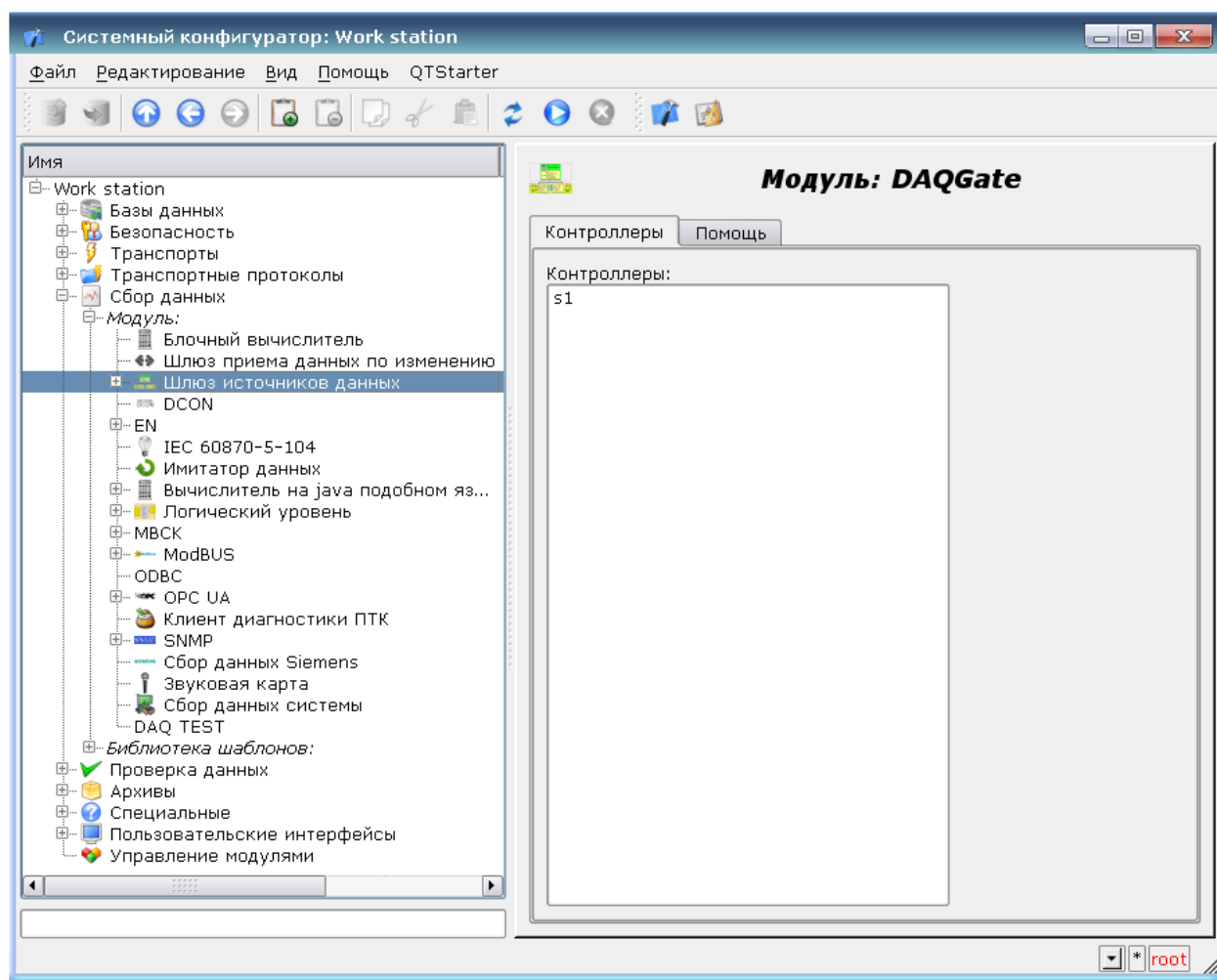


Рисунок 66

В правой части окна конфигуратора во вкладке «Контроллеры» нажать правую клавишу «мыши» и выбрать «Добавить» (рисунок 67). В диалоговом окне «Установка имени элемента» (рисунок 68) задать идентификатор (ID) шлюза сопряжения и его имя, например, «Gate». Нажать кнопку «ОК». В строке меню «Шлюз источников данных» появится вкладка «Gate».

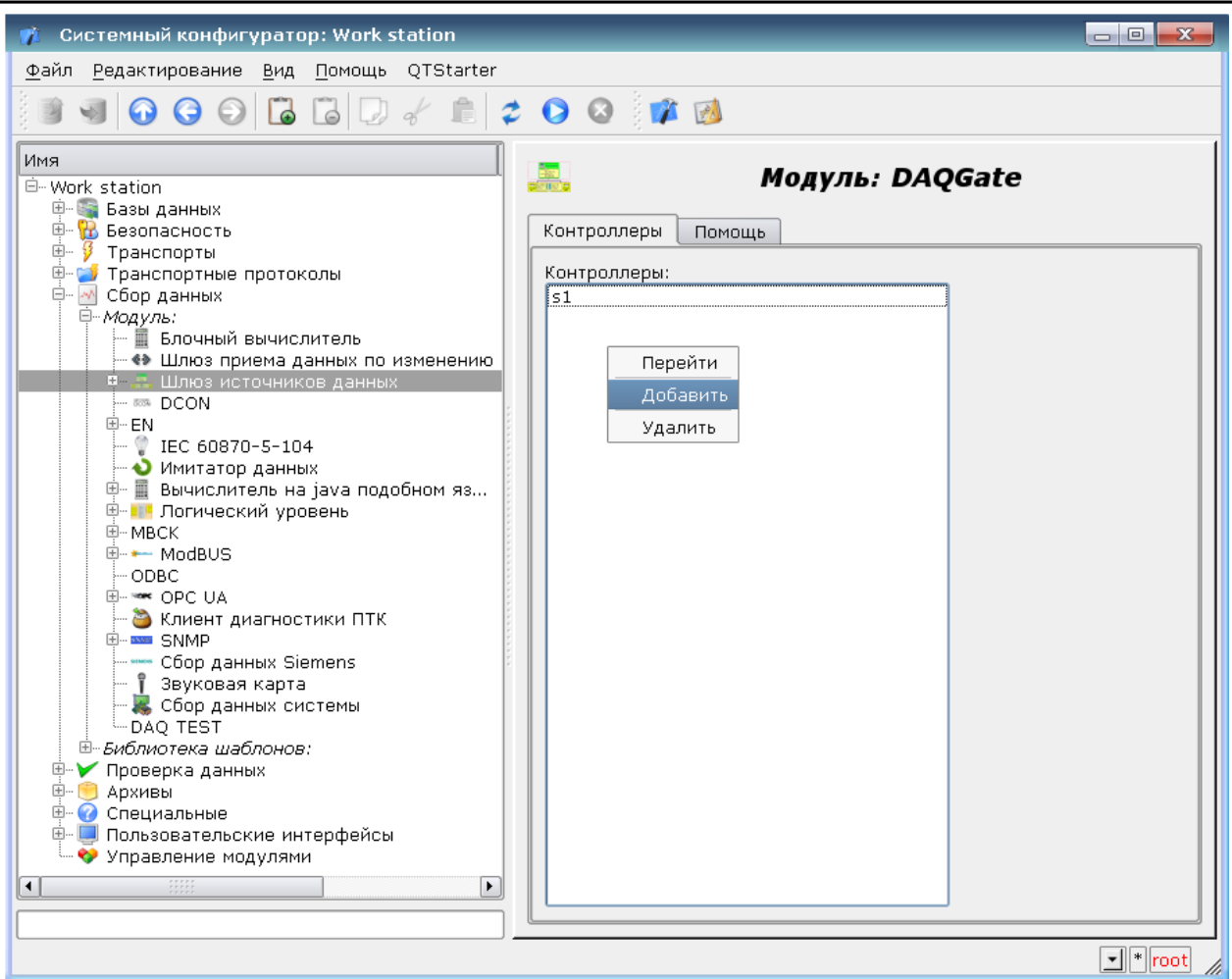


Рисунок 67

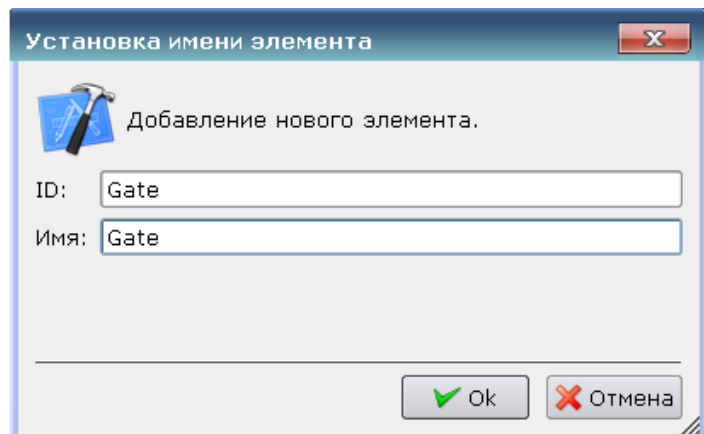


Рисунок 68

В левой части окна конфигуризатора двойным нажатием левой клавиши «мыши» выбрать вкладку «Gate», при этом главное окно примет вид, изображенный на рисунке 69.

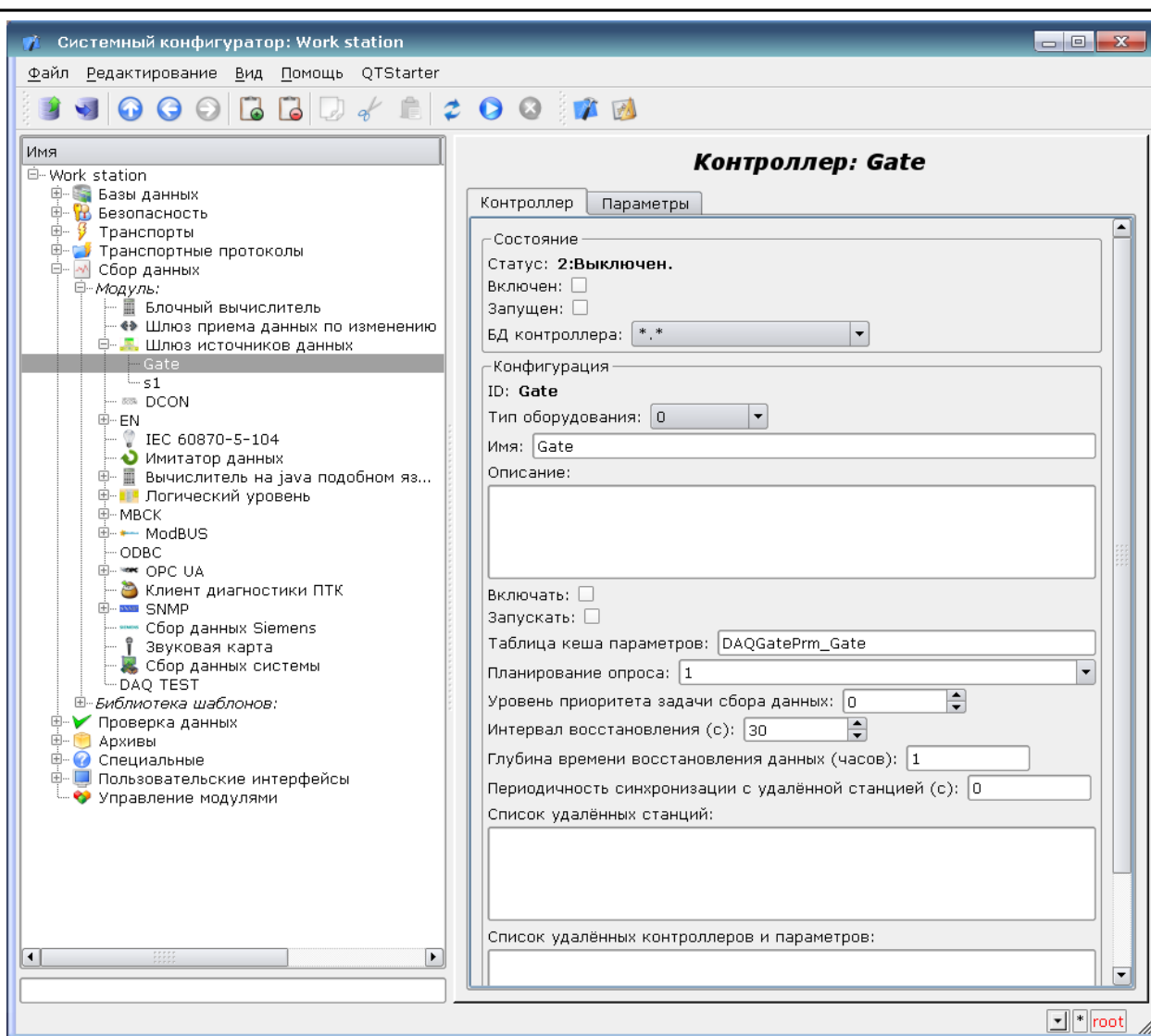


Рисунок 69

В окне «Контроллер:Gate» во вкладке «Контроллеры» в поле ввода «Описание» ввести информацию о подключаемом шлюзе.

Левой клавишей «мыши» установить отметки «Включать» и «Запускать». Это обеспечит автоматический запуск контроллера при следующем старте подсистемы обработки данных.

В поле ввода «Интервал восстановления» задать время восстановления соединения с подсистемой интеграции - 3 с.

В поле «Список удаленных станций» ввести имя подсистемы интеграции.

В поле «Список удаленных контроллеров и параметров» указать контроллеры системы интеграции, с которых будут передаваться данные. Имя контроллера должно совпадать с именем контроллера в шлюзе (рисунок 70).

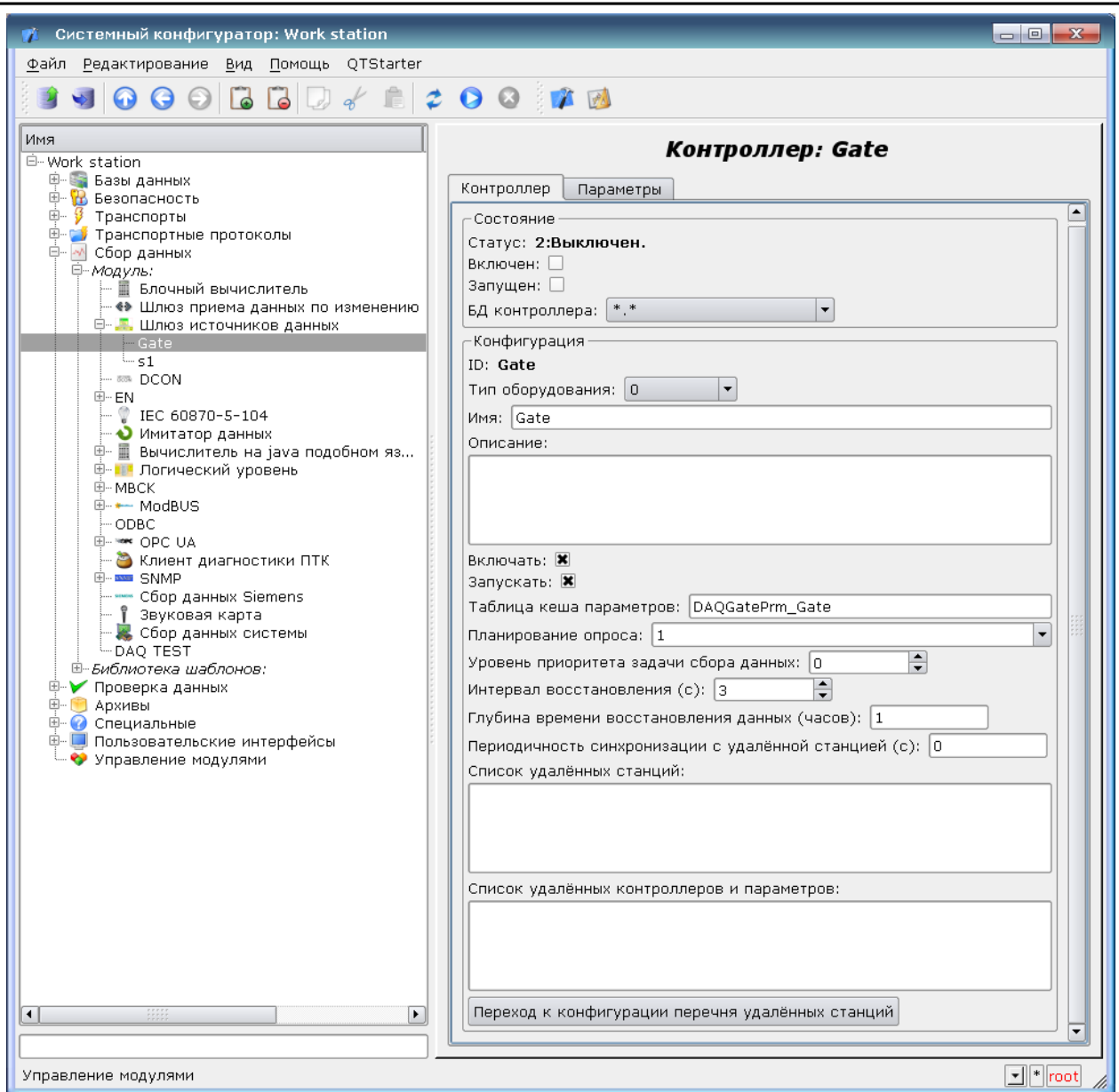


Рисунок 70

Левой клавишей «мыши» выбрать кнопку «Переход к конфигурации перечня удаленных станций». Основное окно программы примет вид, изображенный на рисунке 71.

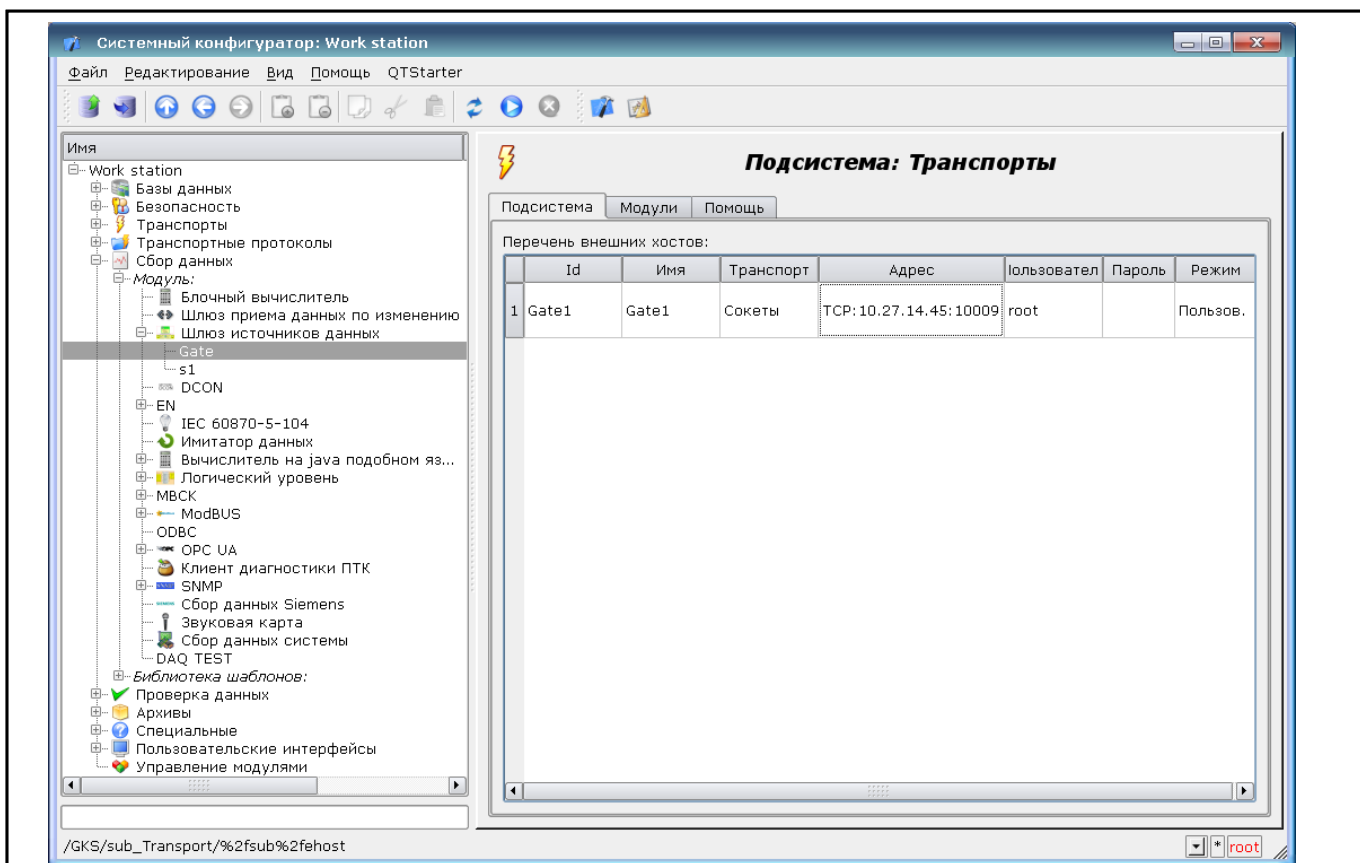


Рисунок 71

В правой части окна системного конфигуриатора во вкладке «Подсистема» в столбце «Адрес» задать IP-адрес и порт, по которому будет осуществляться связь со шлюзами.

Сохранить конфигурацию в базе данных.

При следующем запуске подсистемы связь с настроенной подсистемой интеграции будет установлена автоматически.

## 2.3 Модуль источника данных TANGO

Окно модуля источника данных TANGO кроме стандартных вкладок «Контроллеры» и «Помощь» содержит вкладку «Загрузка данных» (рисунок 72), с помощью которой можно провести проверку значений атрибутов либо результата выполнения команды для используемого устройства TANGO. Для этого необходимо последовательно заполнить поля в указанном окне и нажать на кнопку «Выполнить», после чего в нижней части окна отобразится результат выполнения проверки. Если в поле «Действие для подтверждения» указано «Присвоение значения атрибуту», то значение будет записано в атрибут, указанный в поле «Подтверждающий объект».

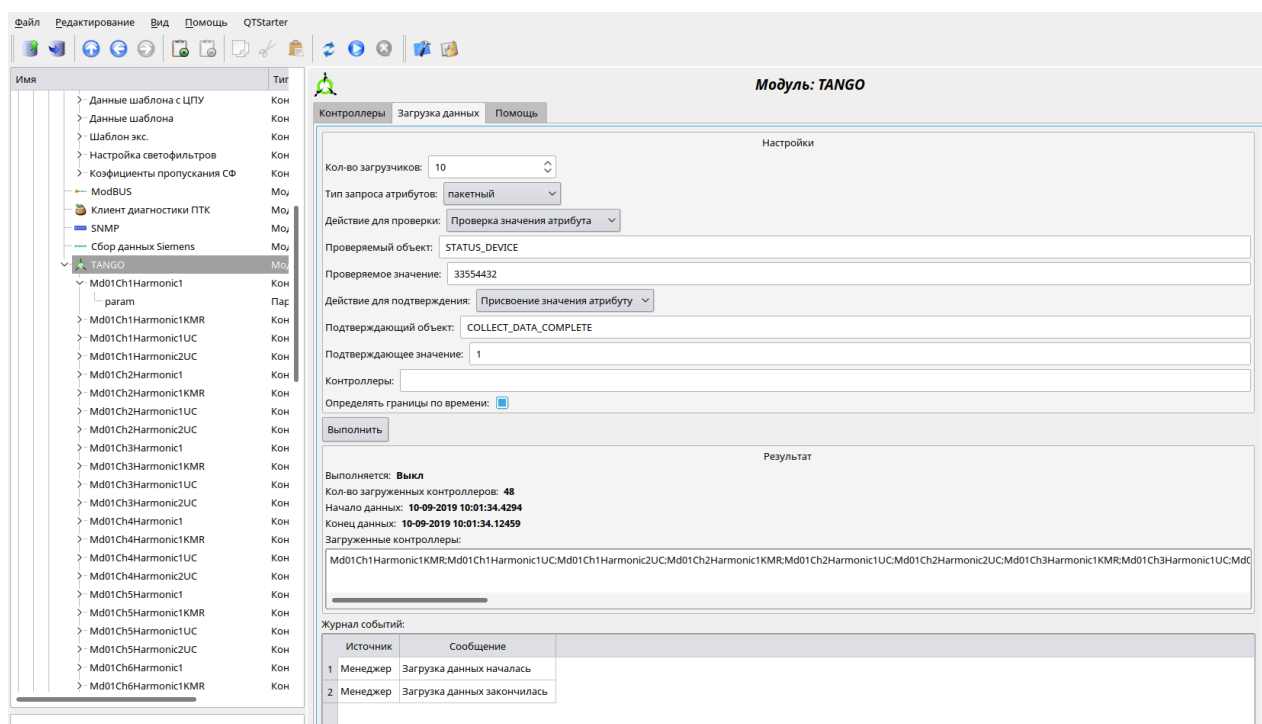


Рисунок72

Для добавления источника данных TANGO, создаётся и конфигурируется контроллер в SCADA-системе. Пример вкладки конфигурации контроллера данного типа изображен на рисунке 73.



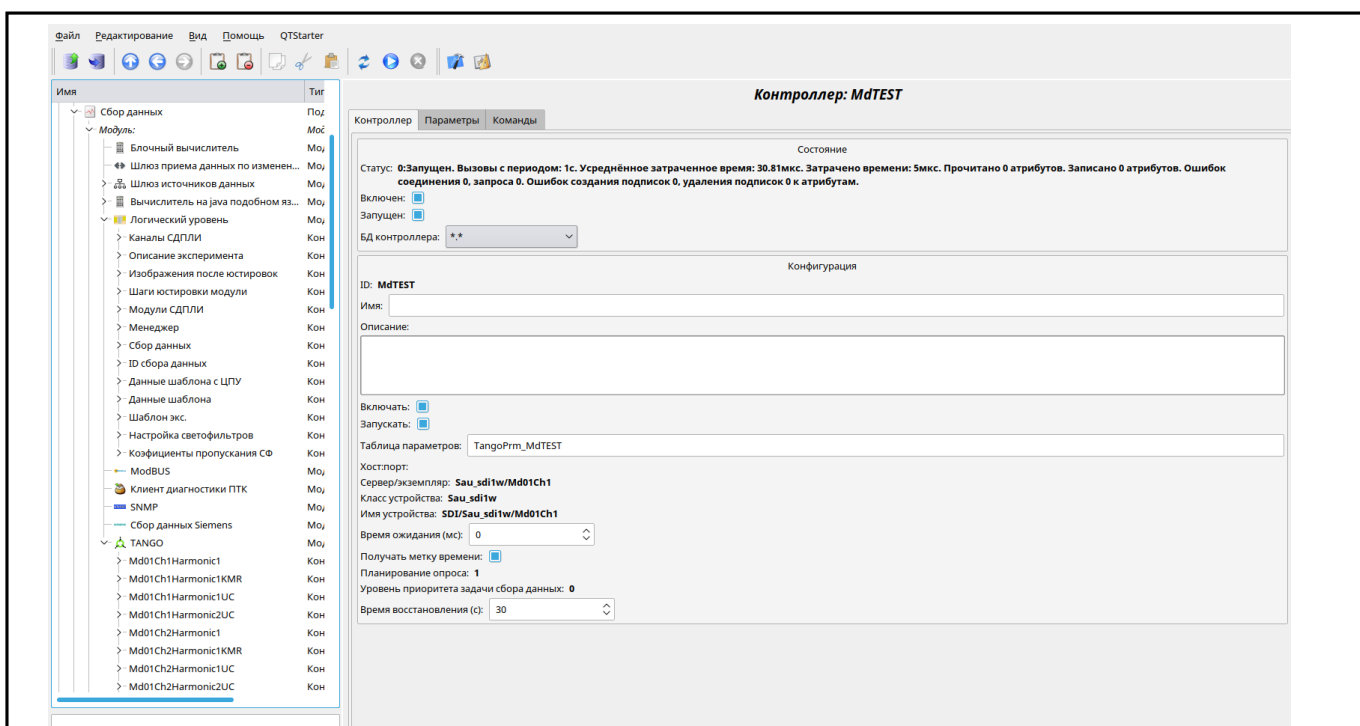


Рисунок 73

С помощью этой вкладки можно установить:

- состояние контроллера, а именно: Статус, "Включен", Запущен" и имя БД, содержащей конфигурацию;
- идентификатор, имя и описание контроллера;
- состояние, в которое переводить контроллер при загрузке: "Включать" и "Запускать";
- имя таблицы для хранения конфигурации параметров контроллера;
- Хост:порт – адрес TANGO-хоста в формате IP:порт. Если поле пустое, то по умолчанию устанавливается значение из переменной окружения TANGO\_HOST;
- Сервер/экземпляр - имя сервера устройства, экземпляра класса, класса устройств и имя устройства. Существует возможность, как задать вручную данные конфигурации, так и выбрать из выпадающего списка, при этом устройства должны быть созданы и включены заранее, а в переменной окружения TANGO\_HOST должен быть определен адрес базы данных TANGO;
- время ожидания – время ожидания ответа от устройства в мс;
- получать метку времени;
- планирование опроса;
- уровень приоритета задачи сбора данных;

- время восстановления соединения в секундах. Указывает промежуток времени, по истечению которого осуществляется повторная попытка запроса к ранее недоступному устройству.
- Конфигурационным полем параметра данного модуля (рисунок 74) является таблица атрибутов. При включении контроллера список возможных для обработки атрибутов загружается из устройства TANGO. Четыре первых колонки таблицы содержат информацию об атрибуте: имя атрибута, доступ к атрибуту, тип данных и формат данных. Эти конфигурации нельзя изменять, они необходимы модулю TANGO для корректной обработки данных атрибутов. Колонки псевдоним и режим позволяют выбрать идентификатор атрибута и режим взаимодействия с устройством: “Отключен” – устройство не опрашивается, “Синхронный” - происходит периодический опрос устройства для получения данных с заданным периодом сбора, ”По изменению” - само устройство инициирует передачу данных при выполнении определенных условия заданных при его конфигурации; ”По требованию” – Скада сама инициирует считывание данных; ” Загрузка” – считывание данных только на момент сбора данных.

Список выбранных атрибутов считывается только в момент включения параметра и не изменяется до его выключения.

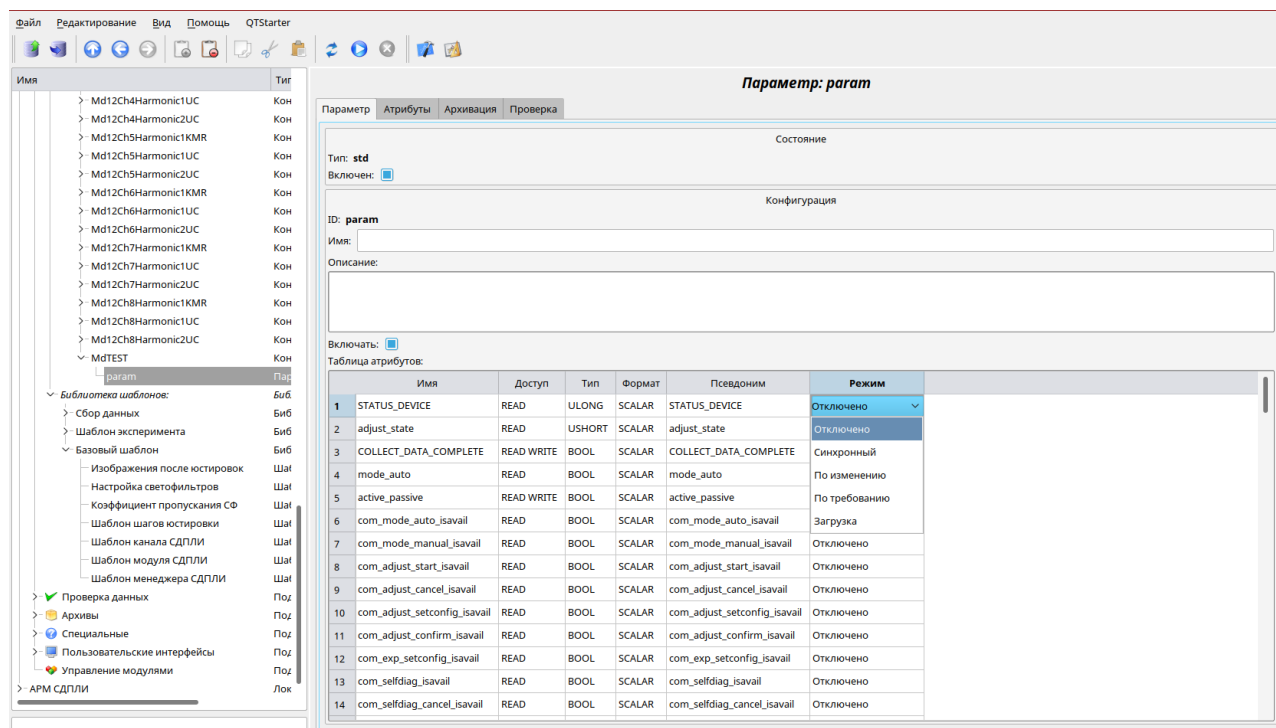


Рисунок 74

В соответствии с выбранными в таблице атрибутами выполняется создание атрибутов параметра (рисунок 75).

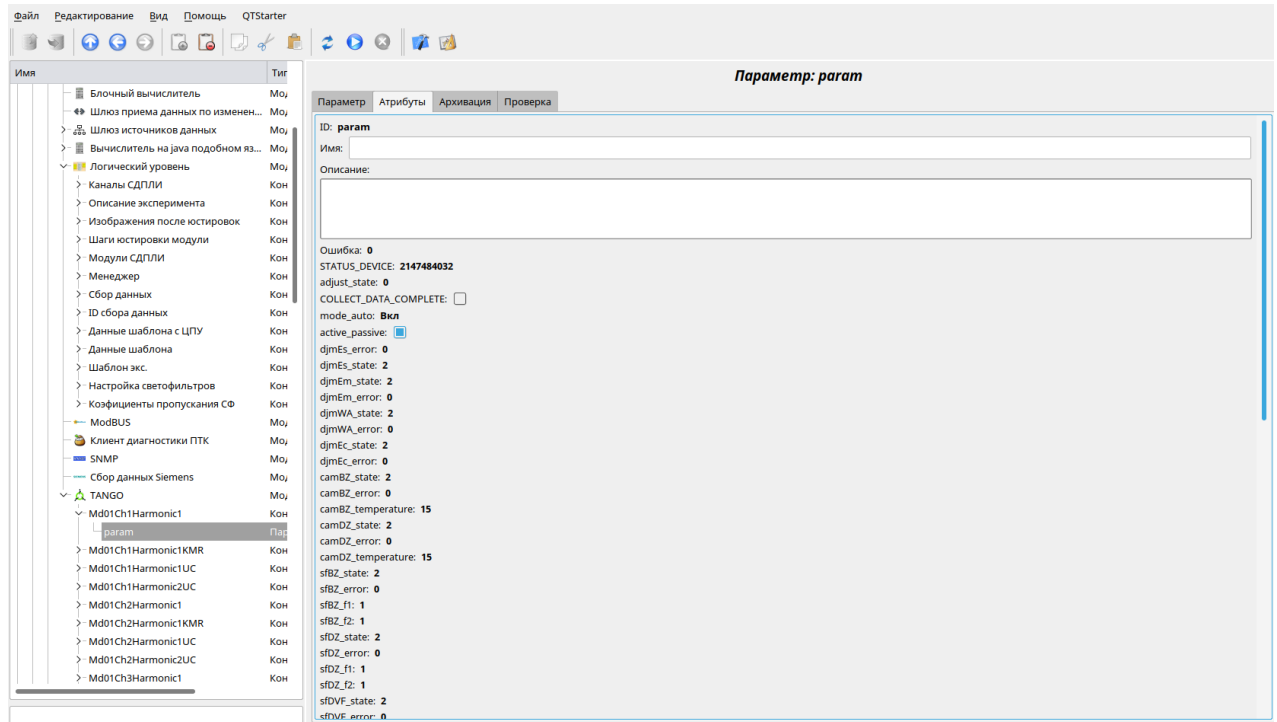


Рисунок 75

## 2.4 Модуль источника данных PCI

Модуль PCI предназначен для взаимодействия ПП «СКАДА А-СОФТ» с платой ввода/вывода PCIe-1760. Модуль PCI обеспечивает чтение данных по входным дискретным каналам и чтение/запись данных по выходным дискретным каналам платы PCIe-1760 (всего – 8 входных и 8 выходных каналов).

## 2.5 Модуль источника данных SNMP

Данный модуль предоставляет возможность собирать информацию и вносить изменения у различных устройств по протоколу SNMP.

SNMP (англ. SimpleNetworkManagementProtocol — простой протокол сетевого управления) — стандартный интернет-протокол для управления устройствами в IP-сетях на основе архитектур TCP/UDP. К поддерживающим SNMP устройствам относятся маршрутизаторы, коммутаторы, серверы, рабочие станции, принтеры, модемные стойки и другие. Протокол используется для контроля подключенных к сети устройств на предмет условий, которые требуют внимания администратора. Он состоит из набора стандартов для сетевого управления, включая протокол прикладного уровня, схему БД и набор объектов данных.

SNMP предоставляет данные для управления в виде переменных, описывающих конфигурацию управляемой системы. Эти переменные могут быть запрошены (а иногда и заданы) управляющими приложениями.

Доступные через SNMP переменные организованы в иерархии. Эти иерархии описываются базами управляющей информации (базы MIB, от англ. ManagementInformationBase).

Базы MIB используют иерархическое пространство имен, содержащее идентификаторы объектов (OID-ы). Каждый OID состоит из двух частей: текстового имени и SNMP адреса в цифровом виде. Базы MIB выполняют вспомогательную роль по переводу имени объекта из текстового формата в формат SNMP (цифровой). Так как структура объектов на устройствах разных производителей не совпадает, без базы MIB практически невозможно определить цифровые SNMP адреса нужных объектов.

На рисунке 77 представлено окно настройки контроллера подсистемы Сбор данных по протоколу SNMP.

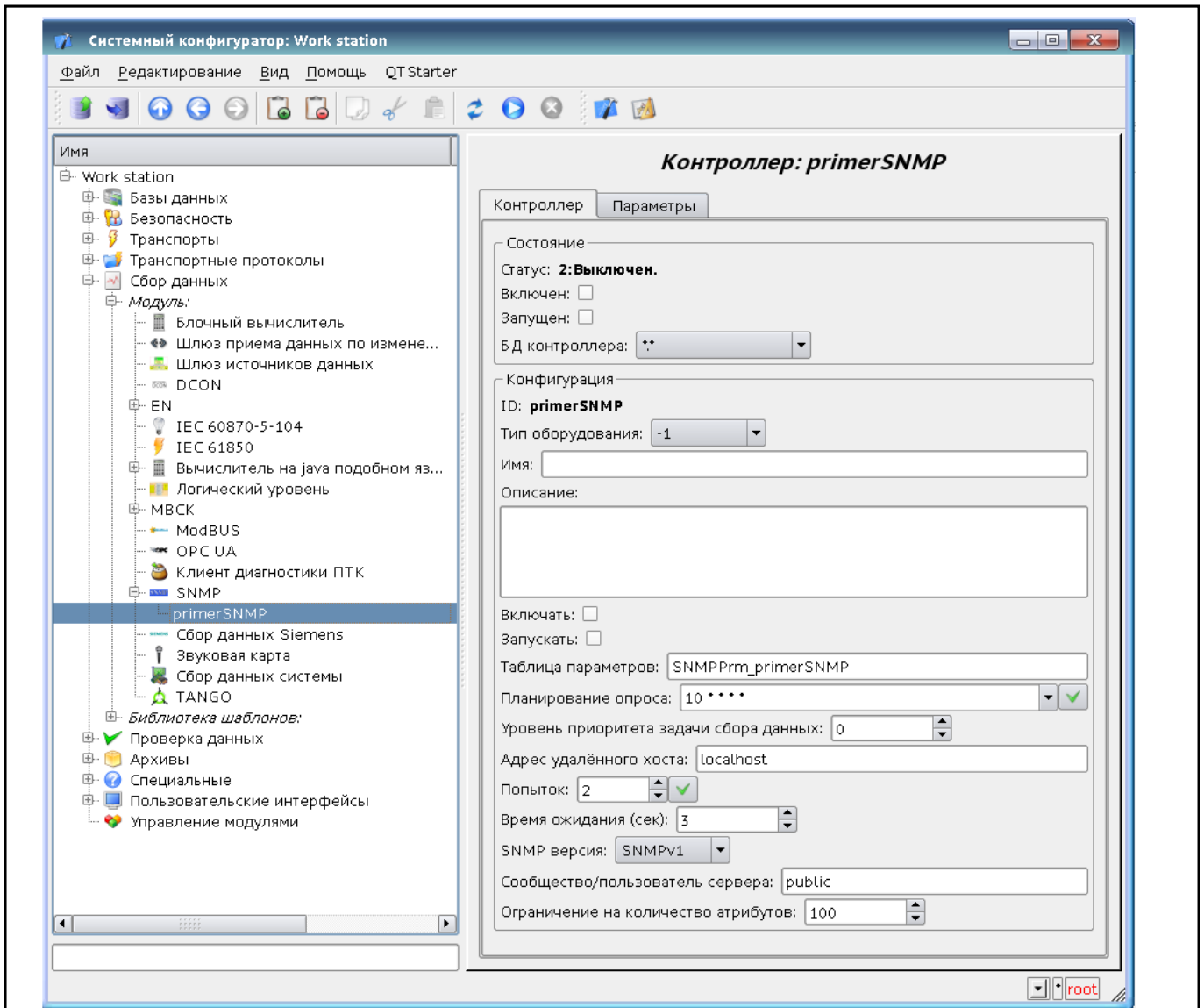


Рисунок 76

С помощью вкладки «Контроллер» можно установить:

- состояние контроллера, а именно: Статус, "Включен", Запущен" и имя БД, содержащей конфигурацию;
- идентификатор, имя и описание контроллера;
- состояние, в которое переводить контроллер при загрузке: "Включать" и "Запускать";
- имя таблицы для хранения конфигурации параметров контроллера;
- планирование опроса;
- уровень приоритета задачи сбора данных (-1 – 99);
- адрес удаленного хоста – SNMP хост агента в IP адресе или доменном имени хоста, также можно установить порт «localhost:161»;
- количество попыток установить соединение;

- время ожидания (сек);
- SNMP версия – SNMPv1, SNMPv2c, SNMPv2u, SNMPv3. Для версии SNMPv3 становятся активными поля настройки уровня безопасности и авторизации/приватности;
- сообщество/пользователь сервера;
- ограничение на количество атрибутов (по умолчанию 100).

Для опроса параметров устройства по протоколу SNMP необходимо добавить контроллер модуля SNMP, соответствующий данному устройству. В поле «Адрес удаленного хоста» указать IP адрес опрашиваемого устройства. В атрибуте «SNMP версия» указать версию протокола SNMP.

На рисунке 78 приведен пример настройки контроллера модуля SNMP для ИБП.

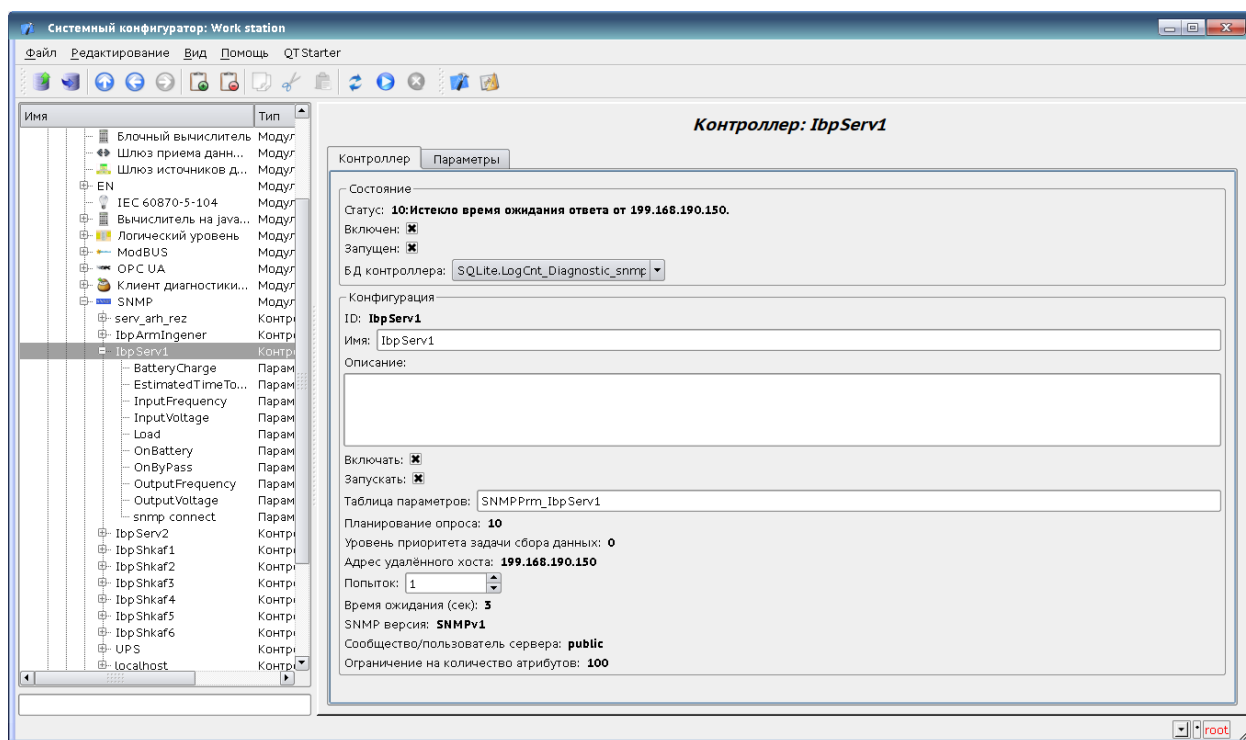


Рисунок 77

В созданном контроллере необходимо добавить требуемое число параметров. Основным атрибутом параметра является «Список OID (разделение след. строкой)». В нем необходимо указать OID опрашиваемого параметра. На рисунке 79 приведен пример параметра модуля SNMP для заряда батареи ИБП.

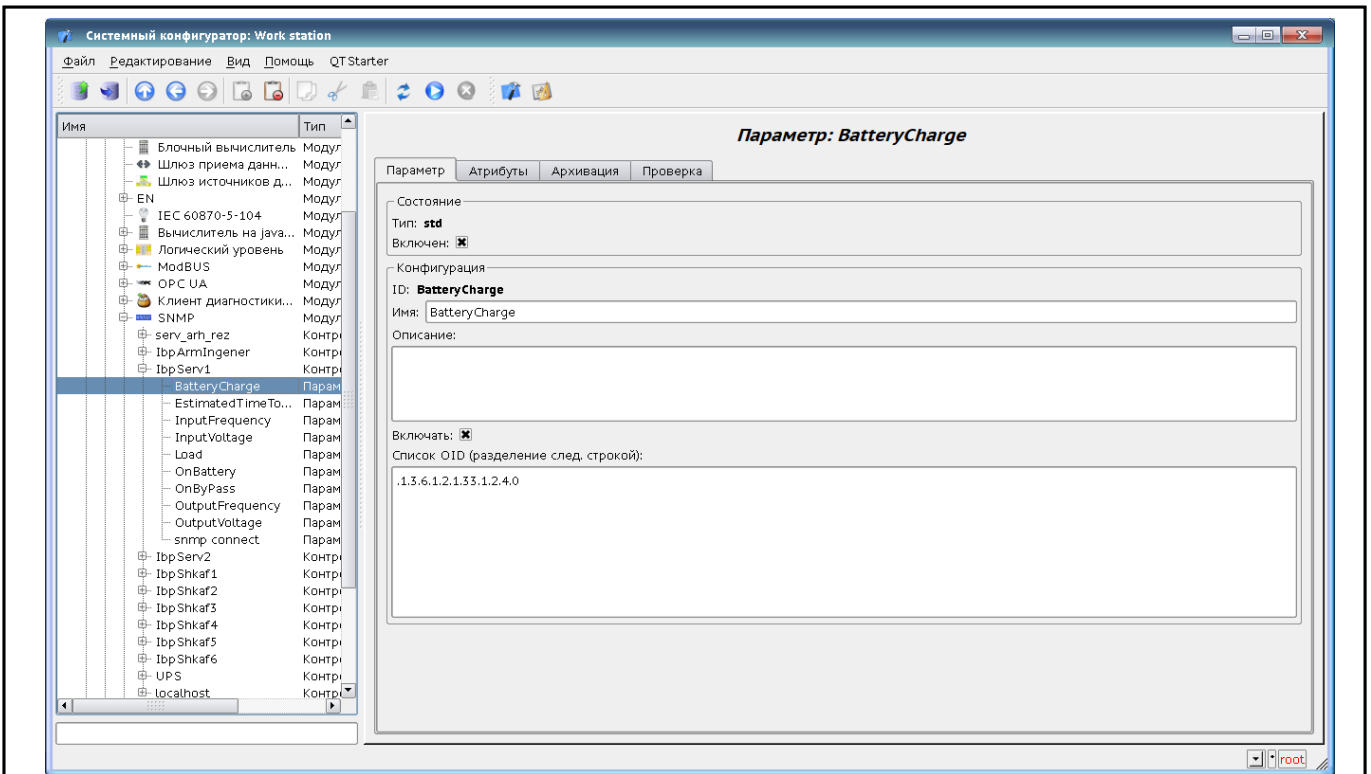


Рисунок 78

## 2.6 Модуль источника данных IEC 61850

Протокол IEC 61850 является стандартом «Сети и системы связи на подстанциях», описывающий форматы потоков данных, виды информации, правила описания элементов энергообъекта и свод правил для организации событийного протокола передачи данных.

Основным требованием к системе сбора данных в стандарте является обеспечение способности микропроцессорных электронных устройств к обмену технологическими и другими данными. IEC 61850 является объектно-ориентированным протоколом, нацеленный на автоматизацию подстанций, и значительно расширяет возможности предшествующих стандартов. Он определяет требования к описанию электрических систем на всех уровнях, начиная от уровня системы в целом, заканчивая конфигурацией отдельного терминала релейной защиты и автоматики (РЗА).

Согласно IEC 61850 устройства РЗА объединены шиной, по которой сами устройства обмениваются данными между собой и передают эти данные на верхний уровень. Такая архитектура удобна тем, что применение технологической шины значительно уменьшает количество медных проводов, что упрощает настройку, проектирование и эксплуатацию системы.

Данные от терминалов релейной защиты по станционной шине могут передаваться на верхний уровень оператору, кроме того, у контролирующих органов, имеющих соответствующий уровень доступа, есть возможность получать оперативные данные с любой подстанции и с любого терминала РЗА. Эта информация позволяет контролировать деятельность подчиненных служб, что повышает надежность энергетических объектов в целом.

Настройка связи осуществляется путем конфигурирования контроллера данных в модуле IEC 61850 подсистемы «Сбор данных». Добавление контроллера модуля IEC осуществляется выбором пункта «Добавить» при нажатии правой кнопки мыши и указания ID и имени. Окно конфигуратора представлено на рисунке 80 и содержит следующие поля для настройки:

- состояние контроллера, а именно: Статус, "Включен", Запущен" и имя БД, содержащей конфигурацию;
- идентификатор, имя и описание контроллера;
- состояние, в которое переводить контроллер при загрузке: "Включать" и "Запускать";
- имя таблицы для хранения конфигурации параметров контроллера;



- планирование опроса в формы секундной периодичности или в форме стандарта CRON;
- уровень приоритета задачи сбора данных (-1 – 99);
- IP адрес – источника данных;
- TCP порт соединения.

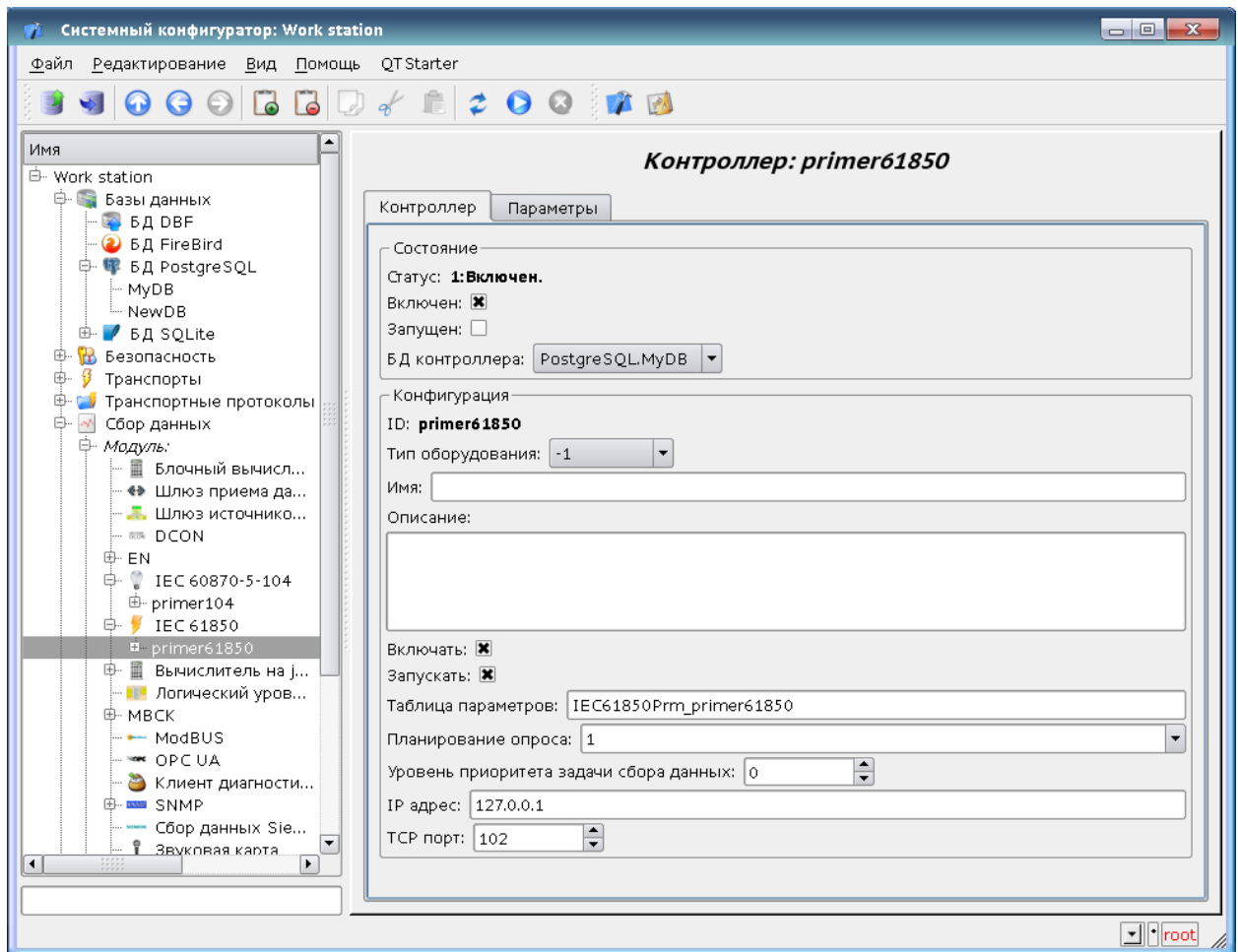


Рисунок 79

Конфигурирование параметров контроллера модуля IEC осуществляется в окне «Параметр» после их добавления (рисунок 81).

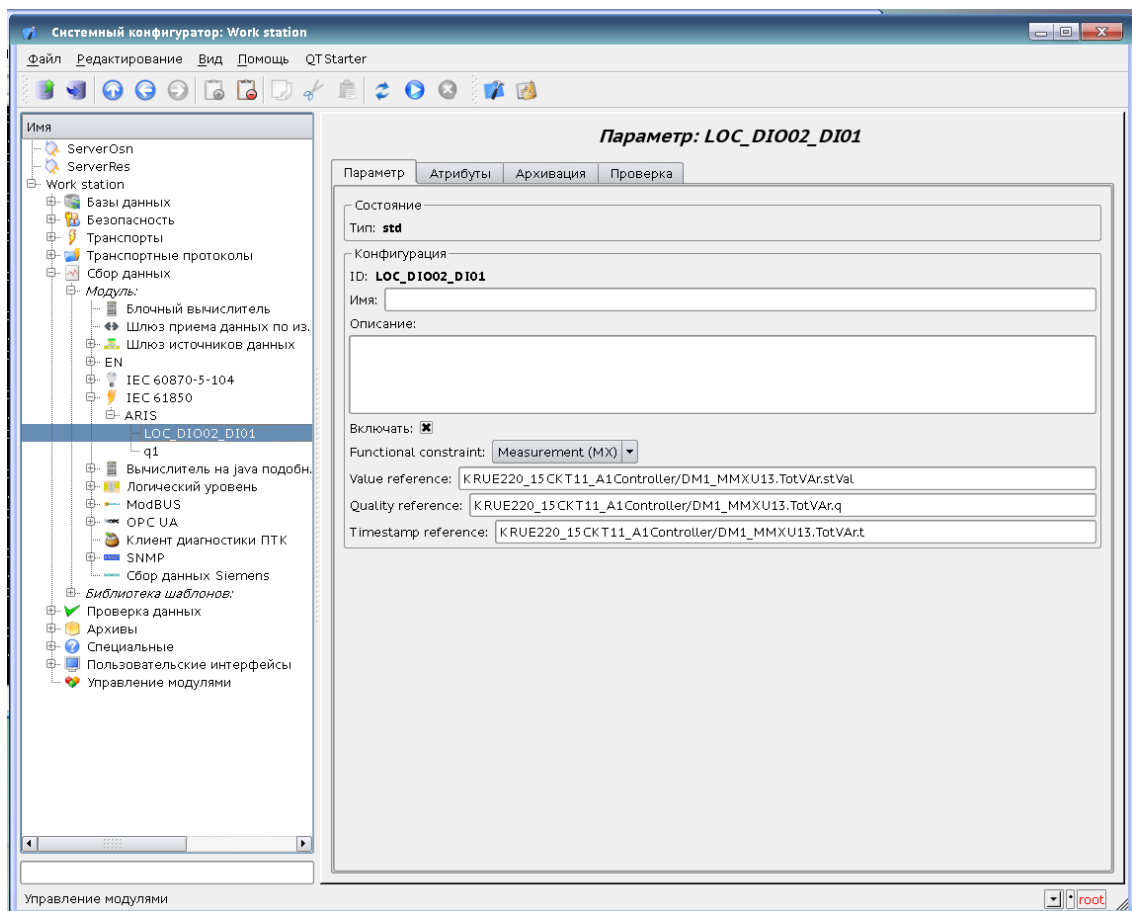



Рисунок 80

Окно содержит следующие индивидуальные поля для редактирования:

- функциональные ограничения (Functional constraint) - принимает значения:
  - ST – статус;
  - MX – измерение;
  - SV – подстановочные значения;
  - DC – описание;
  - EX – свойства расширенного определения;
  - BL – блокировка значения;
  - CO – контроль;
- Value reference – ссылка на значение контроллера (в формате <контроллер>/<параметр>.<атрибут>);
- Quality reference – ссылка на качество контроллера (в формате <контроллер>/<параметр>.<атрибут>);
- Timestamp reference – отметка времени.

Правильно указав ссылки на значения, получаемые с контроллеров по данному протоколу, можно проверить получение данных, нажав на пиктограмму . Окно редактирования параметра примет вид, представленный на рисунке 82. При этом во вкладке атрибуты можно просмотреть полученные с контроллера значения (рисунок 83).

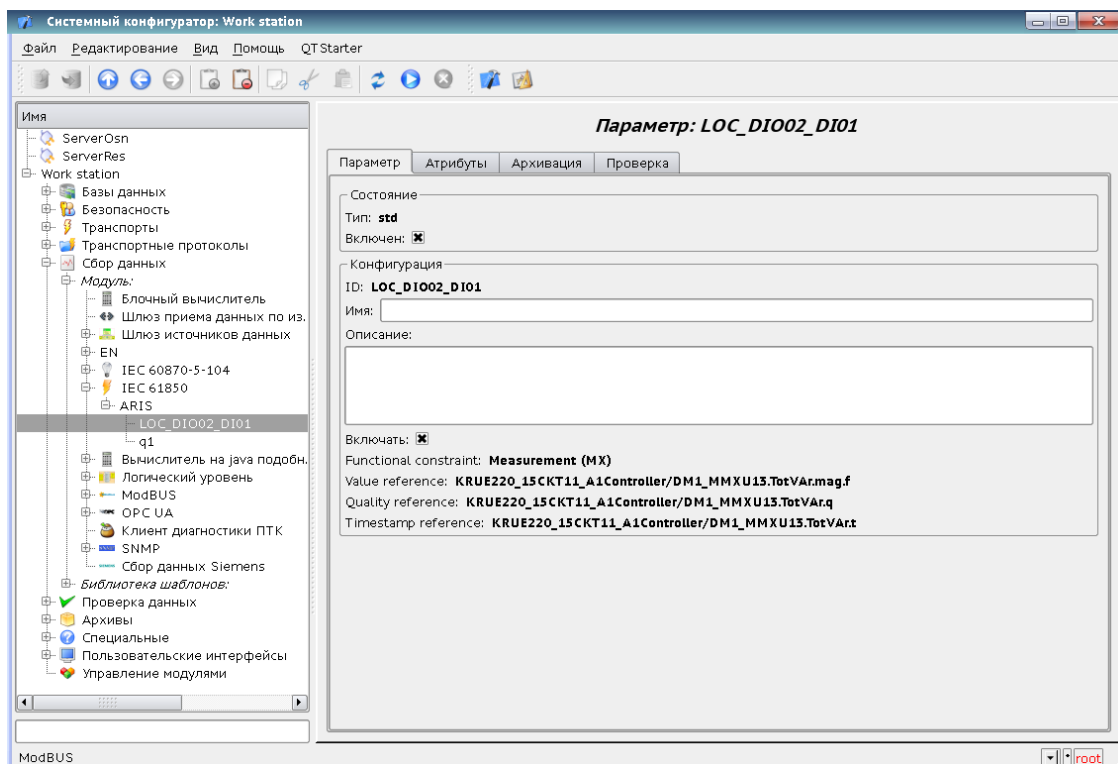


Рисунок 81

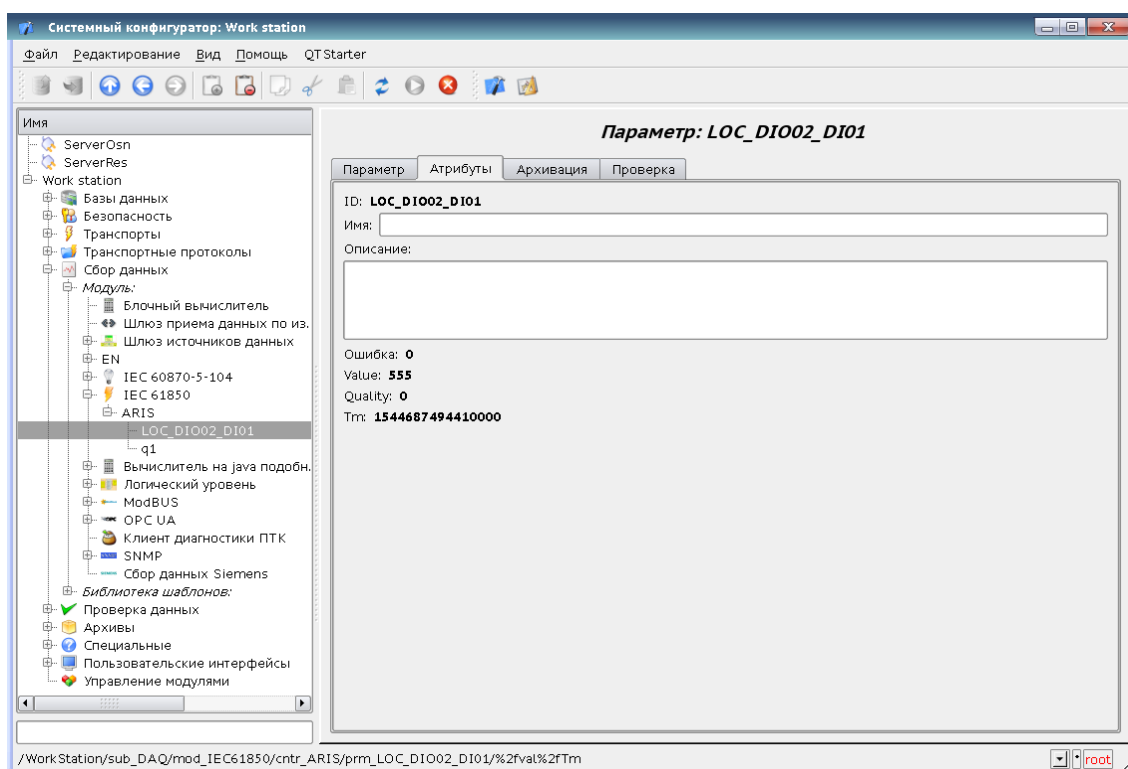


Рисунок 82

## 2.7 Модуль источника данных IEC 60870-5-104

Протоколы серии 60870-5 основаны на трехуровневой модели называемой «Архитектура повышенной производительности» (ERA), являющейся упрощенным вариантом семиуровневой модели ИСО.

Протокол IEC 60870-5-104 распространяется на устройства и системы телемеханики с передачей данных последовательными двоичными кодами для контроля и управления территориально распределенными процессами. Протокол регламентирует использование сетевого доступа по протоколу TCP/IP. Обеспечивает достаточно высокую функциональность при решении задач телеуправления, телесигнализации и телеизмерений, интеграции данных устройств в системы управления.

В основу протоколов положен обмен таблицами сигналов, причем типы данных, которыми осуществляется обмен, жестко фиксированы. Протокол не предусматривает возможность передачи сигналов реального времени.

Конфигурирование сбора данных по протоколу IEC 60870-5-104 осуществляется в окне конфигурирования контроллера (рисунок 84).

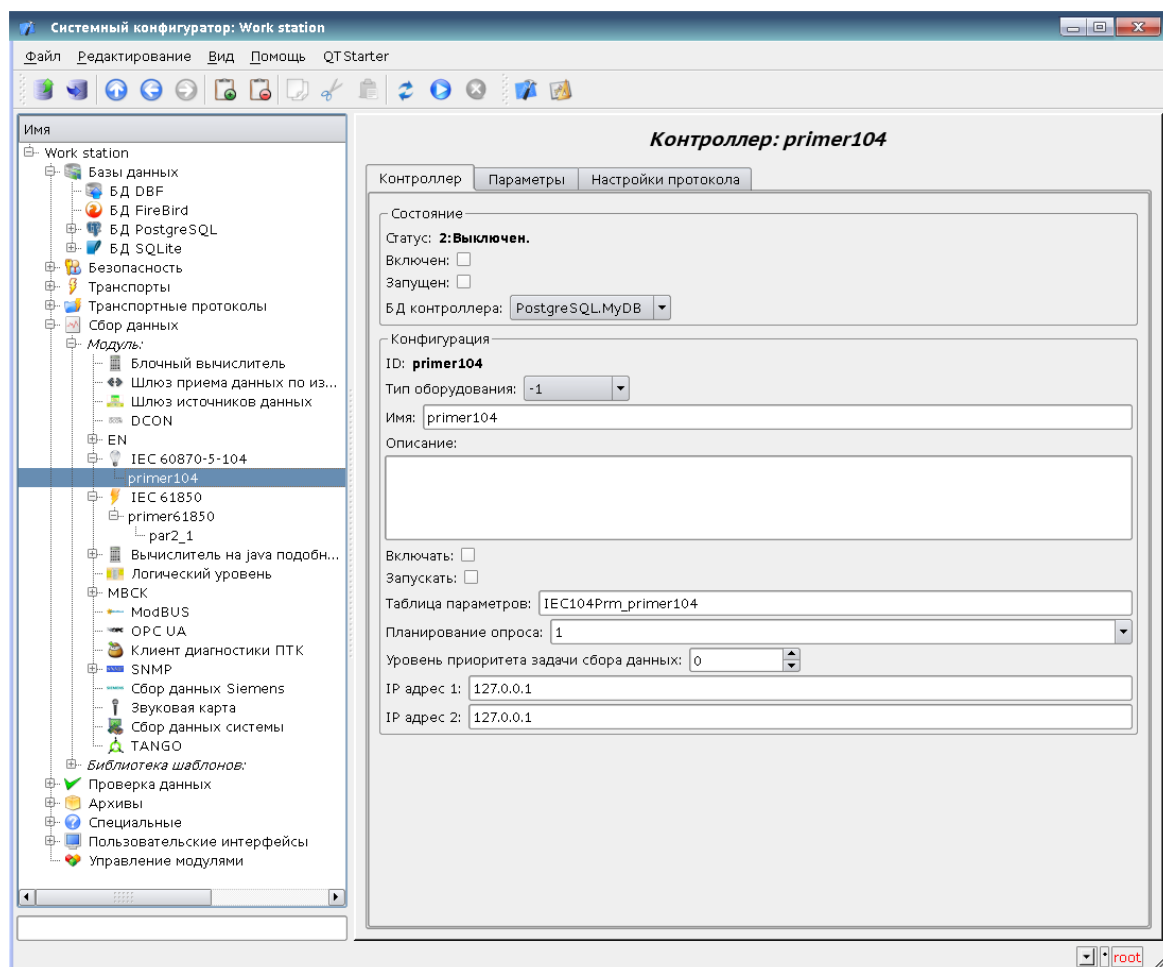


Рисунок 83

Индивидуальными настройками контроллера данного модуля являются поля настройки IP адресов, а также настройки протокола (рисунок 85).

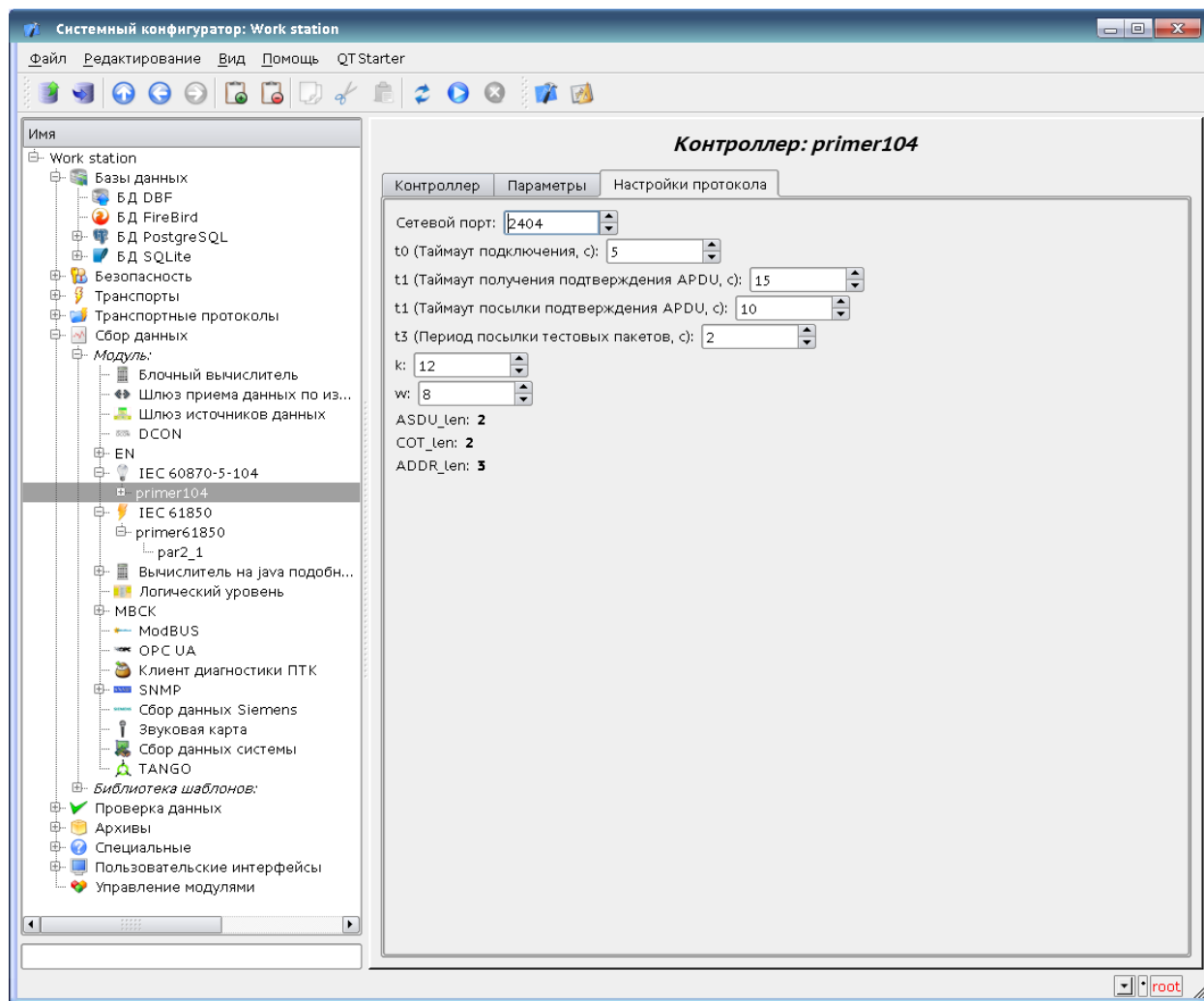


Рисунок 84

В ПП «СКАДА А-СОФТ» реализованы возможности настройки следующих параметров:

- сетевой порт – по умолчанию 2404;
- таймаут подключения (с);
- таймаут получения подтверждения APDU(с);
- таймаут отправки подтверждения APDU(с);
- период отправки текстовых пакетов(с);
- k- максимальное число неподтвержденных пакетов (значение от 2 до 99);
- w- количество пакетов до подтверждения (значение от 1 до 98).

Значения параметров устанавливаются в соответствии с сопроводительной документацией на контроллеры.

По умолчанию предустановлены следующие атрибуты:

длина адреса ASDU - ASDU\_len: 2 байта;

длина причины передачи - COT\_len: 2 байта;

общая длина адреса объекта - ADDR\_len: 3 байта.

На рисунке 86 изображено окно конфигурирования параметра контроллера модуля IEC 60870-5-104.

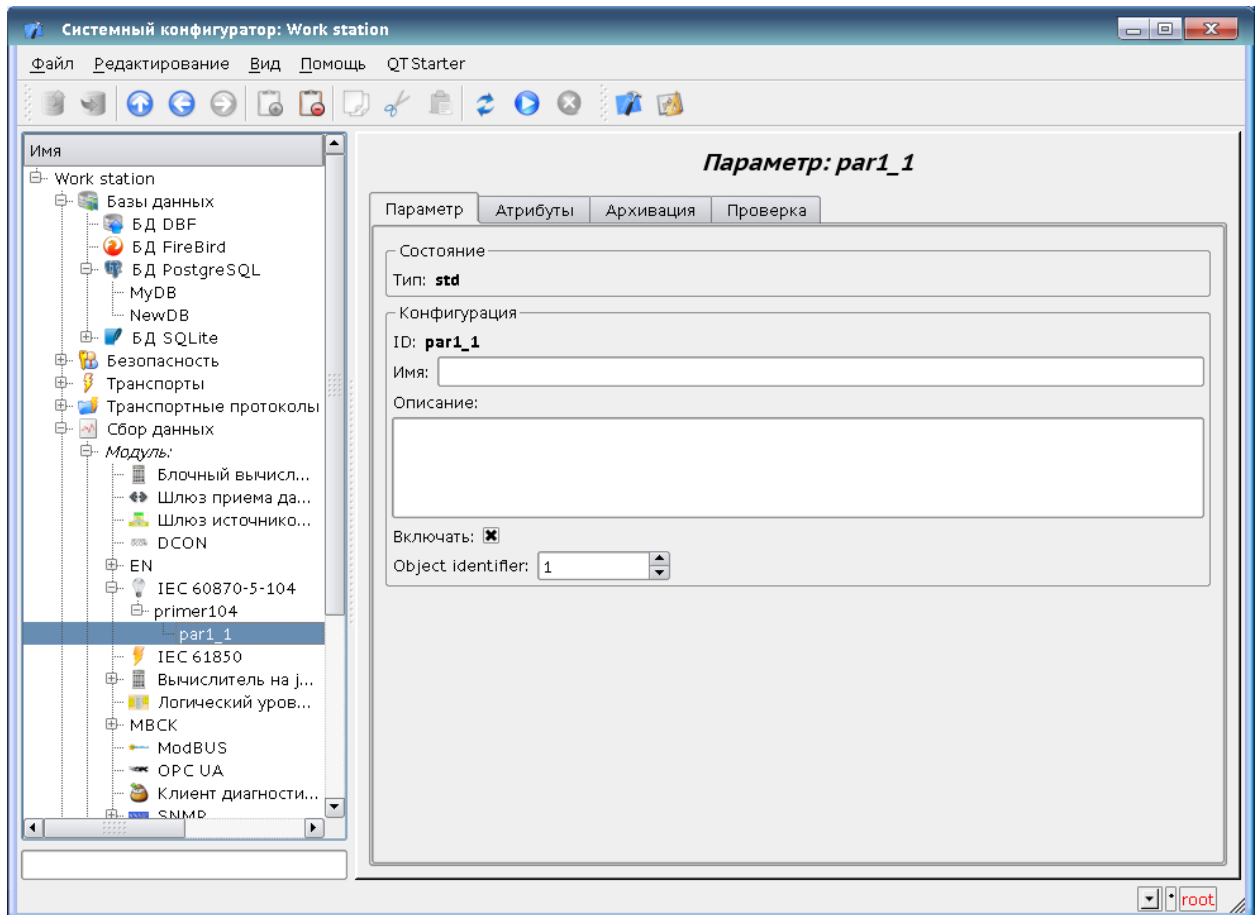


Рисунок 85

Индивидуальным атрибутом является поле для указания идентификатора объекта (object identifier), значение устанавливается в соответствии с документацией на контроллер.

## 2.8 Модуль Клиент диагностики ПТК

Модуль «Клиент диагностики ПТК» подсистемы «Сбор данных» предназначен для сбора ПП «СКАДА А-СОФТ» параметров о техническом состоянии используемого программно-технического комплекса. Функционирование модуля связано с настройкой файлов `dac` и `dsc`.

Для настройки сбора информации с параметров оборудования предварительно для всех серверов и АРМ системы необходимо произвести следующие действия:

1) На узле, используя «Менеджер установки пакетов», установить следующие пакеты AstraLinux:

`libxerces-c28`

`snmp`

`snmptt`

`tkmib`

`snmpd`

`libsnmp-base`

`libsnmp15`

`snmptrapfmt`

`lm-sensors`

`fancontrol`

`sensord.`

2) Скопировать с изделия программного ПМ26 на настраиваемый узел в папку `/opt` файл `dac_dsc_1.0-1_amd64.deb`. (сервисы и шаблоны `dac dsc`)

3) Запустить терминал `Fly` и установить указанный пакет командой:  
`dpkg -i dac_dsc_1_0-1.amd64.deb.`

4) На настраиваемом узле открыть файл `/etc/snmp/snmpd.conf` для редактирования (рисунок 87) и добавить к процессам строку 93: `proc scada` и строку 98: `dlmod ntp /usr/lib/ntp.so`

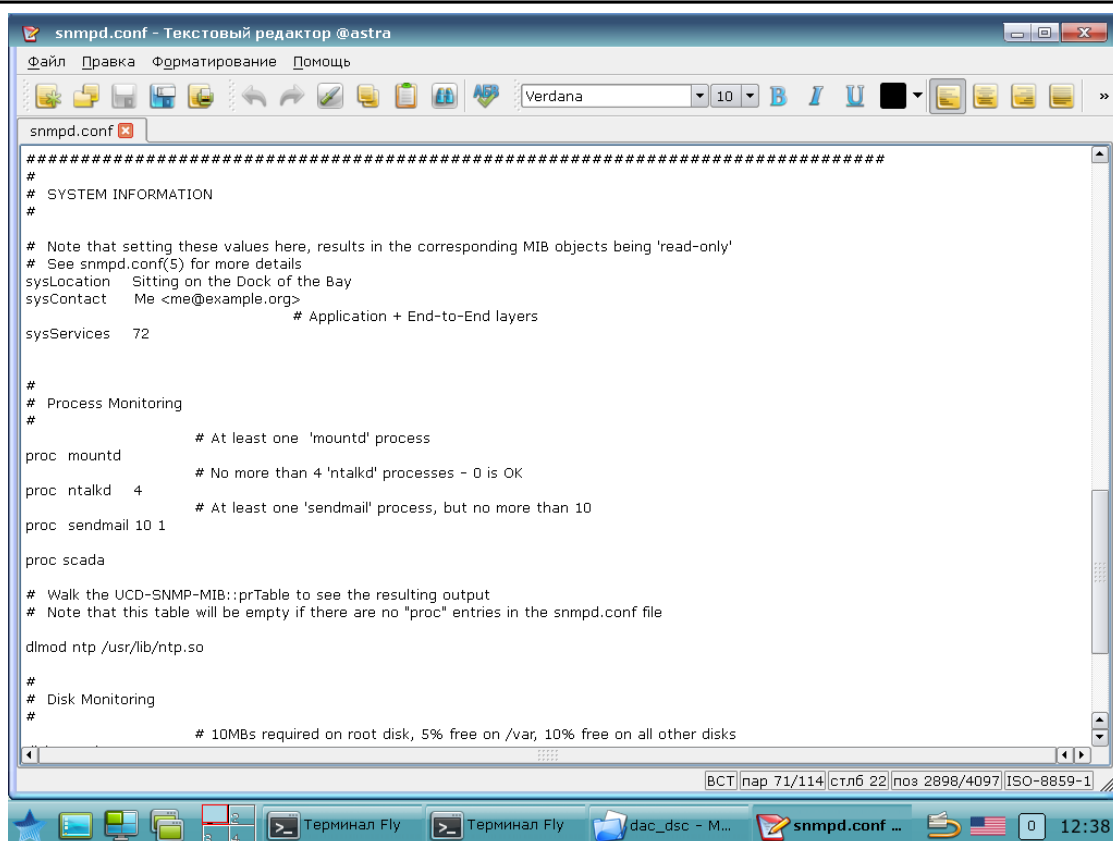


Рисунок 86

5) Сохранить файл.

6) Перезапустить snmpd, выполнив в терминале Fly следующую команду:  
`service snmpd restart`

7) Проверить включение сервисов snmpd, dac, dsc, выполнив команду:  
`chkconfig <имя_сервиса>`

Состояние сервисов должно быть on, при необходимости перезапустить командами:

```
service <имя_сервиса> stop  
service <имя_сервиса> start
```

8) Перезагрузить систему

9) Файлы dac.xml, представляют из себя пустой шаблон для добавления параметров конкретного проекта. Файлы находятся в директории /etc/dac.

10) Файлы dsc.xml, представляют из себя пустой шаблон для добавления параметров конкретного проекта. Файлы находятся в директории /etc/dsc.

11) В терминале Fly на всех узлах выполнить команды:

```
chmod -R 777 /etc/dac/dac.xml  
service dac stop  
service dac start
```



В случае сервера выполнить дополнительно:

```
chmod -R 777 /etc/dsc/dsc.xml
service dsc stop
service dsc start
```

### 2.8.1 Составление конфигурационного файла *dsc.xml*.

В теге <base> содержится перечень принимаемых параметров, каждый из которых находится в отдельном теге <snmp>. Основными полями тега <snmp> являются id и addr. Поле id – это девятизначный идентификатор, который назначается данному параметру и прописывается в файле dsc.xml и дублируется в поле Identifier параметра контроллера модуля «Клиент диагностики ПТК» подсистемы «Сбор данных». Поле addr – это OID, от которого принимается значение.

Тег <snmphosts> в случае сервера содержит список узлов сети, у которых отсутствует возможность запуска сервиса dsc и создания файла dsc.xml. К таким узлам относятся коммутаторы, ИБП, АВР. В этом списке в теге <host> содержатся шестизначный идентификатор узла и его IP адрес. Все параметры, получаемые от данного узла, имеют идентификаторы, у которых первые 6 знаков являются id узла, а 3 остальных – присваиваются по порядку. В случае сервера в теге <base> прописаны параметры, получаемые как от самого сервера, так и от других узлов, расположенных в теге <host>.

В случае АРМа тег <snmphosts> состоит из одной строки, причем в поле addr указан адрес “127.0.0.1”.

Тег <network> содержит трехзначный id, по которому файл dsc.xml сервера идентифицирует данный узел.

Тег <clients> содержит список серверов, которым данный узел намерен передавать параметры. В теге имеется поле id – для идентификации сервера и поле addr\_b – для IP адреса сервера.

Пример файла dsc.xml приведен в приложении 1.

### 2.8.2 Составление конфигурационного файла *dsc.xml*.

В теге <network\_c > расположено поле id данного сервера, которое присутствует в теге <clients> файлов dsc.xml всех узлов, которые передают параметры данному серверу. Тег <clients> файла dsc.xml содержит id = “901” и адрес “127.0.0.1”, это означает, что модуль «Клиент диагностики ПТК» подсистемы «Сбор данных» ПП «СКАДА А-СОФТ» работает на одном сервере с сервисом dsc.

В теге <servers> перечислены все узлы сети, включая данный сервер, на которых работает сервис dас и от которых принимаются параметры. В списке отсутствуют узлы сети, не имеющие сервиса dас. Поле id элементов списка является идентификатором узла, присвоенного в поле id тега <network> файла dас.xml, имеющемся на этом узле. Поле addr\_b является IP адресом узла.

В теге <base> файла dsc.xml содержится список параметров, получаемых от сервисов dас каждого из опрашиваемых узлов.

В теге <through> для каждого параметра указывается поле id, представляющее девятизначный идентификатор, который был назначен этому параметру в файле dас.xml. Поле serv устанавливается равным полю id тега <server> файла dsc.xml, а также полю id тега <network> файла dас.xml, в котором параметр был определен. В поле vtype тега <through> устанавливается тип параметра.

Файлы dас.xml создаются на всех узлах, где будет запущен сервис dас - это АРМы и сервера. Файлы dsc.xml создаются только на серверах. Сервисы dsc запускаются только на серверах. Файлы dас.xml и dsc.xml должны храниться на одном сервере. Пример файла dsc.xml приведен в приложении 2.

Все параметры, определенные в файлах dас.xml и dsc.xml должны быть созданы в контроллере модуля «Клиент диагностики ПТК» подсистемы «Сбор данных». Идентификатор параметра должен совпадать с девятизначным идентификатором определенным в файлах dас и dsc. Тип параметра должен быть установлен Analog. На рисунке 88 представлено окно настройки параметра контроллера модуля «Клиент диагностики ПТК».

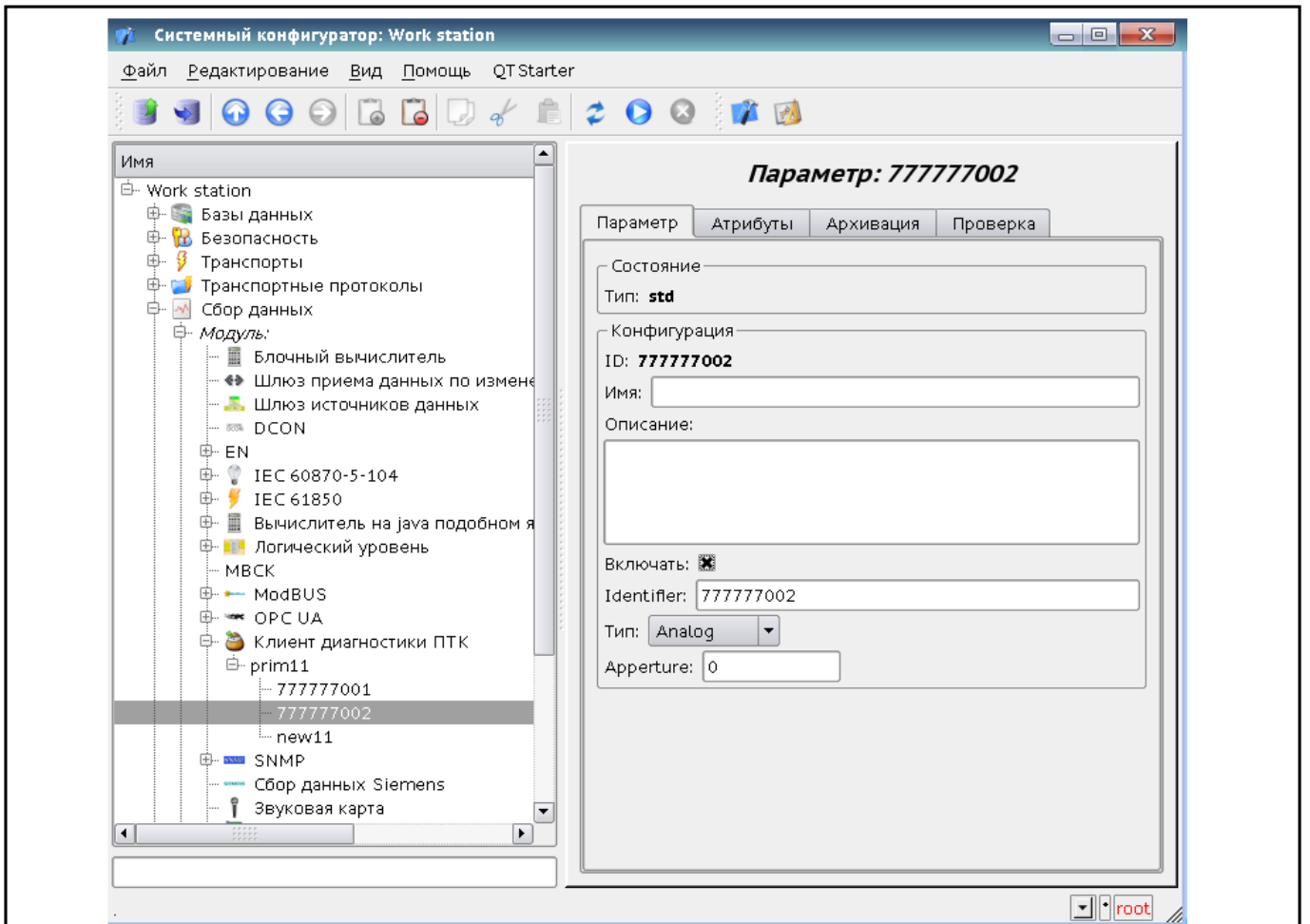


Рисунок 87

## 2.9 Модуль источника данных OPC UA

В модуле OPC UA подсистемы «Сбор данных» производится настройка узла-клиента, сервер настраивается в модуле OPC UA подсистемы Транспортные протоколы (описание приведено в части 1 настоящего руководства оператора).

Для настройки клиента OPC UA необходимо создать контроллер в модуле OPC UA подсистемы «Сбор данных».

Для контроллера кроме стандартных настроек состояния и конфигурации доступны следующие поля:

- Период синхронизации с удаленной станцией (сек) - для отключения периодической синхронизации установить значение ноль;
- Конечный узел – указывается IP адрес сервера и порт соединения;
- Политика безопасности – позволяет выбрать алгоритм безопасности: нет, Basic128rsa15, Basic256. В зависимости от выбранного значения изменяется длина приватного ключа (PEM);
- Режим безопасности сообщения – для выбора доступны следующие значения: нет, подпись, подпись&шифрование. Наиболее безопасным является подпись&шифрование;
- Сертификат (PEM) - указывается при необходимости, в соответствии с настройками сервера;
- Приватный ключ (PEM) - указывается при необходимости, в соответствии с настройками сервера;
- Аутентификация пользователь – имя пользователя;
- Аутентификация пароль – пароль;
- Ограничение количества атрибутов параметра: 100.

На рисунке 89 представлено окно настройки контроллера “testOPC”.

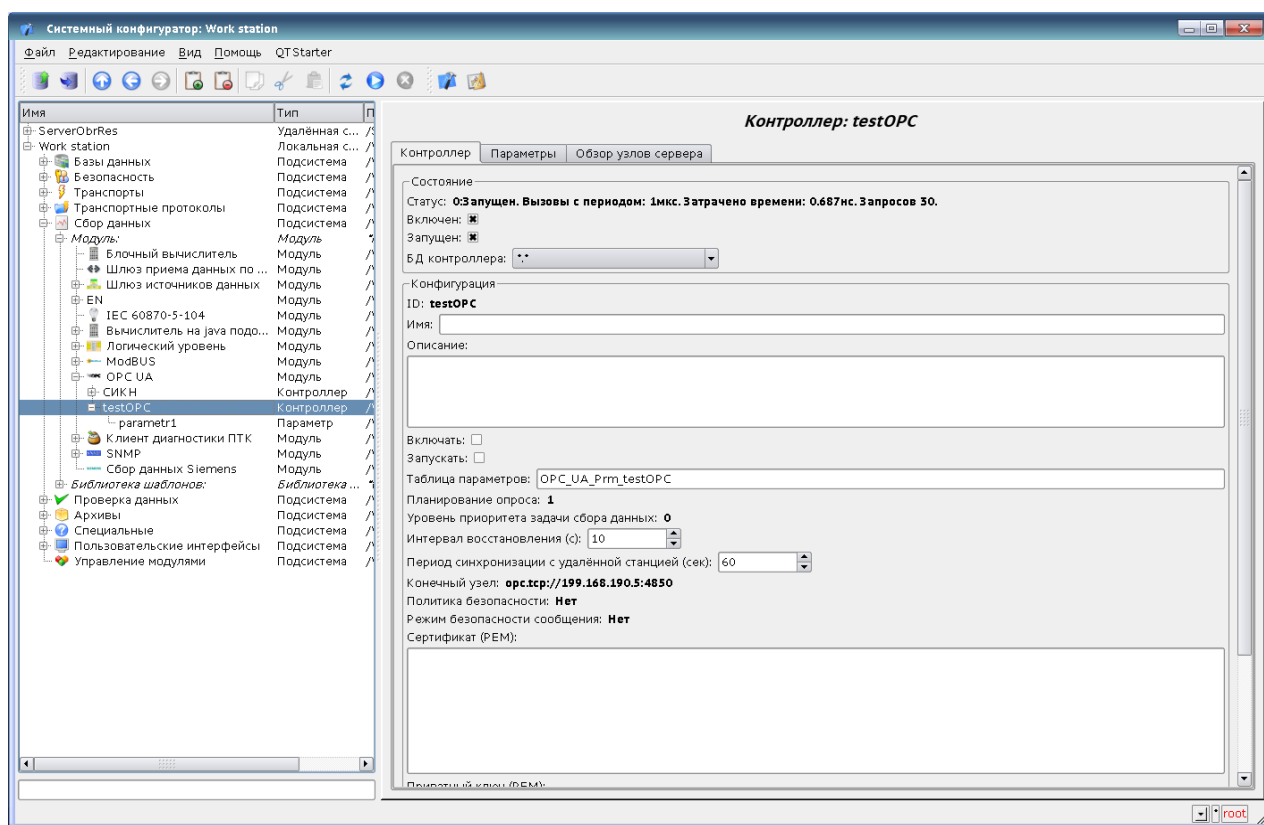


Рисунок 88

Добавляя параметры контроллера можно указать список узлов, представляющий из себя список переменных и контейнеров (объектов). Добавление узла производится последовательным выбором необходимого объекта в поле «Добавить узел» (рисунок 90). После включения параметра все переменные переносятся в перечень атрибутов параметра (рисунок 91) и автоматически заполняется поле список узлов. В подсказке к данному полю описан формат описания переменных (рисунок 92).

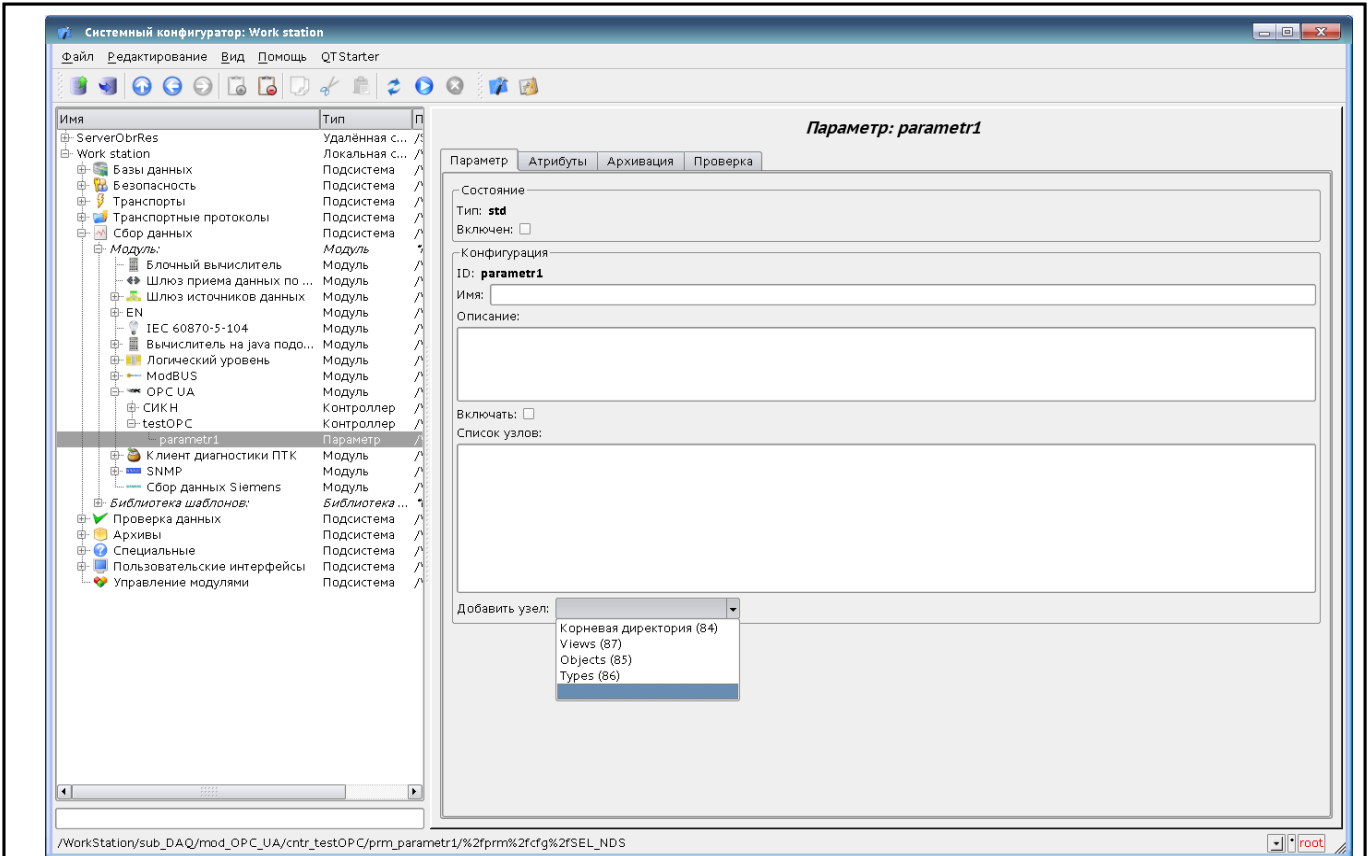


Рисунок 89

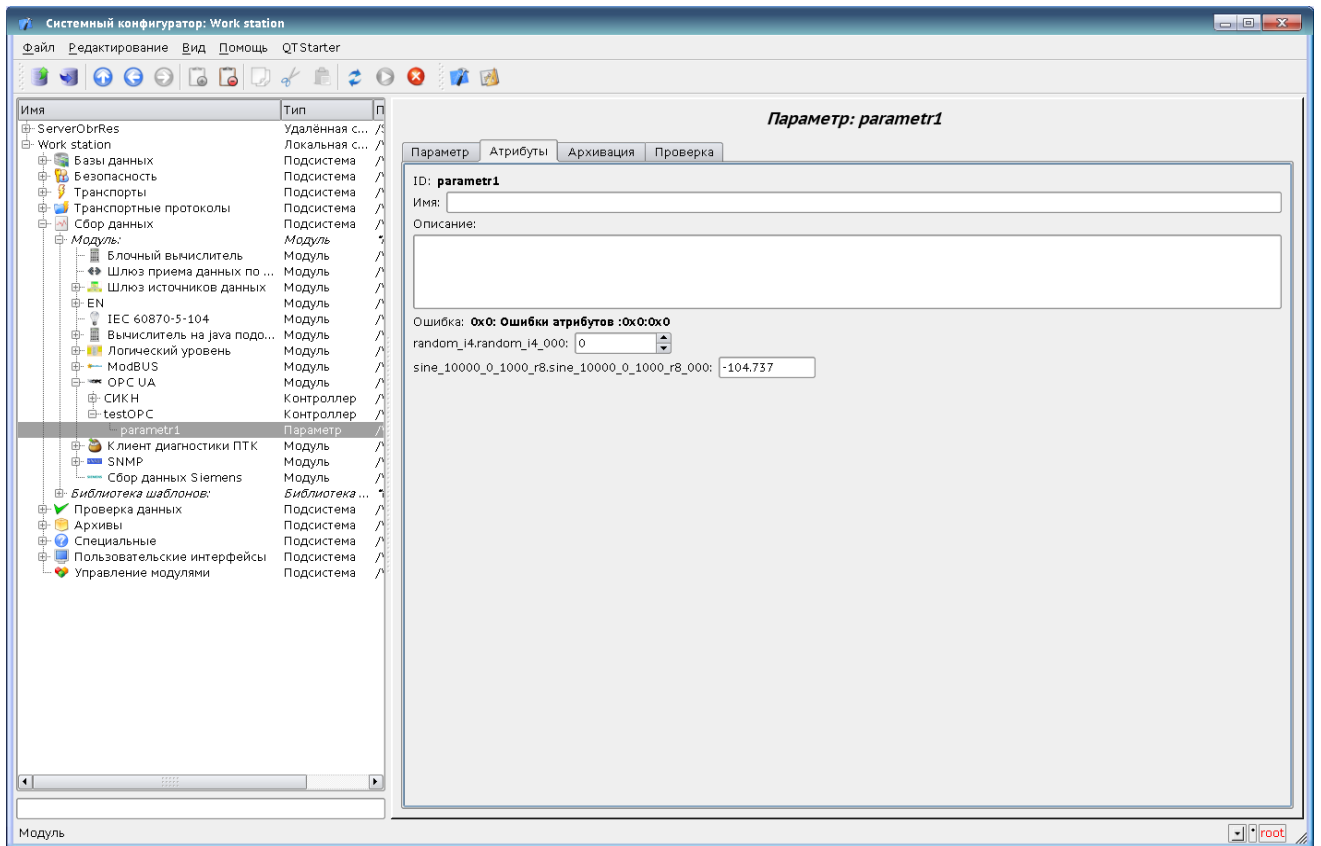


Рисунок 90

Список переменных и контейнеров (Объектов). Все переменные будут помещены в перечень атрибутов параметра. Переменные записываются отдельными строками в формате: [ns:id].

Где:

ns - область имён, числом; нулевое значение может быть опущено;

id - идентификатор узла, числом, строкой, строкой байт или GUID.

Пример:

84 - корневая директория;

3:"BasicDevices2" - узел базовых устройств в области имён 3 и в виде строки;

4:"61626364" - узел в области имён 4 и в виде строки байт;

4:{40d95ab0-50d6-46d3-bffd-f55639b853d4} - узел в области имён 4 и в виде GUID.

Рисунок 91

При подключении клиента также необходимо принять соглашение о сертификате сервера (согласиться в случае его отсутствия).

## 2.10 Модуль источника данных HART

Модуль HART предназначен для взаимодействия СКАДА с HART-устройствами.

Он позволяет конфигурировать параметры HART-устройств без использования HART-коммуникатора. Данные возможности важны, когда необходимо дистанционно с АРМ инженера настраивать параметры HART-устройств, находящихся в удаленных, или труднодоступных местах, позволяя конфигурировать, диагностировать и калибровать HART-устройства с АРМ инженера/оператора.

Алгоритм сбора данных модуля HART позволяет сравнивать и проверять полученные результаты собираемых данных со значениями, передаваемыми по каналу 4..20 мА на контроллер ТПТС.

Модуль HART использует механизм резервирования серверов СКАДА, что гарантирует непрерывную работу модуля HART и СКАДА в случае нештатного отключения одного из серверов, использует централизованный механизм архивирования СКАДА, что позволяет обеспечить архивирование переменных модуля HART поступающих от HART-устройств за определенный временной интервал.

При открытии модуля HART кроме стандартных вкладок «Контроллеры» и «Помощь» доступна вкладка «Настройка модуля», которая при установке соответствующего флага позволяет выводить в консоль запуска платформы отладочную информацию.

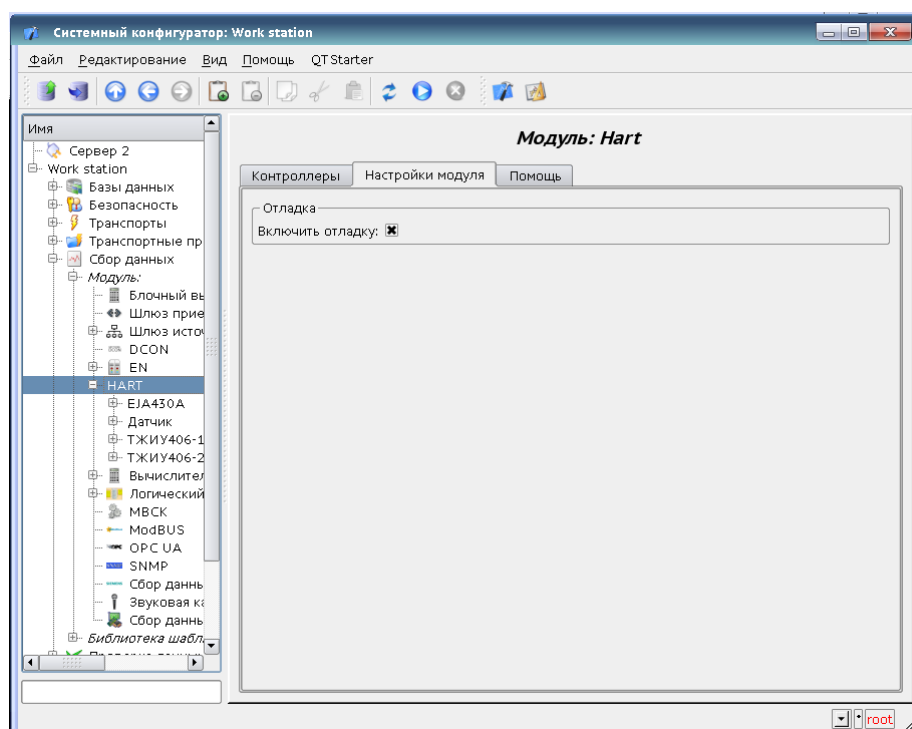


Рисунок 92



Для добавления нового контроллера в модуль HART необходимо выбрать пункт контекстного меню «Добавить» и ввести ID и имя добавленного устройства. На вкладке «Контроллер» созданного элемента отражаются настраиваемые параметры для подключения HART-устройства (рисунок 94).

**Контроллер: EJA430A**

Контроллер | Параметры

Состояние

Статус: **2:Выключен.**  
Включен:   
Запущен:   
БД контроллера: \*.\*

Конфигурация

ID: EJA430A  
Имя:  
Описание:

Включать:   
Запускать:   
Таблица параметров: HartPrm\_EJA430A  
Адрес транспорта: Serial.com1  
Широковещательный адрес мультиплексора: 0  
Широковещательный адрес устройства: 0  
Канал мультиплексора: 0  
Команды для опроса: 1,3  
Период опроса(мс): 500  
Попытки опроса датчика: 15  
Попытки выполнения команды: 25  
Расширенный адрес мультиплексора: B003000000  
Расширенный адрес устройства: B7041F4FAD

\* root

Рисунок 93

Индивидуальными настройками контроллера являются:

*Таблица параметров* - определяет имя таблицы, где будут храниться параметры контроллера;

*Адрес транспорта* – содержит адрес транспорта, созданного в модуле «Транспорты» - «Последовательный интерфейс» - «Выходной транспорт», на котором будет организовано взаимодействие с HART-устройством (создание транспорта описано в части 1 руководства оператора);

*Широковещательный адрес мультиплексора* - адрес подключаемого HART-мультиплексора, настраиваемый на корпусе прибора;

*Широковещательный адрес устройства* – по умолчанию у HART-устройств этот адрес нулевой;

*Канал мультиплексора* – номер одного из 16 или 32 каналов на плате MCR-S-MUX-TV мультиплексора, к которому подсоединено подключаемое HART-устройство;

*Период опроса(мс)*- период опроса датчика по заданному каналу мультиплексора;

*Попытки опроса датчика* – количество попыток опроса датчика в случае отсутствия ответа;

*Попытки выполнения команды* – количество попыток запроса выполнения команд у логического контроллера;

*Расширенный адрес мультиплексора* – значение в поле устанавливается автоматически в результате выполнения команды cmd0;

*Расширенный адрес устройства* – значение в поле устанавливается автоматически в результате выполнения команды mux0;

*Команды для опроса* – позволяет выбрать существующую команду для опроса (по введенному OID), при этом на вкладке «Атрибуты» отразятся атрибуты, соответствующие выбранной команде, а в поле «Описание» соответствующее описание команды. Для команд «mux1, mux2, mux3» допустимо вводить в данном поле только номера команд в диапазоне от 1 до 3 через «,» (например: «1,2,3»). В таблице 1 представлено описание поддерживаемых команд. Если команда не поддерживается HART-устройством, на передней панели мультиплексора загорается красный индикатор. Если команды для опроса не указаны, то опрос данного устройства производиться не будет.

Таблица 1

Идентификатор (OID)	Краткое описание команды
cmd0	Чтение уникального идентификатора HART-мультиплексора
mux0	Чтение уникального идентификатора HART-устройства
mux1	Чтение первичной переменной
mux2	Чтение первичной переменной как величины тока и в процентах от диапазона
mux3	Чтение четырех основных переменных и токового значения первичной переменной
mux14	Чтение информации сенсора первичной переменной
mux15	Чтение информации о выходном сигнале по первичной переменной
mux34	Запись значения демпфирования первичной переменной
mux35	Запись значения диапазона первичной переменной
mux36	Установка значения верхней границы диапазона сенсора

Идентификатор (OID)	Краткое описание команды
mux37	Установка значения нижней границы диапазона сенсора
mux38	Сброс флага изменения конфигурации
mux40	Вход/выход из режима фиксированного токового значения первичной переменной
mux41	Самотестирование прибора
mux43	Установка нуля первичной переменной
mux45	Подстройка нуля ЦАП первичной переменной
mux46	Подстройка коэффициента усиления ЦАП первичной переменной
mux47	Запись функции преобразования первичной переменной процесса
mux48	Чтение дополнительного статуса HART-устройства
mux108	Запись номера команды для монополюсного режима
mux109	Управление режимом монополюсной работы

Разработанные команды конфигурирования имеют следующие особенности:

- команда 34 (OID - mux34) задает величину демпфирования первичной переменной. Для ее отправки необходимо задать «Время демпфирования»;

- команда 35 (OID - mux35) записывает данные диапазона значений датчика. Для ее отправки необходимо ввести «Код единиц диапазона» (целое), и значения верхнего и нижнего предела диапазона (вещественное). Целые значения кодов единиц перечислены в приложении 3.

- команды 45, 46 (OID - mux45, mux46) задают верхнее и нижнее значение тока ЦАП (вещественные). Это команды из разряда калибровочных. В некоторых датчиках фиксировано значение 4..20мА;

- команда 40 (OID - mux40) задает фиксированное значение тока первичной переменной. Для этого в поле вводится вещественное значение для входа в режим фиксированного тока. Отправленное нулевое значение означает выход из режима фиксированного тока;

- команда 47 (OID - mux47) задает функцию преобразования первичной переменной. Значение 0 соответствует линейной функции. Могут быть заданы другие функции передачи, например, квадратный корень;

- команды 36, 37, 43 (OID - mux36, mux37, mux43) задают верхнюю границу, нижнюю границу и нуль диапазона. При нажатии на кнопку выполнения команды, выбранная величина устанавливается согласно текущему значению первичной переменной.

Для конфигурирования связи по HART-протоколу необходимо для контроллера создать параметр и выбрать в поле «OID» имя HART-команды. При этом, на вкладке

«Атрибуты» можно убедиться что, в зависимости от введенного идентификатора команды - OID, вид окна будет отличаться (например, рисунки 95 - 100).

**Параметр: cmd0**

Параметр | Атрибуты | Архивация | Проверка

ID: **cmd0**

Имя: cmd0

Описание:  
Чтение уникального идентификатора мультиплексора

Ошибка: **0**

Выполнить команду:

ID изготовителя: **176**

Тип устройства: **3**

Количество преамбул: **2**

Ревизия универсальных команд: **5**

Специфичная для устройства общ. ревизия: **1**

Ревизия программного обеспечения: **6**

Ревизия аппаратной части: **Ревизия=0 Сиг.код=1**

Доп. функции устройства:

Расширенный адрес: **b0 3 0 0 0**

Рисунок 94

**Параметр: mux15**

Параметр | Атрибуты | Архивация | Проверка

ID: **mux15**

Имя:

Описание:  
Чтение информации о выходном сигнале по первичной переменной

Ошибка: **0**

Выполнить команду:

Код выбора тревоги: **0**

Код функции передачи: **0**

Единицы измерения: **КПа**

Верхнее значение диапазона П1: **10000**

Нижнее значение диапазона П1: **0**

Значение демпфирования переменной П1: **0.5**

Код защиты от записи: **0**

Код частной метки: **55**

Рисунок 95

**Параметр: mhx35**

Параметр | Атрибуты | Архивация | Проверка

ID: **mhx35**

Имя:

Описание:  
Запись значения диапазона первичной переменной

Ошибка: **0**

Выполнить команду:

Последний статус: **passed configuration changed**

Код единиц измерения диапазона:

Верхний предел диапазона:

Нижний предел диапазона:

Рисунок 96

**Параметр: mhx2**

Параметр | Атрибуты | Архивация | Проверка

ID: **mhx2**

Имя:

Описание:  
Чтение первичной переменной как величины тока и в процентах от диапазона

Ошибка: **0**

Выполнить команду:

Величина тока переменной П1(мА): **4.00075**

Процент переменной П1 от ее значения диапазона: **0.0047056**

Рисунок 97

**Параметр: mhx45**

Параметр | Атрибуты | Архивация | Проверка

ID: **mhx45**

Имя:

Описание:  
Команда выполняет подстройку нуля ЦАП первичной переменной

Ошибка: **0**

Выполнить команду:

Последний статус: **Not in proper current mode (fixed at 4 mA or 20 mA)**

Нуль ЦАП переменной П1 (мА):

Рисунок 98

**Параметр: mux1**

Параметр | Атрибуты | Архивация | Проверка

ID: **mux1**

Имя:

Описание:

Чтение первичной переменной

Ошибка: **0**

Выполнить команду:

Единицы измерения П1: **кПа**

Переменная 1: **0.36481**

Рисунок 99

При выполнении команд конфигурирования HART-устройства (команды от 34 до 48) в поле «Последний статус» возвращается статус выполнения команды. Расшифровка статусов команд модуля представлена в приложении 4.

### 3 Настройка резервирования

#### 3.1 Резервирование модуля «Базы данных»

Для настройки резервирования модуля «Базы данных» необходимо в модуле «Транспорты» создать внешний хост на каждом из узлов. Для этого *на стороне основного узла*: в поле Id ввести имя резервного узла, а в поле «Адрес» ввести IP адрес резервного узла. *На стороне резервного узла*: в поле Id ввести имя основного узла, а в поле «Адрес» ввести IP адрес основного узла. В поле «Режим» выбрать «Пользов. и Системн.». На рисунке 101 приведен пример модуля «Транспорты» для резервного узла.

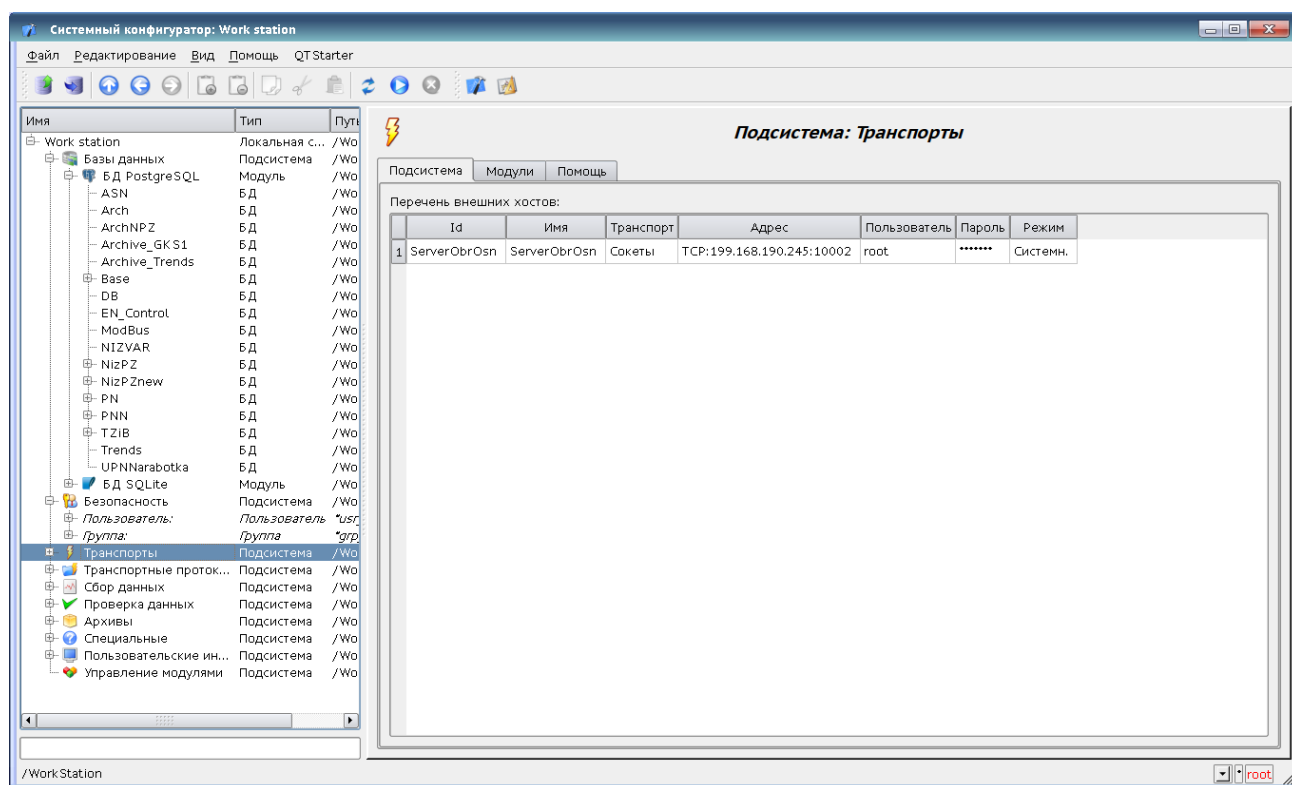


Рисунок 100

Далее для каждого узла выполнить следующие действия:

- в окне «Базы данных» на вкладке резервирование в таблице «Станции» добавить узел;
- *на стороне основного узла* - в поле Id ввести имя резервного узла, добавленного в транспорты, *на стороне резервного узла* - в поле Id выбрать имя основного узла, добавленного в транспорты;
- в таблице «Таблицы» создать по строке на каждую из таблиц, которые требуется резервировать;

- в поле «БД» необходимо ввести путь до базы данных, содержащей нужную таблицу;
- в поле «Таблица» ввести Id таблицы;
- в поле «Предп. исп.» выбрать то же имя узла, что и в таблице «Станции»;
- сохранить сделанные изменения.

На рисунке 102 приведен пример настройки модуля «Базы данных» для резервного узла.

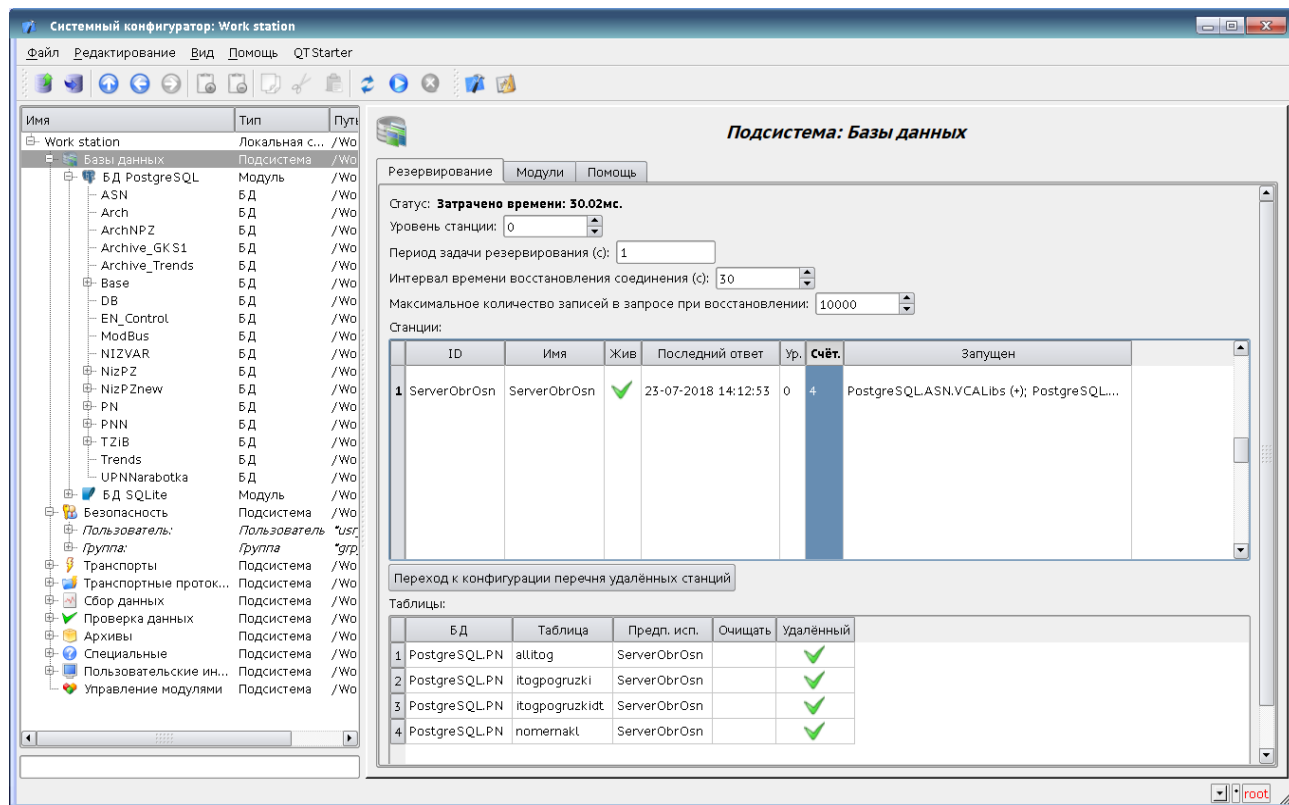


Рисунок 101



### 3.2 Резервирование модуля «Сбор данных»

Для работы механизма синхронизации данных на каждом сервере хранится список адресов серверов, от которых возможно получение данных. При восстановлении работоспособности сервера после его отказа каждый контроллер SCADA-системы (настроенный на синхронизацию данных) запускает отдельный поток, в котором выполняется загрузка из другого сервера необходимых архивных данных по каждому атрибуту. Если список состоит из нескольких серверов, то выбирается работающий сервер, содержащий аналогичный контроллер. В случае отсутствия подходящего сервера процесс синхронизации прекращается по данному контролеру. Начало запрашиваемого временного промежутка по каждому архиву атрибута выбирается исходя из последней метки времени поступивших данных в архив текущего сервера. Окончание запрашиваемого временного промежутка задается уже на сервере, с которым синхронизируют данные: указывается время начала процесса получения архива. Переданные данные атрибута, помещаются в архив, независимо от процесса архивации атрибута, то есть данные не помещаются в буфер архиватора, а сразу записываются в БД.

Для увеличения быстродействия процесса синхронизации передача данных между серверами выполняется через отдельный транспорт, не задействованный в других операциях SCADA-системы. Кроме того, запись архивов в БД происходит по отдельным подключениям к ней для каждого контроллера SCADA-системы.

Для настройки резервирования модуля «Сбор данных» необходимо создать внешний хост в модуле «Транспорты» (как описано в пункте 3.1). Далее для каждого узла выполнить следующие действия:

- в поле «Резерв.» выбрать «Архивное»;
- в поле «Предп. исп.» выбрать то же имя узла, что и в таблице «Станции»;
- сохранить сделанные изменения.

На рисунке 103 приведен пример настройки модуля «Сбор данных» для резервного узла.

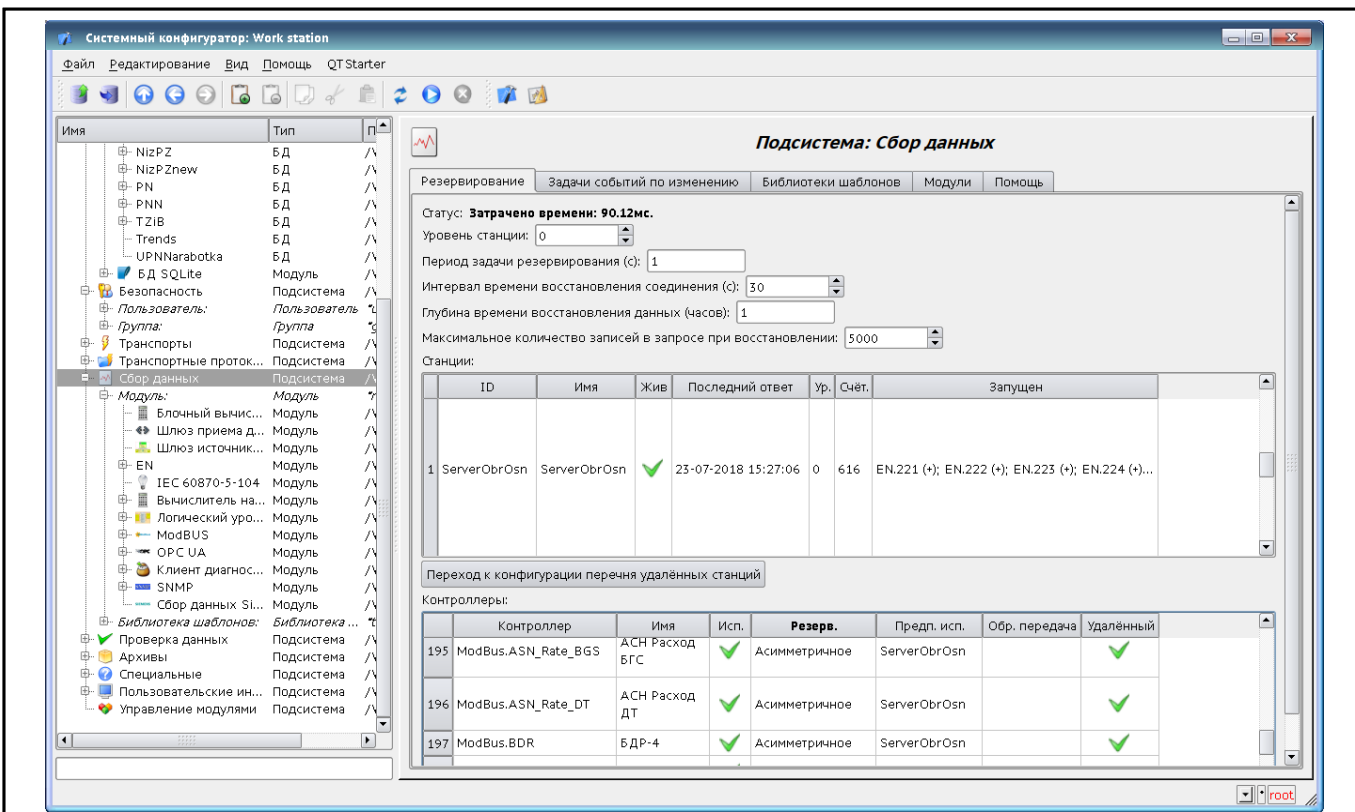


Рисунок 102

## **4 Рекомендации по работе с проектом**

### **4.1 Редактирование графической части проекта АСУ ТП**

При разработке графической части проекта все элементы, группы элементов и видеокадры необходимо создавать во вкладке «Виджеты», а затем во вкладке «Проект» подключать разработанные видеокадры. При внесении изменений в проект все изменения необходимо вносить в виджеты проекта (вкладка «Виджеты»). В этом случае изменения автоматически будут отражены в проекте. При редактировании виджетов проекта в окне «Проект» после перезагрузки изменения будут утрачены.

### **4.2 Настройка отображения проекта на удаленном АРМ**

Для настройки удаленного подключения к проекту: в системном конфигураторе необходимо выбрать «Пользовательский интерфейс» → «Рабочий пользовательский интерфейс» → «Удаленное взаимодействие». Данная вкладка предоставляет возможность конфигурирования вывода видеокадров на экраны удаленного АРМ (рисунок 104). Для этого выбрать в списке станций для удаленного открытия видеокадров на экранах «Local». Задать для каждого экрана группу пользователей, для которых разрешено управление отображением видеокадров 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 видеокадров на экраны для удаленного рабочего места, необходимо выбрать название этого рабочего места из списка станций для удаленного открытия видеокадров. Затем задать количество экранов и их имена для выбранного рабочего места. Также можно запросить имена экранов с удаленной станции, нажав кнопку «Обновить».

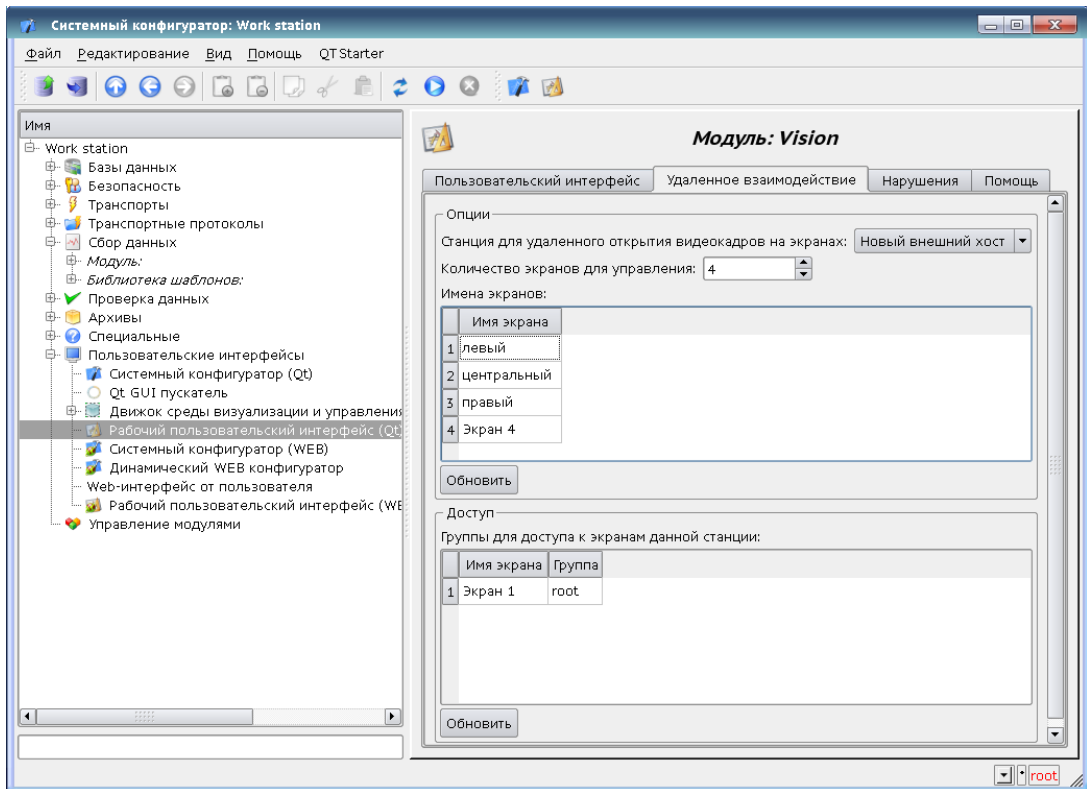


Рисунок 103

### 4.3 Настройка отображения проекта на нескольких мониторах

Для открытия окна проекта на нескольких мониторах одного АРМ необходимо в окне системного конфигуратора выбрать «Пользовательские интерфейсы»→«Рабочий пользовательский интерфейс (Qt)», на вкладке «Пользовательский интерфейс» ввести порядок отображения проектов на мониторах в строке «Перечень запускаемых проектов» в формате: 'PrjName - 1', где PrjName - название проекта; 1 - номер монитора (рисунок 105). Сохранить сделанные изменения.

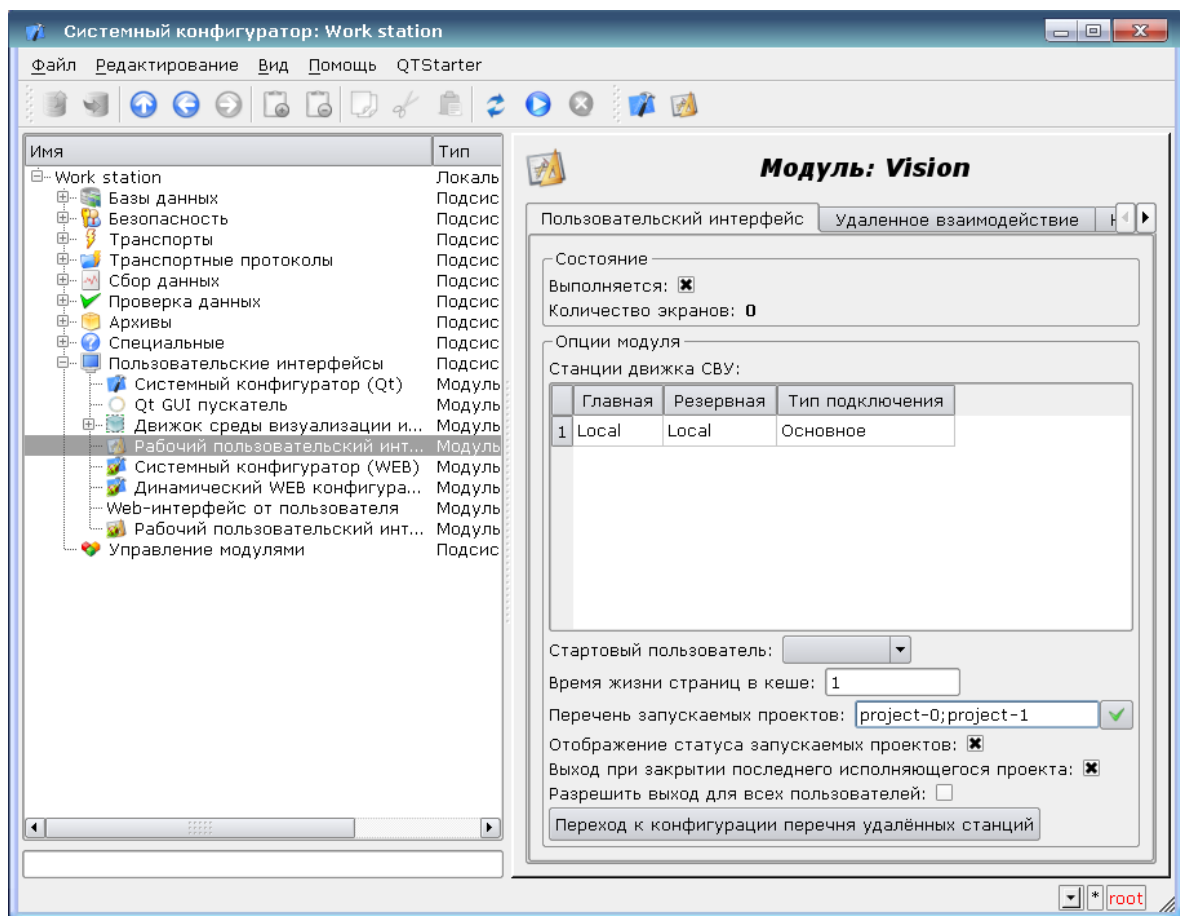


Рисунок 104

## 4.4 Сохранение проекта АСУ ТП и его частей

### 4.4.1 Рекомендации

Для сохранения проекта АСУ ТП и его частей рекомендуется использовать СУБД PostgreSQL, где сохраняется по умолчанию.

Для сохранения проекта и его частей необходимо создать новую БД. Количество БД должно быть достаточным для реализации поставленной задачи и не должны содержать идентичные элементы.

Сохранение архивов необходимо производить в отдельной БД.

### 4.4.2 Что не рекомендуется делать при сохранении проекта

Не рекомендуется один и тот же элемент сохранять в разных БД, т.к. это приводит к непредсказуемым результатам при одновременной загрузке БД. Под проектом АСУ ТП понимаем набор сконфигурированных БД, источников данных, включая их обработку и

сохранение в архивы, а также сконфигурированный графический интерфейс оператора. Не путать с вкладкой «Проект» графического редактора.

Не рекомендуется непосредственно менять стандартные конфигурации и элементы стандартных библиотек, а также сохранять собственные, новые, библиотеки и элементы в базах данных стандартных библиотек.

#### 4.5 Перенос конфигурации СКАДА из одного проекта в другой

При переносе отдельной БД необходимо произвести ее копирование с источника и загрузку в новом проекте приемника.

Перенос конфигурации производится в зависимости от типа БД:

- для БД *SQLite* – необходимо скопировать нужный файл \*.db из директории БД старого проекта в директорию баз данных нового (например, /var/spool/scada/DATA/). После этого необходимо запустить ПП «СКАДА А-СОФТ» и провести подключение БД в СКАДА в модуле БД *SQLite*, т.е. создать объект БД, ввести для нее корректные параметры в поле «Адрес» и сохранить сделанные изменения. В строке состояние поставить галочку «Включен» и нажать на кнопку «Загрузить систему из этой БД» (производится аналогично описанному в пункте 1.2.2 настоящей части руководства оператора).

- для БД *PostgreSQL* – необходимо создать дамп копируемой БД. Для этого в окне терминала Fly ввести команду:

```
pg_dump -U postgres -Ft -c -d <имя_БД> | gzip > /путь приемника/<ИмяБД>.tar.gz
```

После чего поместить дамп в директорию /var/opt АРМ приемника и восстановить БД во временную новую БД. Для этого необходимо из системного меню вызвать окно терминала Fly и запустить оболочку psql от имени непривилегированного пользователя ОС - postgres (администратор БД), выполнив команду:

```
psql -U postgres
```

При этом система попросит ввести пароль для пользователя *postgres*, заданный при установке базового ПО. В результате, в окне терминала высветится приглашение postgres=#.

Далее следует создать новую базу данных командой:

```
create database <"имя временной БД">;
```

Выйти из программной оболочки psql командой \q.

Затем выполнить команду:

```
gunzip -c /var/opt/<имяБД>.tar.gz | pg_restore -U postgres -  
Ft -d <"имя временной БД"> -v
```

После этого необходимо запустить ПП «СКАДА А-СОФТ» и в модуле БД PostgreSQL создать БД с именем временной БД (в соответствии с 1.2), ввести для нее корректные параметры в поле «Адрес», сохранить сделанные изменения. В строке состояние поставить галочку «Включен» и нажать на кнопку «Загрузить систему из этой БД». Далее для каждого загруженного контроллера необходимо изменить «БД контроллера» на «PostgreSQL.GenDB» и сохранить данные в БД.

После пересохранения всех контроллеров в основной БД проекта временную БД можно удалить. Для этого необходимо в окне терминала Fly ввести следующие команды:

```
psql -U postgres  
drop database <"имя временной БД">;  
\q
```

и провести перезагрузку ПП «СКАДА А-СОФТ».

## ПРИЛОЖЕНИЕ 1

### Пример листинга файла dac.xml

```
<?xml version="1.0" encoding="utf8"?>
<config xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation='dac.xsd' codepage="utf8">

  <dirs>
    <dir name="etc" path="etc" />
    <dir name="log" path="log" />
    <dir name="tmp" path="tmp" />
  </dirs>

  <view x="5" y="25" width="1024" height="600" />

  <debug level="8" wnd="no" log="no" />

  <snmphosts>
    <host id="100001" addr="127.0.0.1" /> <!-- ServerCpuMain -->
    <host id="100003" addr="199.168.190.22" />
  <!--MOXA Modbus1-->

  </snmphosts>

  <threads>
    <thread type="snmp" tupdt="1000" mupdt="60000"
caption="datathread_snmp" logid="001"/>
  </threads>

  <network id="001" port_r="3725" port_t="3726" logid="002" /> <!--
080-->

  <clients>
    <client id="055" addr_b="127.0.0.1" addr_r="127.0.0.1"
caption="dsc"/>
    <client id="056" addr_b="199.168.190.16"
addr_r="199.168.190.16" caption="dsc"/>
  </clients>

  <logs>
    <log id="001" caption="snmpthread"/>
    <log id="002" caption="udpserver"/>
  </logs>

  <base>
  <!-- ServerCpuMain 100001 -->
  <!-- snmpd connect -->
  <snmp id = "100001000"
ptype = "value"
vtype = "short"
addr = ".1.3.6.1.2.1.1.3.0"
fval = "1"
gval = "0"
rexp = ".*\s+"
update = "1000" >

  </snmp>

  <!-- ipHCInOctets(loadnetwork) -->
  <snmp id = "100001004"
ptype = "value"
```



```
        vtype = "string"
        addr  = ".1.3.6.1.2.1.31.1.1.1.2.4"
        rexp  = ".*\s+"
        sdep  = "100001000"
        update = "10000" >
</snmp>
<!-- ipHCOctets(loadnetwork) -->
<snmp    id    = "100001005"
        ptype = "value"
        vtype = "string"
        addr  = ".1.3.6.1.2.1.31.1.1.1.15.4"
        rexp  = ".*\s+"
        sdep  = "100001000"
        update = "10000" >
</snmp>

<!-- physicalIoiallMb -->
<snmp    id    = "100001008"
        ptype = "value"
        vtype = "double"
        rexp  = ".*\s+"
        addr  = ".1.3.6.1.2.1.25.2.3.1.5.1"
        sdep  = "100001000"
        math  = "* ;1024 / ;1024 / ;1024"
        send  = "no"
        update = "10000" >
</snmp>

<!-- physicalUsedlMb -->
<snmp    id    = "100001010"
        ptype = "value"
        vtype = "double"
        rexp  = ".*\s+"
        addr  = ".1.3.6.1.2.1.25.2.3.1.6.1"
        sdep  = "100001000"
        math  = "* ;1024 / ;1024 / ;1024"
        send  = "no"
        update = "10000" >
</snmp>

<!-- physicalFreelMb -->
<calc    id    = "100001011"
        ptype = "value"
        vtype = "double"
        math  = ",100001008 -,100001010" >
</calc>
<!-- discIoiallMb -->
<snmp    id    = "100001012"
        ptype = "value"
        vtype = "double"
        rexp  = ".*\s+"
        addr  = ".1.3.6.1.2.1.25.2.3.1.5.32"
        sdep  = "100001000"
        math  = "* ;4096 / ;1024 / ;1024"
        send  = "no"
        update = "10000" >
</snmp>

<!-- discUsedlMb -->
<snmp    id    = "100001013"
        ptype = "value"
        vtype = "double"
        rexp  = ".*\s+"
```

```

        addr      = ".1.3.6.1.2.1.25.2.3.1.6.32"
        sdep      = "100001000"
        math      = "* ;4096 / ;1024 / ;1024"
        send      = "no"
        update    = "10000" >

</snmp>

<!-- discFree 1Mb -->
<calc      id      = "100001014"
          ptype    = "value"
          vtype    = "double"
          math     = ",100001012 -,100001013" >

</calc>

<!-- MOXA Modbus1 -->
<!-- snmpd connect -->
<snmp      id      = "100003000"
          ptype    = "value"
          vtype    = "short"
          addr     = ".1.3.6.1.2.1.1.3.0"
          fval     = "1"
          gval     = "0"
          rexp     = ".*\s+"
          update   = "1000" >

</snmp>
<!-- ifOperStatus1 -->
<snmp      id      = "100003001"
          ptype    = "value"
          vtype    = "short"
          rexp     = ".*\s+"
          addr     = ".1.3.6.1.2.1.2.2.1.8.1"
          sdep     = "100003000"
          update   = "5000" >

          <rule key="1" value="0" />
          <rule key="2" value="1" />
          <rule key="3" value="1" />

</snmp>
<!-- ifOperStatus2 -->
<snmp      id      = "100003002"
          ptype    = "value"
          vtype    = "short"
          rexp     = ".*\s+"
          addr     = ".1.3.6.1.2.1.2.2.1.8.2"
          sdep     = "100003000"
          update   = "5000" >

          <rule key="1" value="0" />
          <rule key="2" value="1" />
          <rule key="3" value="1" />

</snmp>
<!-- ifOperStatus3 -->
<snmp      id      = "100003003"
          ptype    = "value"
          vtype    = "short"
          rexp     = ".*\s+"
          addr     = ".1.3.6.1.2.1.2.2.1.8.3"
          sdep     = "100003000"
          update   = "5000" >

          <rule key="1" value="0" />

```

```
                <rule key="2" value="1" />
                <rule key="3" value="1" />

                </snmp>
<!-- ifOperStatus4 -->
    <snmp      id      = "100003004"
              ptype   = "value"
              vtype   = "short"
              rexp    = ".*\s+"
              addr    = ".1.3.6.1.2.1.2.2.1.8.4"
              sdep    = "100003000"
              update  = "5000" >

                <rule key="1" value="0" />
                <rule key="2" value="1" />
                <rule key="3" value="1" />

            </snmp>

        </base>
</config>
```

## ПРИЛОЖЕНИЕ 2

### Пример листинга файла dsc.xml

```
<?xml version="1.0" encoding="utf8"?>
<config xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation='dsc.xsd' codepage="utf8">

  <dirs>
    <dir name="etc" path="etc" />
    <dir name="log" path="log" />
    <dir name="tmp" path="tmp" />
  </dirs>

  <debug level="8" log="no" />

  <network_c id="055" port_r="3726" port_t="3725" updt="500"
lupdt="1000" vupdt="1000" logid="001" />
  <network_s id="000" port_r="3728" port_t="3727" logid="002" />

  <servers>
    <server id="001" addr_b="127.0.0.1" addr_r="127.0.0.1"
caption="ServerCpuMain" />
    <server id="081" addr_b="199.168.190.16"
addr_r="199.168.190.16" caption="ServerCpuRes" />
    <server id="005" addr_b="199.168.190.28"
addr_r="199.168.190.28" caption="ArmIngener" />
    <server id="006" addr_b="199.168.190.34"
addr_r="199.168.190.34" caption="ArmAsn" />
  </servers>

  <clients>
    <client id="901" addr_b="127.0.0.1" addr_r="127.0.0.1"
port="3724" caption="scada" />
  </clients>

  <logs>
    <log id="001" caption="udpclient"/>
    <log id="002" caption="udpserver"/>
  </logs>

  <base>

  <!-- ServerCpuMain -->
  <!-- PU 100001 -->
    <through id="100001000" serv="001" vtype="short" />
    <through id="100001004" serv="001" vtype="string" />
    <through id="100001005" serv="001" vtype="string" />
    <through id="100001008" serv="001" vtype="double" />
    <through id="100001010" serv="001" vtype="double" />
    <through id="100001011" serv="001" vtype="double" />
    <through id="100001012" serv="001" vtype="double" />
    <through id="100001013" serv="001" vtype="double" />
    <through id="100001014" serv="001" vtype="double" />

  <!-- MOXA Modbus1 -->
    <through id="100003000" serv="001" vtype="short" />
    <through id="100003001" serv="001" vtype="short" />
  </base>
</config>
```

```
<through id="100003002" serv="001" vtype="short" />
<through id="100003003" serv="001" vtype="short" />
<through id="100003004" serv="001" vtype="short" />

<!-- ServerCPUres -->

<through id="101181000" serv="081" vtype="short" />
<through id="101181004" serv="081" vtype="string" />
<through id="101181005" serv="081" vtype="string" />
<through id="101181008" serv="081" vtype="double" />
<through id="101181010" serv="081" vtype="double" />
<through id="101181011" serv="081" vtype="double" />
<through id="101181012" serv="081" vtype="double" />
<through id="101181013" serv="081" vtype="double" />
<through id="101181014" serv="081" vtype="double" />

<!-- ARmIngener -->

<through id="100005000" serv="005" vtype="short" />
<through id="100005004" serv="005" vtype="string" />
<through id="100005005" serv="005" vtype="string" />
<through id="100005008" serv="005" vtype="double" />
<through id="100005010" serv="005" vtype="double" />
<through id="100005011" serv="005" vtype="double" />
<through id="100005012" serv="005" vtype="double" />
<through id="100005013" serv="005" vtype="double" />
<through id="100005014" serv="005" vtype="double" />
<through id="100005015" serv="005" vtype="short" />
<through id="100005016" serv="005" vtype="short" />
<through id="100005017" serv="005" vtype="string" />
<through id="100005018" serv="005" vtype="string" />
<through id="100005019" serv="005" vtype="string" />
<through id="100005020" serv="005" vtype="string" />

<!-- ArmAsn -->

<through id="100006000" serv="006" vtype="short" />
<through id="100006004" serv="006" vtype="string" />
<through id="100006005" serv="006" vtype="string" />
<through id="100006008" serv="006" vtype="double" />
<through id="100006010" serv="006" vtype="double" />
<through id="100006011" serv="006" vtype="double" />
<through id="100006012" serv="006" vtype="double" />
<through id="100006013" serv="006" vtype="double" />
<through id="100006014" serv="006" vtype="double" />
<through id="100006015" serv="006" vtype="short" />
<through id="100006016" serv="006" vtype="short" />
<through id="100006017" serv="006" vtype="string" />
<through id="100006018" serv="006" vtype="string" />
<through id="100006019" serv="006" vtype="string" />
<through id="100006020" serv="006" vtype="string" />

</base>

</config>
```

**ПРИЛОЖЕНИЕ 3**

**Коды единиц измерения**

Код	Единицы измерения
6	PSI
7	Бар
11	Паскаль
12	Килопаскаль
16	Галлон/Минуту
17	Литр / Минуту
19	Метр кубический/Час
20	Feet / Секунду
21	Метр / Секунду
22	Галлон / Секунду
24	Литр / Секунду
26	Feet кубический / Секунду
27	Feet кубический / День
32	Градус Цельсия
33	Градус Фаренгейта
35	Кельвин
36	Милливольт
38	Вольт
40	Галлон
41	Литр
43	Метр Кубический
50	в минуту
51	в секунду
52	в час
53	в день
112	Feet кубический
130	Feet кубический / Час
131	Feet кубический / Мин
132	Баррель/Секунду
133	Баррель/Минуту
134	Баррель/Час
135	Баррель/День
136	Галлон/Час
138	Литр/Час
235	Галлон/День
237	Мегапаскаль
246	Литр/День
247	Децибел

**ПРИЛОЖЕНИЕ 4**

**Расшифровка статусов команд модуля HART**

<b>Статус</b>	<b>Тип статуса</b>	<b>Расшифровка</b>
passed	1	Выполнено успешно
Passed parameter too small	1	Введенный параметр слишком мал
Write-protect mode	1	Режим защиты от записи
Set to nearest possible value	1	Установлено в ближайшее возможное значение
Update in progress	1	Обновление в процессе
Not in proper current mode (fixed at 4 mA or 20 mA)	1	Не верное значение тока (фиксировано от 4 до 20 мА)
In multidrop mode	1	В multidrop режиме
Applied process too high	1	Значение слишком велико
Applied process too low	1	Значение слишком мало
Lower range value too low	1	Нижний предел слишком мал
Lower range value too high	1	Нижний предел слишком высок
Upper range value too high	1	Верхний предел слишком высок
Upper range value too low	1	Верхний предел слишком мал
Both range values out of limits	1	Оба предела выходят за диапазон
Span too small	1	Значение слишком мало
Pushed upper range value over limit	1	Верхнее значение вышло за диапазон
Command not implemented	1	Команда не разработана
Field device malfunction	2	Неисправность HART-устройства
Configuration changed	2	Конфигурация изменена
Cold start	2	Ожидается запуск
More status available	2	Больше статуса доступно (в 48 команде)
Analogue output current fixed	2	Значение тока установлено
Analogue output saturated	2	Значение тока вернулось в нормальный режим
Non primary variable out of limits	2	Не основная переменная вышла за границу диапазона
Primary variable out of limits	2	Основная переменная вышла за границу диапазона

### Перечень принятых сокращений

БД	база данных
ОЗУ	оперативное запоминающее устройство
ОС	операционная система
ПО	программное обеспечение
ПП	программная платформа
СВУ	среда визуализации и управления
API	application programming interface (программный интерфейс приложения)
SCADA	диспетчерское управление и сбор данных (Supervisory Control And Data Acquisition)



